

CIENCIAS DE LA
COMPUTACIÓN
PARA EL AULA

Manual para docentes

2º CICLO PRIMARIA



MANUAL 2º CICLO PRIMARIA

AUTORES (por orden alfabético)

Carlos Areces (Capítulo 2)
Luciana Benotti (Introducción y Capítulo 5)
Joshep Joel Cortez Sánchez (Capítulo 1)
Raul Fervari (Anexo I)
Ezequiel García (Capítulos 6 y 7)
Marcos Gómez (Capítulo 4)
María Cecilia Martínez (Introducción y Capítulo 1)
Carlos Martín Onetti (Anexo II)
Eduardo Sebastián Rodríguez Pesce (Anexo II)
Nicolás Wolovick (Capítulos 6 y 7)

COORDINADORA DEL EQUIPO DE AUTORES

Luciana Benotti

COORDINADORA PEDAGÓGICA

María Cecilia Martínez

REVISORES DE CONTENIDOS

Julián Dabbah
Pablo Factorovich
Alfredo Olivero

EDITORES

Ignacio Miller
Alejandro Palermo

CORRECTORA

Luz Rodríguez

DISEÑADORES GRÁFICOS

Luciano Andújar
Jaqueline Schaab

ILUSTRADORAS

Romina Castro
Paula Ramos

DISTRIBUCIÓN
LIBRE Y GRATUITA

COLECCIÓN CIENCIAS DE LA COMPUTACIÓN PARA EL AULA

EDITORES GENERALES

Hernán Czemerinski
Vanina Klinkovich

SUPERVISOR DISCIPLINAR

Franco Frizzo

FUNDACIÓN DR. MANUEL SADOSKY

COORDINADORES DE INICIATIVA PROGRAM.AR

María Belén Bonello
Fernando Schapachnik

DIRECTOR EJECUTIVO

Esteban Feuerstein

PRESIDENTE

Ministro de Ciencia, Tecnología e Innovación Productiva
José Lino Barañao

Ciencias de la computación para el aula : 2do. ciclo de primaria : libro para docentes / Carlos Areces ... [et al.] ; contribuciones de Franco Frizzo ; Pablo Matías Factorovich ; Alfredo Olivero ; compilado por Luciana Benotti ; María Cecilia Martínez ; coordinación general de Hernán Czemerinski ; Vanina Klinkovich ; editado por Ignacio David Miller ; Alejandro Palermo ; editor literario Luz Luz María Rodríguez ; ilustrado por Luciano Andújar ... [et al.] ; prólogo de María Belén Bonello ; Fernando Pablo Schapachnik. - 1a ed. - Ciudad Autónoma de Buenos Aires : Fundación Sadosky, 2018.
Libro digital, PDF - (Ciencias de la Computación para el aula / Klinkovich, Vanina; Czemerinski, Hernán; 1)

Archivo Digital: descarga
ISBN 978-987-27416-5-5

1. Informática. I. Areces, Carlos II. Frizzo, Franco, colab. III. Factorovich, Pablo Matías, colab. IV. Olivero, Alfredo, colab. V. Benotti, Luciana, comp. VI. Martínez, María Cecilia, comp. VII. Czemerinski, Hernán, coord. VIII. Klinkovich, Vanina, coord. IX. Miller, Ignacio David, ed. X. Palermo, Alejandro, ed. XI. Luz María Rodríguez, Luz, ed. Lit. XII. Andújar, Luciano, ilus. XIII. Bonello, María Belén, prolog. XIV. Schapachnik, Fernando Pablo, prolog.

CDD 005.1



CIENCIAS DE LA
COMPUTACIÓN
PARA EL AULA

Manual para docentes

2º CICLO PRIMARIA

ÍNDICE

7	PRÓLOGO
11	INTRODUCCIÓN
31	CAPÍTULO 1: CIUDADANÍA DIGITAL Y SEGURIDAD
77	CAPÍTULO 2: ALGORITMOS Y PROGRAMAS
107	CAPÍTULO 3: EVENTOS, CONDICIONALES Y CICLOS
135	CAPÍTULO 4: PROCEDIMIENTOS
197	CAPÍTULO 5: REPRESENTACIÓN DE DATOS
239	CAPÍTULO 6: LA COMPUTADORA
269	CAPÍTULO 7: LA MEMORIA Y LA UNIDAD CENTRAL DE PROCESAMIENTO
297	GLOSARIO

PRÓLOGO

La tarea de prologar estos manuales se asemeja, para nosotros, a la de colocar el cartel que dice “bienvenidos” en la puerta de un edificio de varios pisos. Este gesto, que es final e inaugural a la vez, corona años (literalmente hablando) de duro trabajo y, a la vez, anticipa la espera ansiosa de la etapa siguiente: su uso en las aulas.

Se trata de los **primeros manuales escolares sobre Ciencias de la Computación en el escenario editorial argentino, que se ponen a disposición del público de manera libre y gratuita**, y que se suman a un pequeño grupo de ejemplos pioneros a nivel mundial en esta temática. Somos conscientes de que, al hablar de “manuales escolares sobre Ciencias de la Computación”, queda mucho por aclarar. Comencemos por el principio: ¿por qué Ciencias de la Computación?

Ciencias de la Computación es el nombre que recibe el área del conocimiento que aporta una serie de saberes (programación, funcionamiento de las computadoras e Internet, Inteligencia Artificial, etc.) **que resultan fundamentales para comprender el mundo cada vez más tecnológico en el que viven y se desarrollan los alumnos que transitan su escolaridad hoy en día.** Sin estos conocimientos, su comprensión de la realidad se verá limitada, y no podrán participar como ciudadanos activos e informados en los debates actuales sobre las múltiples interacciones entre la tecnología informática y la sociedad. La Argentina ha decidido avanzar sobre esta materia y es por eso que **el Consejo Federal de Educación declaró, mediante su resolución 263/15, que la enseñanza y el aprendizaje de programación es de importancia estratégica para fortalecer el desarrollo socioeconómico de la Nación.**

Estos manuales **se concibieron para el aula**, como una herramienta para el docente, al que le brindan secuencias didácticas detalladas junto con fichas de trabajo para entregar a sus estudiantes. ¿Para qué aula, para qué docentes, para qué estudiantes? En principio, estas actividades están pensadas para el aula argentina. Este material fue **escrito en su totalidad por y para argentinos y argentinas**, tomando como referencia la realidad de la escuela argentina. Esto se refleja en el lenguaje, en las referencias y en los marcos culturales que se utilizan, características que no impiden que demos la bienvenida e incentivemos su uso en otros países de la región y del mundo.

Los cuatro manuales que componen esta colección **cubren, respectivamente, el primer ciclo de la educación primaria, el segundo ciclo de la educación primaria, el primer ciclo de la educación secundaria y el segundo ciclo de la educación secundaria.** El rango etario define, en cada caso, el recorte de temas, la profundidad con que son abordados, el registro del texto y la línea estética. En su

mayoría **son manuales iniciales** (es decir, tres de ellos están concebidos para alumnos y alumnas que dan sus primeros pasos en Informática, a distintas edades). El manual destinado al segundo ciclo de la escolaridad primaria tiene como antecedente nuestro primer manual para programar en el aula¹.

En cuanto a sus destinatarios principales, los docentes, estos manuales buscan interpelar a un conjunto de profesionales cuyas formaciones en el área son heterogéneas. Es por esto que, sin pretender presentar explicaciones teóricas exhaustivas, a lo largo de los distintos capítulos hay desarrollos conceptuales que contribuyen a que aquellos y aquellas docentes que no poseen un dominio fluido de ciertos temas puedan contar con las nociones fundamentales. Al respecto, vale destacar que la Fundación Sadosky ha capacitado hasta abril de 2018 (a través de convenios con universidades públicas de todo el país) a más de 1500 docentes, que se suman a los que han formado los ministerios de educación nacional y provinciales. Estos docentes encontrarán particular provecho en nuestro material.

Aunque gran parte del material fue testeado y consultado con los y las docentes de primaria y secundaria que tomaron nuestros cursos, somos conscientes de que solo su uso de manera sistemática en la escuela permitirá mejorarlo.

Asimismo, resulta pertinente aclarar que estos manuales **están pensados para abordar contenidos de Ciencias de la Computación en espacios disciplinares específicos**. Distintas jurisdicciones del país (Neuquén, CABA, Tucumán, entre otras) cuentan con estos espacios curriculares, mientras que otras están discutiendo su incorporación. Desde Program.AR entendemos que este material constituye un aporte a ese camino, que se suma a las anteriormente publicadas Planificaciones anuales para Tecnología de la Información de 3^{er} y 4^{to} año de CABA².

En cuanto al enfoque didáctico, las secuencias propuestas están pensadas, en buena parte, desde la **perspectiva del aprendizaje por indagación**. Imaginamos que los manuales serán usados por docentes y estudiantes que transitan un camino de descubrimiento, asociado a la tecnología informática que media en buena parte de nuestras interacciones con el mundo.

¹ Manual de actividades para Program.AR, disponible en <http://program.ar/manual-docentes-primaria/>

² Disponibles en <http://program.ar/planificacion-anual-ti3/> y <http://program.ar/planificacion-anual-ti4/>.

Merece ahondarse en el proceso que hoy encuentra un hito en la publicación de estos manuales. Al pensar de qué manera debía construirse este material y quiénes debían ser los encargados de hacerlo, recurrimos a la herramienta que más garantías ofrece en términos de calidad, apertura y transparencia: **se realizó una convocatoria pública a las universidades que componen el sistema de generación de conocimiento de nuestro país.** Como resultado de esa convocatoria, y mediante la evaluación de un jurado internacional, resultaron elegidas cuatro universidades nacionales que formaron equipos autorales compuestos por profesionales de las Ciencias de la Computación y de la Educación.

Los manuales fueron escritos por colegas de la **Universidad Nacional del Centro de la Provincia de Buenos Aires** (primer ciclo de primaria), la **Universidad Nacional de Córdoba** (segundo ciclo de primaria), la **Universidad Nacional de Quilmes** (primer ciclo de secundaria) y la **Universidad Nacional de La Plata** (segundo ciclo de secundaria), con quienes estamos profundamente agradecidos por su compromiso y profesionalismo en un proceso que fue novedoso para todos los involucrados. Cada manual tiene una impronta propia, a través de la cual se traducen las diferentes miradas, prioridades y valoraciones que los distintos colegas otorgan a diversos aspectos de las Ciencias de la Computación.

Deseamos destacar la labor de todo el **equipo de la Fundación Sadosky** que participó en el desarrollo de estos manuales. Estas palabras apenas resumen un proceso que implicó años de trabajo y, por sobre todas las cosas, el compromiso de todos los que lo hicieron posible. En primer lugar, agradecemos a Hernán Czemerinski y Vanina Klinkovich, editores generales de la colección, quienes realizaron un gran trabajo dirigiendo este proyecto desde su génesis en 2016 y se ocuparon, además, de revisar, corregir, unificar y armonizar todas las visiones.

Sumamos a este agradecimiento a Franco Frizzo, que ha supervisado el contenido de toda la colección, y a Jaqueline Schaab, a cargo del diseño gráfico. Queremos también destacar la participación del conjunto de revisores, desarrolladores y gestores de la Fundación Sadosky que participaron en este proyecto (por orden alfabético): Julián Dabbah, Pablo Factorovich, Mariana Labhart, Alfredo Sanzo, Herman Schinca, Daniela Villani. Sin su colaboración, estos manuales no serían una realidad.

Dedicamos un párrafo especial al querido Alfredo Olivero, a la vez mentor y compañero de ruta, de un humor tan incisivo como su inteligencia, que fue parte de este equipo y a quien extrañamos mucho.

Asimismo, no queremos dejar de agradecer al equipo de legales, administración y gestión de la Fundación Sadosky compuesto por Roxana Ríos, Andrea Córdoba, Rosa Córdoba, Mariano Tiseyra, Melina Rodríguez y a su Director Ejecutivo Esteban Feuerstein. Queremos también expresar nuestro agradecimiento a Santiago Ceria, quien ocupaba la Dirección Ejecutiva al momento de comenzar este proyecto.

Al mismo tiempo, fue necesario contar con los servicios de profesionales encargados de la edición y corrección de los textos, el diseño y la ilustración. Es por esto que también agradecemos al equipo cuyo talento permitió tener este material en su forma actual: Ediciones Colihue, Ignacio Miller, Alejandro Palermo, Luciano Andújar, Celeste Maratea y Luz Rodríguez.

En cuanto al **uso del género gramatical en esta colección**, hemos decidido respetar la norma vigente del uso del masculino para grupos mixtos. Conscientes de que esta decisión deja pasar una oportunidad para contribuir a la construcción de una norma más inclusiva, optamos por un texto que resultara menos disruptivo.

Esta colección se pone a disposición del público con **licencia Creative Commons**¹, como una forma de incentivar la creación de obras derivadas. Dicho de otra forma, fomentamos activamente que las y los colegas generen sus propias versiones de este material y las compartan con la comunidad.

Estos manuales son para nosotros una versión 1.0 que, con un fuerte anclaje en el aula, **no deja de tener carácter experimental**. Muchos de los temas abordados cuentan con pocos antecedentes en la bibliografía internacional: existe mucho material para enseñar programación inicial a niños y adolescentes, mucho menos sobre cómo funciona una computadora, y muy poco sobre otras cuestiones tales como el funcionamiento de Internet, la representación de la información o la Inteligencia Artificial. Sabemos también que algunos temas que merecerían tener un lugar han quedado afuera, algo que pensamos subsanar en las próximas ediciones.

Nos encantaría enriquecer este material con los aportes de la comunidad: docentes, académicos, investigadores e interesados en la temática en general están invitados a acercarnos sus comentarios, críticas y sugerencias en info@program.ar. A su vez, queda abierta la invitación a revisar periódicamente nuestro sitio web o seguirnos en las redes sociales, para mantenerse al tanto de las futuras versiones.

María Belén Bonello y Fernando Schapachnik
Coordinadores de la Iniciativa Program.AR
Fundación Dr. Manuel Sadosky

¹ Específicamente, una licencia Creative Commons Atribución-No Comercial-CompartirIgual 4.0 Internacional, cuyos detalles pueden consultarse en <https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>.



INTRODUCCIÓN

Por Luciana Benotti y María Cecilia Martínez

El presente manual ha sido elaborado para ayudar a los docentes del segundo ciclo del nivel primario en la enseñanza de conceptos centrales de las Ciencias de la Computación. Esta asignatura nos ayuda a entender cómo funciona una computadora por dentro y cómo es posible programarla para resolver problemas.

Cada capítulo encara un área central dentro de la disciplina, cuyo abordaje consideramos necesario para entender el mundo tecnológico donde vivimos. Se proponen actividades con descripciones paso a paso para ponerlas en práctica en las aulas. Además, muchas de esas actividades vienen acompañadas de una ficha para repartir a los estudiantes, con consignas e información relevante sobre el tema trabajado en la actividad.

En esta introducción respondemos preguntas básicas sobre las motivaciones y consideraciones de este manual y explicamos los principios que guiaron el desarrollo de nuestra tarea. Además, presentamos un resumen de los objetivos y los contenidos de cada capítulo.

LAS CIENCIAS DE LA COMPUTACIÓN COMO OBJETO DE ESTUDIO

A partir del desarrollo de las computadoras personales que se produce en la década de 1980, las escuelas comienzan a ofrecer formación en computación enfatizando diferentes aspectos del área (Levis y Cabello, 2007). En los años ochenta predomina un enfoque técnico que aborda el *hardware* y los comandos básicos. Los años noventa se centran en la enseñanza del uso de programas de oficina. En la primera década del milenio se busca integrar las TIC (Tecnologías de la Información y la Comunicación) para potenciar los contenidos escolares. Todos estos enfoques conciben a la computadora como una herramienta para automatizar o digitalizar algunas de las tareas que la escuela ya realizaba en forma manual, tales como escribir un texto o armar balances de contabilidad. Sin embargo, no abordan todavía la computación como objeto de estudio. Recién a partir de 2010 nacen iniciativas a nivel mundial para incluir la enseñanza de las Ciencias de la Computación en la escuela.

Este manual concibe la Computación como una disciplina al igual que Matemática o Lengua. En él se presenta un conjunto de conceptos fundamentales que permiten entender cómo funcionan la mayoría de los programas que usamos, desde los videojuegos hasta la tarjeta para pagar el transporte.

Conocer los fundamentos de la computación también nos permite pensar qué tipo de soluciones podría aportar una computadora a diferentes problemas sociales; por ejemplo, en el campo de la salud, se acude a la computación para realizar diagnóstico por imágenes. Entender qué funciones puede o no realizar una computadora es lo que varios académicos han denominado **pensamiento computacional**¹ (Zapata Ros, 2015). Muchos de los conceptos fundamentales de la disciplina, varios de los cuales provienen de la lógica y la matemática, se abordan en este manual con el objetivo de comprender temáticas actuales y relevantes para nuestros jóvenes. Esos conceptos incluyen, en primer lugar, conocimientos sobre la seguridad informática y las redes de computadoras, que permiten una mayor conciencia en el uso de la tecnología de los ciudadanos digitales superconectados. Además, se abordan conceptos y técnicas de programación esenciales que ponen al alcance de los alumnos la posibilidad de manejar el lenguaje de las computadoras y pasar de ser usuarios a ser creadores de *software*. Finalmente, se propone un viaje al interior de los sistemas de computación y un paseo por las técnicas de representación de la información, que muestran que las computadoras son máquinas extremadamente básicas, aun cuando las percibamos como inteligentes por su habilidad para representar y procesar inmensos volúmenes de información casi instantáneamente.

El pensamiento computacional requiere la comprensión de las capacidades de una computadora y la aptitud para expresar un problema de forma tal que una computadora lo pueda resolver. Crear un programa es una forma de externalizar nuestros pensamientos en un lenguaje no ambiguo. Esto nos permite analizar nuestros razonamientos y encontrar errores en ellos. La forma más efectiva de desarrollar el pensamiento computacional es aprender Ciencias de la Computación.

¹ Si bien existen otras acepciones de la expresión *pensamiento computacional*, en este manual se usa la de Zapata Ros (2015).

A pesar de su nombre, el pensamiento computacional es una habilidad humana. Nosotros pensamos, las computadoras siguen instrucciones. Esta sutil distinción está en la base de las actividades sobre pensamiento computacional propuestas en este manual, que además no requieren el uso de una computadora; por este motivo las llamamos “actividades desenchufadas”. Así como el pensamiento computacional puede tener lugar sin una computadora de por medio, el mero uso de una computadora en el aula no logra desarrollar el pensamiento computacional.

Los principales conceptos¹ que se abordan en este manual representan áreas específicas y de gran importancia práctica para la disciplina. Nuestro enfoque se centra en estas ideas prácticas y no en teorías abstractas y generales.



Se utilizaron los siguientes criterios para la selección de los conceptos trabajados: (i) que tengan una amplia importancia en distintas áreas de las Ciencias de la Computación; (ii) que sirvan como base para aprender o construir otras ideas de la disciplina; (iii) que generen entusiasmo en estudiantes jóvenes y potencien la posibilidad de que profundicen los saberes impartidos, guiados por su propia curiosidad; y (iv) que sigan siendo relevantes para las Ciencias de la Computación durante los próximos diez años, al menos.

Las actividades propuestas permiten que los estudiantes construyan los conceptos fundamentales de la computación poniendo en práctica niveles de pensamiento de alto orden a partir de la identificación de patrones, la interacción sujeto-máquina, la creatividad, la resolución de problemas, el desarrollo del pensamiento abstracto, el método colaborativo y la metacognición.

¹ Estos conceptos coinciden con los que se proponen en el K12 Computer Science Framework, creado colaborativamente por la Association for Computer Machinery (ACM) y la Computer Science Teacher Association (CSTA) de Estados Unidos en 2016, que plantea estándares para la creación de material didáctico en el área de las Ciencias de la Computación.

Estas actividades se basan en grandes pilares de la computación:

- Hacer computación y **programar** son actividades creativas.
- La **abstracción** permite reducir los detalles para centrarse en la información relevante para resolver un problema.
- La **información** accesible facilita el desarrollo de saberes.
- Los **algoritmos** permiten expresar soluciones a problemas que se resuelven con una computadora.
- Los artefactos digitales, los sistemas y las redes que los conectan promueven enfoques computacionales para resolver problemas a través de la **coordinación** y la **comunicación**.
- La computación permite innovaciones en **otros campos**, incluyendo ciencias exactas, ciencias naturales, humanidades, artes, medicina, ingeniería, etc.

De esta manera, la enseñanza de este campo del conocimiento en la escuela contribuye a la formación del pensamiento crítico de nuestros estudiantes. En ese sentido, realiza un aporte sustantivo a los objetivos generales del último ciclo del nivel primario.

¿QUÉ PRÁCTICAS DE ENSEÑANZA Y DE APRENDIZAJE PROPONE ESTE MANUAL?

El desarrollo de las actividades propuestas en este manual está guiado por las teorías del aprendizaje por descubrimiento y busca que los alumnos logren construir los conceptos por sí mismos. Si bien la programación ingresa a algunas escuelas a fines de los años ochenta, su enseñanza no siempre ha reflejado el espíritu constructor que permite la genuina comprensión de conceptos. En ocasiones, se copian programas en una pizarra y los alumnos deben reescribirlos en su computadora, o bien los estudiantes deben programar ejercicios desconectados de sus realidades y sus intereses. Es por ello que diseñar experiencias de enseñanza de programación es tan importante como transmitir los conceptos necesarios.

En efecto, la enseñanza debe ser pensada. Resulta fundamental que una actividad tan importante como la transmisión de la cultura a las nuevas generaciones sea planificada. Por más experiencia y creatividad que posea el docente, la enseñanza no puede quedar librada a la improvisación. El docente debe reflexionar sobre su actividad profesional y prever sus acciones teniendo presentes las particularidades de sus estudiantes y del contexto, en función de las características del objeto de conocimiento que pretende transmitir (Davini, 2008).

La elaboración de proyectos significativos nos remite a lo que Papert denomina *resonancia cultural*, es decir, lo que el niño o joven pueda registrar dentro de su esquema cultural. Como Papert, creemos que las mejores experiencias de aprendizaje, para la mayoría de la gente, vienen cuando están activamente comprometidos en diseñar y crear cosas, especialmente cosas que son significativas para ellos o para otros alrededor de ellos (Resnick y Silverman, 2005). Siguiendo enfoques de la pedagogía auténtica, la experiencia de la enseñanza debe diseñarse de modo tal que la única forma de completar el aprendizaje sea desplegando habilidades que el docente considera relevante que los alumnos desarrollen (Newmann, 1996).

Desde esta perspectiva, una estrategia que permite secuenciar los contenidos para promover la construcción de los conceptos es la enseñanza basada en proyectos. En ese sentido, los capítulos introducen los temas con una pregunta, una noticia del diario, un juego, una anécdota o la descripción de una situación cotidiana, familiar y relevante para los estudiantes. Se formulan preguntas que permiten recuperar nociones previas que los alumnos puedan tener sobre los conceptos a trabajar, y actividades para encontrar regularidades y patrones desde los cuales construir abstracciones. Se proponen, además, proyectos para profundizar los conceptos y complejizar las prácticas de programación de los alumnos. Entre la presentación de las situaciones que facilitan la comprensión de los fundamentos de la computación y los proyectos, se ofrecen actividades intermedias, que luego se ponen en práctica. En ese sentido, se sostiene que los saberes se construyen a lo largo de las secuencias didácticas de manera espiralada.

En este manual hay una intención didáctica: se seleccionaron conceptos que son centrales en la disciplina y que no han cambiado por varias décadas. A continuación describimos los contenidos que no han sido incluidos por decisión de diseño.

Este no es un manual sobre el lenguaje de programación Scratch. Scratch es la herramienta elegida para ilustrar algunos conceptos fundamentales de la programación, pero estos son comunes a casi cualquier otro lenguaje. Además, no es nuestro objetivo introducir todas las características de Scratch, sino solo aquellas partes que tiene en común con la mayoría de los lenguajes de programación.

Por otro lado, este manual no enseña a usar ningún *software*. Intencionalmente, muchas de las actividades no requieren el uso de una computadora. Por lo tanto, la tecnología no actúa como barrera al momento de acceder a los conocimientos. El objetivo es enseñar cómo funciona internamente la tecnología, no usar la tecnología en sí.

Por último, este manual no incluye definiciones completas enciclopédicas de los conceptos que aborda. El aprendizaje se realiza a través del uso contextualizado de la terminología apropiada.

UN EQUIPO DE DESARROLLO INTERDISCIPLINARIO

En el año 2013, el Ministerio de Ciencia, Tecnología e Innovación Productiva de la Nación y la Fundación Sadosky desarrollan el proyecto Program.AR, que busca promover la enseñanza de las Ciencias de la Computación en las escuelas primarias y secundarias. Esta iniciativa busca dar una respuesta a la preocupación que existe en la Argentina y en el mundo por el desconocimiento que tienen nuestros jóvenes y niños sobre las computadoras que utilizan de modo intensivo. Los chicos son grandes usuarios y consumidores de tecnologías elaboradas por otros, pero no pueden modificarlas ni crearlas ni producirlas, porque no comprenden cómo funcionan. Este manual es un instrumento más para una política educativa orientada a mejorar la oferta de enseñanza en contenidos de las Ciencias de la Computación en las escuelas.

Su desarrollo requirió la intervención de educadores, investigadores y organizaciones que delinearon los conceptos y prácticas en Ciencias de la Computación para todos los estudiantes del segundo ciclo de primaria de la Argentina. Este proyecto fue guiado por un comité formado por representantes de la Fundación Sadosky, del Ministerio de Ciencia, Tecnología e Innovación Productiva de la Nación, del área de Educación del Instituto de Humanidades (IDH-CONICET) y de la Sección de Computación de la Facultad de Matemática, Astronomía, Física y Computación (FAMAF) de la Universidad Nacional de Córdoba (UNC).

Los autores de este manual trabajan desde el año 2012 en el área de la Enseñanza de las Ciencias de la Computación en tres líneas principales: (i) desarrollo de material didáctico, que incluye plataformas informáticas para aprender computación y planes de clases disponibles de forma libre y gratuita en la página de nuestro equipo (<http://masmas.unc.edu.ar/>); (ii) diseño y ejecución de cursos de capacitación docente (con unos trescientos docentes capacitados anualmente en la provincia de Córdoba); y (iii) desarrollo de experiencias piloto de enseñanza de programación en escuelas primarias y secundarias de diversos contextos. Este material se basa en los saberes construidos a partir de las diferentes experiencias realizadas en las tres líneas descritas.

Las Ciencias de la Computación son una de las disciplinas actuales que crece más rápido. A nivel mundial, se vienen realizando una gran cantidad de investigaciones sobre la enseñanza de esta materia en primaria y secundaria. Aún quedan muchas lecciones por aprender, pero los sistemas educativos de distintos países están tomando medidas para aumentar el acceso de los estudiantes a las Ciencias de la Computación. La comunidad que ha impulsado y sustentado este proyecto cree que el desarrollo de material educativo de calidad es un paso inicial para informar, inspirar y orientar el trabajo de formación e implementación necesario para lograr que las Ciencias de la Computación sean accesibles a todos los estudiantes.

¿QUÉ DISTINGUE A ESTE MANUAL?

El material fue pensado siguiendo orientaciones de la didáctica específica de la enseñanza de las Ciencias de la Computación de diferentes autores tales como Resnick, Papert y Bell. Ellos recuperan aportes del aprendizaje por descubrimiento, el constructivismo y la pedagogía auténtica; este manual propone que los estudiantes aprendan Ciencias de la Computación a partir de proyectos y actividades que se caracterizan por:

Plantear preguntas, tareas y situaciones significativas para los estudiantes

Desarrollar un videojuego, analizar situaciones de delitos informáticos y aprender a enviar mensajes secretos son ejemplos de algunas situaciones que pueden motivar a los estudiantes y que activan la apropiación de conceptos centrales de las Ciencias de la Computación.

Presentar actividades que no tienen una sola solución

Recuperando la premisa de Resnick sobre planificar actividades de piso bajo, techo alto y paredes anchas (Resnick, 2008), se ofrecen propuestas de programación y representación de conceptos que permiten obtener resultados básicos (piso bajo) y al mismo tiempo brindan la posibilidad de construir proyectos más complejos y sofisticados a medida que se exploran las temáticas y se conocen y combinan sus herramientas (techo alto). Además, admiten la alternancia en una amplia gama de exploraciones y habilitan una gran variedad de proyectos diferentes (paredes anchas). Por lo tanto, cada propuesta o actividad no será igual para todos los grupos de estudiantes de una clase, lo que facilita que cada grupo desarrolle su potencial y sus gustos.

Trabajar con desafíos para resolver problemas

Muchas de las actividades presentan “ideas poderosas”, en el sentido de que permiten imaginar la automatización de muchas tareas a partir de una idea sencilla. Por ejemplo, la noción de condicional posibilita programar a un Pac-Man para que titile cuando come una cereza; pero ese mismo concepto también permite programar una alarma de incendio para que se dispare cuando detecta humo. Pero para que estas ideas poderosas no sean trivializadas, es necesario que el estudiante “descubra” el potencial que poseen para resolver un problema significativo para él. Es por ello que el trabajo con desafíos resulta una estrategia útil para favorecer la emergencia de tales ideas e invita a nuestros estudiantes a explorarlas y entenderlas. La propuesta por desafíos consiste en invitar a los estudiantes a resolver un problema, pero intencionalmente no les ofrecemos todos los conceptos ni los pasos para resolverlo. En cambio, les brindamos tiempo de exploración.

Trabajar con actividades “desenchufadas”

Recuperando la visión de CS Unplugged (Bell, Fellows y Witten, 2009), el manual propone una serie de actividades para trabajar sin computadoras. Estas actividades permiten comprender en profundidad las nociones que se presentan en cada caso. Representar con el cuerpo, al aire libre, con afiches, con cartas y otros elementos facilita la comprensión de diferentes conceptos de computación y el acceso a la esencia del concepto más allá de su automatización. La falta de computadoras y de acceso a Internet no es un problema para este tipo de actividades. En este manual, la mayoría de los conceptos de Ciencias de la Computación se pueden aprender a través de actividades grupales que no requieren el uso de una computadora. Los estudiantes juegan con cartas para aprender números binarios, participan en un juego de rol para comprender cómo la información fluye dentro de los componentes de una computadora y cantan canciones para descubrir qué son los procedimientos. Estas actividades se combinan en el manual con ejercicios de programación para proveer una rica experiencia a los estudiantes.

Programar en entorno con bloques

Programar con bloques de colores que identifican tipos de funciones diferentes facilita la apropiación intuitiva de grupos de conceptos relacionados. Por ejemplo, los bloques amarillos en Scratch permiten manejar el comportamiento del programa usando estructuras de control como repeticiones y condicionales. La programación por bloques también contribuye a que los estudiantes se centren en las ideas y no en la sintaxis de los lenguajes de programación. En otros lenguajes, el error sintáctico es muy frecuente y resulta en que el programa no funcione por más que el conjunto de ideas y funciones de alto orden haya estado bien planteado. Scratch evita los errores sintácticos porque el programa se construye como un rompecabezas de bloques. De la misma manera que sucede en un rompecabezas, Scratch no deja combinar bloques que generarían un error de sintaxis. Esto permite que los estudiantes se apropien de los conceptos centrales de la programación sin el obstáculo frustrante de los errores de sintaxis.

Alentar el trabajo colaborativo

Los intercambios entre los estudiantes promueven la colaboración entre compañeros, motivando la formulación de estrategias para sortear problemas o ideas para continuar desarrollando cada proyecto. Las actividades propuestas, en su mayoría, requieren para ser completadas la colaboración entre los alumnos. El trabajo en equipos no solo es una característica propia de la disciplina, sino que, además, sabemos desde la psicología cognitiva que promueve los aprendizajes genuinos.

Abordar el error y las emociones

En términos generales, podemos decir que el aprendizaje de algo nuevo conduce inevitablemente a atravesar una serie de estados y emociones, que van desde la frustración frente a la dificultad hasta el entusiasmo y la satisfacción cuando se consigue dar con una solución a un problema. Estas emociones se potencian cuando se trabaja en un ambiente con propuestas que desafían a los estudiantes. Resulta importante que, como docentes, acompañemos el proceso de aprendizaje ayudando a los estudiantes a manejar sus emociones. Los invitamos a evitar la sanción del error y en cambio recuperarlo como una nueva oportunidad para que nuestros estudiantes reflexionen sobre su propia producción.

En resumen, este manual ofrece actividades que requieren pensamiento de alto orden y la aplicación de diferentes recursos y sistemas de representación con el objetivo de estimular la construcción de conceptos centrales de la disciplina. Como dijimos, no se prioriza la cobertura de una gran cantidad de contenidos, sino que se vuelve a los mismos conceptos fundamentales una y otra vez desde distintos ángulos. Esta estrategia busca lograr la abstracción progresiva y la comprensión profunda de los conceptos.

¿CUÁLES SON LOS PRINCIPIOS QUE GUIARON EL DESARROLLO DEL MANUAL?

1. Amplia accesibilidad

El manual fue pensado para que docentes de nivel primario, sin formación específica en computación, puedan apropiarse de manera genuina de los conceptos presentados. La estructura de cada secuencia didáctica también refleja esta intención: antes de introducir las fichas de trabajo para los alumnos, se desarrollan conceptualmente los diferentes temas usando ejemplos, analogías y descripciones en un lenguaje coloquial. Estas secciones previas a las fichas fueron escritas para los docentes. Nuestra experiencia en investigación y desarrollo didáctico en la enseñanza de las Ciencias de la Computación nos muestra que una gran mayoría de docentes, con diversas trayectorias de formación y sin conocimientos previos en computación, puede apropiarse de conceptos del área ofrecidos a modo introductorio y presentarlos a sus alumnos de manera efectiva (Martínez, Gómez, Moresi, Benotti, 2016). El material fue diseñado para que los conceptos y técnicas de Ciencias de la Computación, como por ejemplo la programación, puedan ser accesibles a todos los estudiantes argentinos del segundo ciclo de primaria sin importar su nivel socioeconómico ni sus conocimientos previos sobre el tema. Todos los capítulos proponen actividades que enseñan conceptos y prácticas de las Ciencias de la Computación a través de materiales de nulo o muy bajo costo, y muchas de ellas no requieren una computadora. Las actividades de programación que exigen el uso de una computadora se presentan a través de la herramienta Scratch, que es gratuita, de código libre y no necesita Internet.

A través de la amplia accesibilidad del material, este manual intenta que todos los chicos incorporen conocimientos fundamentales que les permitan ganar independencia en el mundo intensamente tecnológico en el que viven. Esta independencia les posibilitará participar activa y productivamente y tomar decisiones conscientes sobre el rol de las tecnologías de la información y las comunicaciones en sus vidas y sus trabajos.

Finalmente, el contenido del manual se distribuye bajo una licencia Creative Commons Compartir Igual. Esta licencia permite (y fomenta) la copia y reproducción de todo el material presentado y también su modificación por parte de los docentes para adaptarlo a las realidades de su escuela y del entorno social de sus alumnos. La única condición es que el material modificado se vuelva a compartir bajo la misma licencia.

2. Enfocarse en lo esencial y conocer el entorno

Este manual desarrolla una alfabetización fundamental en Ciencias de la Computación para el segundo ciclo de primaria. No se propone dar un listado exhaustivo de todos los temas de la disciplina que se podrían aprender en este nivel educativo. Aunque el material presenta conceptos que son esenciales para todos los alumnos del nivel de una forma que consideramos accesible, es nuestra intención incentivar a los docentes a adaptar las actividades propuestas, siempre que lo consideren necesario, para respetar los intereses, habilidades y aspiraciones de sus estudiantes.

El material usa un lenguaje sencillo para los docentes y el público en general. Cuando se usan términos técnicos es porque se considera necesario ser fiel al vocabulario disciplinar y para ilustrar de una manera más acabada los conceptos relevantes. Los conceptos técnicos se listan en un glosario, que tiene como objetivo dar una coherencia terminológica a los distintos capítulos.

No se especifica el tiempo que se debe dedicar a cada actividad. Por el contrario, la naturaleza participativa y conceptual de las secuencias didácticas permite una amplia gama de implementaciones en el aula, incluyendo en algunos casos la integración con otras áreas.

3. Actualizarse con investigaciones internacionales

Los autores de este manual creemos, como Isaac Newton, que “podemos ver más lejos sobre los hombros de gigantes” y eso se hace evidente en dos aspectos principales.

En primer lugar, los conceptos abordados se basan no solo en nuestra experiencia como educadores e investigadores en la enseñanza de Ciencias de la Computación a nivel universitario (Brusilovsky *et al.*, 2014), secundario (Benotti, Martínez y Schapachnik, 2014 y 2017), primario e inicial (Martínez, Gómez y Benotti, 2015, Benotti, Gómez y Martínez, 2017), en situaciones de enseñanza tanto con alumnos como con docentes (Martínez, Gómez, Moresi y Benotti, 2016), sino también en una amplia bibliografía internacional sobre buenas prácticas en la enseñanza de la disciplina. La selección de conceptos incluidos está influenciada por estándares internacionales reconocidos, tales como el K-12 Computer Science Framework de la Association for Computing Machinery y la Computer Science Teachers Association de Estados Unidos (CSframework, 2016); e iniciativas similares de Nueva Zelanda (CSUnplugged 2016, CSFieldGuide 2016); y Europa (Bebras, 2016).

En segundo lugar, este manual incorpora prácticas y actividades de otros manuales del mundo, cuidadosamente seleccionadas y adaptadas para ser relevantes respecto del nivel educativo y las particularidades culturales de nuestro país. Se incluyen actividades cuyas licencias así lo permiten y se citan las fuentes correspondientes. La ventaja de reproducir y adaptar actividades ya existentes es que estas han sido ampliamente probadas en entornos educativos y su efectividad está comprobada. Además, el material incluye otras actividades específicamente diseñadas para este manual, que buscan alcanzar un balance entre aquellas que trabajan la apropiación conceptual y otras que permiten el uso y aplicación de estos conceptos en proyectos significativos para los estudiantes. Entre las fuentes de actividades se destacan las de CS Unplugged (<https://csunplugged.org>) y las de Code (<https://code.org>).

4. La tecnología cambia pero los conceptos permanecen

Como ya se dijo, las Ciencias de la Computación constituyen una de las disciplinas que actualmente cambian con mayor rapidez. Esto se hace evidente, por ejemplo, al observar cómo se modifica la forma de una computadora. Probablemente los niños de hoy no reconozcan una computadora de la década de 1980. Así como cambió la apariencia de las computadoras, también cambiaron otras tecnologías relacionadas con las Ciencias de la Computación. Sin embargo, existen conceptos fundamentales de esta disciplina que no se han modificado desde hace más de setenta años. Esos son los conceptos que abordamos en este manual.

Retomando a Rapenning (Rapenning, Webb y Ioannidou, 2010), creemos que el campo de la enseñanza de las Ciencias de la Computación ha llegado a un punto de madurez promovido por los siguientes avances:

Desarrollo de plataformas didácticas para enseñar a programar

En los últimos años, muchos grupos en distintas universidades del mundo han desarrollado varias plataformas didácticas para enseñar a programar. Estas plataformas, por lo general, recuperan el aspecto lúdico, son ricas visualmente y permiten a los estudiantes crear productos tecnológicos nuevos y ver los resultados rápidamente (aquí usamos Scratch, pero también hay otras de desarrollo nacional como Pilas Bloques, Gobstones, Mumuki, Chatbot y UNC Duino). Las plataformas se basan en el concepto de “piso bajo-techo alto”. Esto significa que debería ser fácil para los programadores principiantes programar algo, pero al mismo tiempo estas plataformas deben promover el avance y el desarrollo de los programadores con más experiencia.

Desarrollo curricular

A partir de informes que publicaron varios sectores preocupados por la enseñanza de las Ciencias de la Computación en las escuelas, con gran presencia del sector industrial que demanda mayor número de egresados en las áreas de computación, muchos países han avanzado en acordar y definir qué conceptos es necesario enseñar. Si bien en nuestro país no hay un currículo oficial, sí existen documentos emitidos por el PLANIED del Ministerio de Educación y Deportes, por la Iniciativa Program.AR de la Fundación Sadosky y el CUCEN (Consejo Universitario de Ciencias Exactas y Naturales).

Evolución del modo en que se trabaja con las tecnologías en las aulas

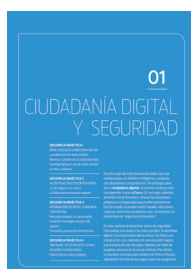
El avance en la didáctica en los últimos años abre una nueva posibilidad de enseñar computación en las escuelas. La escuela argentina enseñó en una primera etapa cuestiones técnicas; luego, a usar programas de oficina, y más tarde a integrar diferentes *softwares* para promover los aprendizajes en diferentes disciplinas. Esta evolución, acompañada de artefactos tecnológicos y pedagógicos, nos invita a pensar cómo seguir mejorando en un nuevo panorama del desarrollo didáctico.

Los avances en estas áreas de la didáctica permiten abordar la enseñanza de la computación en las escuelas de manera genuina, significativa e interesante para los estudiantes. La comprensión de los conceptos contribuye a desterrar el estereotipo de que la computación es solo cosa de los programadores.

¿QUÉ CONTENIDOS ABORDA CADA CAPÍTULO?

Cada capítulo de este manual está integrado por una serie de secuencias didácticas que tienen una coherencia conceptual. Cada secuencia didáctica está conformada por actividades que pueden o no requerir el uso de una computadora. Cada actividad propone formas de abordar la clase, un desarrollo y un cierre. Además, muchas de ellas vienen acompañadas de una ficha para entregar a los estudiantes. Todos los capítulos incluyen un modo de evaluación: se propone utilizar rúbricas y proyectos integradores. Las rúbricas son grillas donde se identifican los aspectos centrales que se evaluarán y los criterios con que se calificará el trabajo.

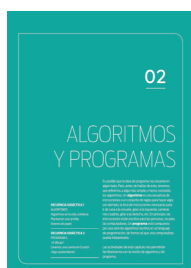
Concebimos la evaluación como parte del proceso de aprendizaje. En este sentido, las rúbricas son herramientas que los estudiantes reciben antes de comenzar a realizar su proyecto integrador evaluable, de manera que puedan orientar y regular su desarrollo. Ofrecer la rúbrica antes de comenzar a realizar un proyecto permite a los estudiantes profundizar la comprensión de la consigna, clarificar cuáles son los conceptos o procesos que se solicita recuperar y analizar cuán lejos están de la meta esperada por el docente. El manual también incluye un glosario para el docente. Las definiciones no fueron pensadas para ser dadas a los estudiantes y se sugiere no presentarlas de forma descontextualizada.



CAPÍTULO 1: CIUDADANÍA DIGITAL Y SEGURIDAD

Parte de la vida de las personas transcurre en el mundo virtual. Ya sea por necesidades de trabajo, trámites, comunicación, diversión u otras, las personas pasan cierto tiempo usando tecnología digital a través de diferentes dispositivos. En la vida cotidiana se comparte mucha información privada a través de redes, cosa que puede provocar eventos que pongan en riesgo la seguridad de la información o engaños ante los que es conveniente aprender a responder del modo correcto. Muchas personas usan mecanismos de autenticación y comunicación segura, sin entender del todo cómo funcionan ni por qué son importantes, lo que da lugar a problemas de seguridad informática.

Al concluir este capítulo, los estudiantes podrán razonar de manera crítica cómo conviene usar las redes sociales y entenderán nociones de privacidad y almacenamiento seguro de la información; serán capaces de recomendar buenas prácticas en el uso de redes sociales y en la creación de contraseñas de acceso; aprenderán por qué es importante elegir contraseñas seguras que sean difíciles de adivinar para otras personas y para las computadoras. Se estudian principios de comunicación segura mediante encriptación, para comprender que, sin esta, todo intercambio de información sería vulnerable. Por último, se ejemplifican algunos engaños típicos que permiten el robo de identidad, las infecciones con diferentes virus y la ejecución de programas maliciosos. El capítulo propone distintas actividades grupales desenchufadas para abordar los diversos temas sin la necesidad de usar dispositivos informáticos.



CAPÍTULO 2: ALGORITMOS Y PROGRAMAS

Es posible que la idea de un programa que se ejecuta en una computadora nos sea familiar. Pero, antes de hablar de programas, tenemos que hablar de algo más simple aunque menos conocido: los algoritmos. Un algoritmo es una secuencia de instrucciones o un conjunto de reglas para hacer algo; por ejemplo, la lista de instrucciones necesarias para ir de casa a la escuela (como “girar a la izquierda, caminar tres cuadras, girar a la derecha”, etc.). Otro ejemplo que seguramente conocemos son los pasos para preparar leche chocolatada: “Poner leche en un vaso, agregar chocolate en polvo, agregar azúcar a gusto”, etc. Notemos que un algoritmo puede ser más o menos detallado (por ejemplo, podemos indicar o no dónde buscar el vaso necesario para preparar la leche chocolatada) y, en principio, está escrito para nosotros, las personas, no para las computadoras. Un programa está formado por una serie de algoritmos escritos de forma tal que una computadora pueda ejecutarlos.

Las actividades de este capítulo permiten que los alumnos se familiaricen con las nociones de algoritmo y programa a través de actividades desenchufadas. Se introduce también Scratch, el entorno y lenguaje de programación que se usa en los siguientes capítulos.



CAPÍTULO 3: EVENTOS, CONDICIONALES Y CICLOS

En este capítulo se introducen algunas estructuras de control comunes a muchos lenguajes de programación: los eventos, los condicionales y los ciclos. Estas construcciones permiten definir cómo es el flujo de ejecución de las instrucciones de nuestros programas. En primer lugar, se presentan herramientas para expresar cómo debe reaccionar un programa ante algún evento externo (por ejemplo, cada vez que se presione una tecla del teclado o un botón del ratón). A continuación, se trabaja sobre sentencias condicionales, que permiten que los programas tomen decisiones. Por último, se ejercita el reconocimiento de patrones y el uso de ciclos, que se utilizan para repetir la ejecución de instrucciones en los programas.

Estos conceptos se abordan inicialmente con actividades desenchufadas, de manera que los estudiantes puedan relacionarlas con nociones que les resultan conocidas. Además, se proponen actividades para desarrollar un proyecto en Scratch que integra cada uno de las estructuras presentadas. Al finalizar el capítulo, los estudiantes habrán creado un programa que podrán ejecutar en una computadora.



CAPÍTULO 4: PROCEDIMIENTOS

A medida que los programas crecen en complejidad y en tamaño, necesitamos de nuevos conceptos y técnicas de resolución de problemas. En este capítulo, a partir del desarrollo del clásico juego Space Invaders, se recuperan y profundizan los conceptos fundamentales de programación trabajados en los capítulos anteriores. A medida que se va desarrollando el juego, se presentan estrategias de división de problemas para modularizar y parametrizar programas. Los estudiantes definirán nuevos procedimientos que resuelven problemas específicos.

Siguiendo la metodología usada en los capítulos anteriores, los conceptos y las técnicas se introducen primero a través de actividades desenchufadas. Por ejemplo, parametrizando dibujos y canciones. Los estudiantes también incorporarán el concepto de variable, y aprenderán cómo definir las, leerlas y modificarlas mientras desarrollan el videojuego.



CAPÍTULO 5: REPRESENTACIÓN DE DATOS

Las computadoras hacen y nos permiten hacer cosas con datos. Nos posibilitan ver, escuchar, crear y editar información: sacar una foto, ver un video, escuchar una canción o escribir un mensaje de chat. Son capaces de hacer cálculos con datos numéricos y nos dejan enviar y recibir datos a través de las redes. Al usar una computadora, jugamos en pequeños mundos inventados que solo existen como datos. Esta información está dentro de la memoria de la computadora y solo una parte se muestra en la pantalla.

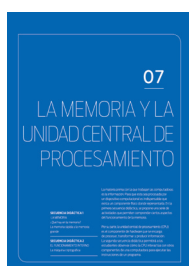
Al finalizar este capítulo, los estudiantes podrán representar diferentes tipos de datos de forma similar a como lo hace una computadora. Representarán números y usarán el sistema de numeración binario. Aprenderán cómo representar palabras no solo con letras, sino también con otros símbolos. Finalmente, se apropiarán de un sistema de representación de colores. Para ello, se proponen diferentes actividades para el aula y un proyecto de programación que permite acercarse a la representación de distintas clases de información. Este capítulo es el último que incluye desafíos de programación que requieren el uso de una computadora.



CAPÍTULO 6: LA COMPUTADORA

Desde su invención, la computadora ha evolucionado a pasos agigantados. En la actualidad, las personas llevan verdaderas supercomputadoras en sus teléfonos celulares. Este proceso de miniaturización ha provocado que la computadora controle procesos de toda índole, así como también que sea utilizada cada vez más en una mayor cantidad de productos de consumo masivo, como celulares, televisores, autos e incluso heladeras y lavarropas. Por lo tanto, resulta fundamental conocer y comprender la tecnología de las computadoras físicas, así como también reconocer aquellos procesos que se apoyan en ellas.

Este capítulo plantea una discusión del concepto de computadora, basada en su función y no en su forma. Los estudiantes aprenderán a identificar las diferentes partes que componen una computadora moderna, así como también la función que cada parte cumple desde el punto de vista del manejo y el procesamiento de información. Además, los alumnos analizarán aparatos que les sean conocidos para preguntarse si están basados o no en computadoras. Estas actividades proponen la discusión grupal en clase, con la idea de poner de manifiesto los supuestos comunes acerca de qué cosa es una computadora, para luego poder reformularlos. También se buscará que se apropien del objeto, a través de actividades de desensamblado de computadoras para ver cómo son por dentro y qué partes las componen.



CAPÍTULO 7: LA MEMORIA Y LA UNIDAD CENTRAL DE PROCESAMIENTO

Continuando la línea del capítulo anterior, en este se estudian la memoria y la unidad central de procesamiento. Estos dos componentes juegan el rol más importante en una computadora. La unidad central de procesamiento tiene la capacidad de manipular datos de entrada y producir datos de salida, es decir, tiene la capacidad de computar. En la memoria, por su parte, se almacenan los datos.

Las actividades propuestas procuran que los estudiantes indaguen los diferentes tipos de memorias: rápidas o grandes, volátiles o permanentes. Además, y en relación con lo visto acerca de la representación de datos en el capítulo 5, los estudiantes aprenderán cómo se almacenan los datos en una memoria.

Para profundizar en el funcionamiento interno de una unidad central de procesamiento, se propone una actividad de juego de rol donde los alumnos construirán una máquina criptográfica. Esta actividad relaciona lo aprendido en los capítulos sobre programación y seguridad. Este capítulo tiene como objetivo que los estudiantes comprendan cómo la combinación de operaciones extremadamente simples sirve para realizar una operación más compleja, esta es la base de la computación.

Además de los temas abordados en este libro, también desarrollamos actividades sobre tipos de datos y redes de computadoras. Este contenido también es libre y gratuito y se encuentra disponible en el sitio web de Program.AR. Se ofrece a continuación un breve resumen de este material complementario:



ANEXO I: TIPOS DE DATOS

En español, tenemos palabras que nos permiten decir que un objeto es de cierto tipo. Decimos que el **9** es un número, que **Juan** es un nombre o que **rojo** es un color. Cuando programamos, podemos hacer lo mismo con las variables de nuestros programas. Por ejemplo, el nombre y el apellido de una persona pueden ser representados como texto, como **"Juan Pérez"**, mientras que la edad de una persona puede ser un número entero como 9, 12, o 25.

En estas actividades se trabaja sobre la noción de tipos de datos. En primer lugar, se pone el foco sobre tipos elementales, tales como Número o Texto. Luego, se trabaja sobre el tipo Lista, que permite representar una secuencia de valores ordenados. Finalmente, se presentan los tipos Pila y Cola, y se trabaja razonando sobre ellos.



ANEXO II: REDES

Desde la aparición de Internet y su uso intensivo para múltiples propósitos, las redes de computadoras están presentes en muchas situaciones de la vida cotidiana. Sin embargo, podemos preguntarnos, en general, en qué consisten, cómo funcionan y cuáles son los principios básicos que posibilitan su existencia. Quizás nos resulten habituales algunos términos como HTTP, IP, mail, servidor, nube, chat y en línea, entre otros, pero ¿cuánto sabemos de ellos?

En estas actividades se abordan estas preguntas y otras más que van surgiendo con el desarrollo de los temas. Se presentan conceptos relacionados con las redes de computadoras a través de actividades desenchufadas que permiten desarrollar y construir en el aula temas tales como comunicación cliente-servidor, comunicación punto a punto, protocolos, fragmentación de la información, empaquetado-desempaquetado, detección de errores, enlaces, traza de un paquete y medio físico de comunicación, entre otros.

Compilado de fichas para estudiantes

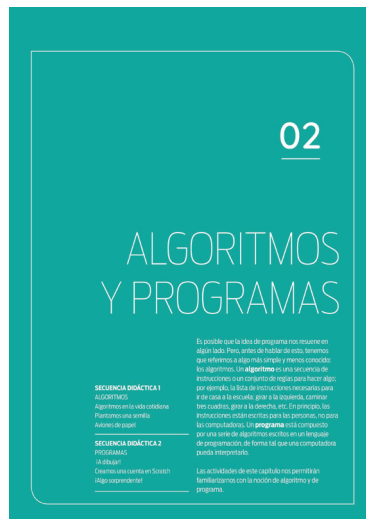
A fin de facilitar en un solo lugar todas las fichas para estudiantes, armamos un archivo en el que se las compila. El PDF es libre y gratuito, se encuentra en versión a color y en blanco y negro, y puede ser descargado desde el sitio web de Program.AR.

¿CÓMO ORGANIZAMOS ESTE MANUAL?

Cada uno de los capítulos del manual tiene las siguientes características:

APERTURA

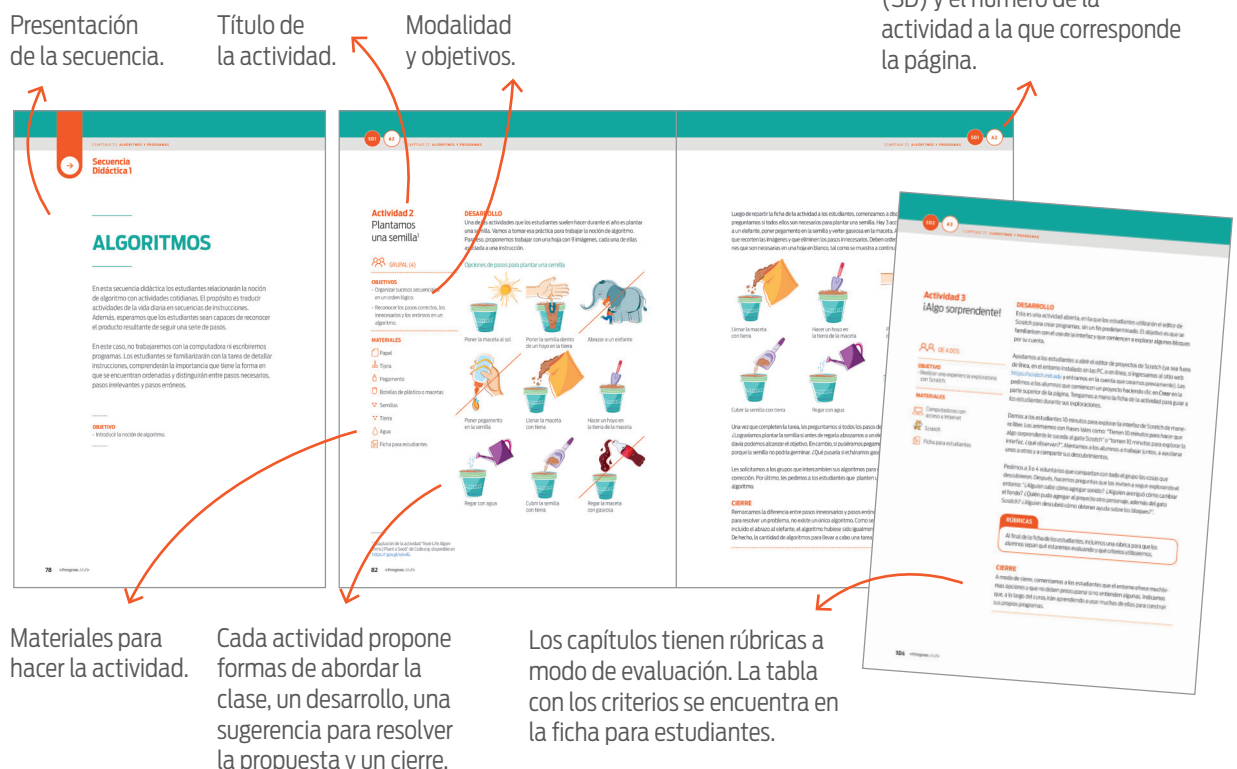
Cada apertura tiene un breve texto introductorio sobre los conceptos abordados en el capítulo y el índice de su contenido.



SECUENCIAS DIDÁCTICAS

Cada capítulo está integrado por secuencias didácticas que tienen una coherencia conceptual. Cada secuencia didáctica está conformada por actividades que pueden o no requerir el uso de una computadora.

Datos de orientación sobre el número de secuencia didáctica (SD) y el número de la actividad a la que corresponde la página.



FICHAS PARA ESTUDIANTES

Muchas actividades vienen acompañadas de una ficha para entregar a los estudiantes.



Datos que completa cada estudiante con el fin de facilitar la tarea de identificar a quién corresponde la ficha.

Datos a tener en cuenta o curiosidades relacionadas con el tema.

Datos sobre el capítulo, secuencia didáctica y actividad a la que corresponde cada ficha.

RÚBRICAS

Algunas fichas tienen rúbricas para que los estudiantes sepan con qué criterios serán evaluados por su docente.

UNIDAD 1

DEB

BUEN

4. Hay muchos modelos distintos de aviones para armar, y también existen valores del que acabamos de hacer. Podemos, por ejemplo, hacer aviones en las alas y agrupar una cola, también dibujar en una hoja instrucciones para hacer modificaciones y otros que se te ocurran y agrégales al algoritmo.

EVALUACIÓN

Estos son los ítems y la forma de calificar que tu docente considerará para evaluar tu desempeño.

CALIFICACIÓN	DEBILIDAD TRABAJAR MÁS	BUENO MUY BUENO	EXCELENTE
División de actividades en instrucciones	El estudiante tiene problemas al dividir una actividad en instrucciones simples.	El estudiante puede dividir una actividad en instrucciones simples, pero no puede explicarlas.	El estudiante puede dividir una actividad en instrucciones simples, pero no puede explicarlas.
Reconstrucción de actividades a partir de instrucciones	El estudiante no puede reconstruir la actividad a partir de instrucciones simples.	El estudiante reconstruye la actividad a partir de instrucciones simples.	El estudiante reconstruye la actividad a partir de instrucciones simples.
Reconstrucción de instrucciones sencillas, necesarias, breves y ordenadas	El estudiante confunde instrucciones sencillas, necesarias y breves.	El estudiante reconstruye la actividad a partir de instrucciones sencillas, necesarias y breves.	El estudiante reconstruye la actividad a partir de instrucciones sencillas, necesarias, breves y ordenadas.
Noción de algoritmo	El estudiante no comprende la noción de algoritmo.	El estudiante comprende la noción de algoritmo, pero no puede explicarla.	El estudiante comprende la noción de algoritmo, pero puede explicarla.

UNIDAD 1 | 10

GLOSARIO

GLOSARIO

El manual también incluye un glosario para el docente. Las definiciones no fueron pensadas para ser dadas a los estudiantes y se sugiere no presentarlas de forma descontextualizada.

REFERENCIAS

Bell, T., Fellows, M. y Witten, I. (2008). *Computer Science Unplugged*. Off-line activities and games for all ages. Recuperado de: <http://csunplugged.org/wp-content/uploads/2015/01/unplugged-book-v1.pdf>

Benotti, L., Martínez, M. C. y Schapachnik, F. (2014). "Engaging High School Students Using Chatbots". En: *Proceedings of the 19th ACM Annual Conference on Innovation and Technology in Computer Science Education*, pp. 63-68.

Benotti, L., Gómez M. y Martínez, M. C. (2017). UNC++Duino: A kit for learning to program robots in Python and C++ starting from blocks. En: *Robotics in education*, pp. 181-192.

Benotti, L., Martínez, M. C. y Schapachnik, F. (2017). "A Tool for Introducing Computer Science with Automatic Formative Assessment". En: *IEEE Transactions on Learning Technologies*.

Brusilovsky, P., Edwards, S., Kumar, A., Malmi, L., Benotti, L., Buck, D., Ihantola, P., Prince, R., Sirki, T., Sosnovsky, S., Urquiza, J., Vihavainen, A. y Wollowski, M. (2014). "Increasing Adoption of Smart Learning Content for Computer Science Education". En: *Proceedings of the Working Group Reports for the 19th ACM Annual Conference on Innovation and Technology in Computer Science Education*, pp. 31-57.

Davini, M. C. (2008). *Métodos de enseñanza. Didáctica general para maestros y profesores*. Buenos Aires, Santillana.

Levis, D. y Cabello, R. (2007). *Medios informáticos en la educación a principios del siglo XXI*. Buenos Aires, Prometeo.

Martínez, M. C., Gómez, M. y Benotti, L. (2015). "A Comparison of Preschool and Elementary School Children Learning Computer Science Concepts through a Multilanguage Robot Programming Platform". En: *Proceedings of the 20th ACM Annual Conference on Innovation and Technology in Computer Science Education*, pp. 159-164.

Martínez, M. C., Gómez, M., Moresi M. y Benotti, L. (2016). "Lessons Learned on Computer Science Teachers Professional Development". En: *Proceedings of the 21th ACM Annual Conference on Innovation and Technology in Computer Science Education*.

Newmann, F. M. (1996). *Authentic achievement: Restructuring schools for intellectual quality*. San Francisco, Wiley.

Rapenning, A., Webb, D. y Ioannidou, A. (2010). "Scalable Game Design and the Development of a Checklist for Getting Computational Thinking into Public Schools". En: *Proceedings of the 41st ACM technical symposium on Computer science education*, pp. 265-269.

Resnick, M. (2008). "Falling in Love with Seymour's Ideas". En: *American Educational Research Association annual conference*.

Resnick M. y Silverman, B. (2005). "Some reflections on Designing Construction Kits for Kids". En: *Proceedings of the 2005 Conference on Interaction Design and Children*, pp. 117-122.

Zapata Ros, M. (2015). "Pensamiento computacional: una nueva alfabetización digital". En: *Revista de educación a distancia*, 46 (4), pp. 1-47.

CIUDADANÍA DIGITAL Y SEGURIDAD

SECUENCIA DIDÁCTICA 1

REDES SOCIALES E IDENTIDAD DIGITAL

La publicación en redes sociales

Mentiras y problemas de ciberseguridad

Decálogo del buen uso de redes sociales
en niños y jóvenes

SECUENCIA DIDÁCTICA 2

AUTENTICACIÓN CON CONTRASEÑAS

¿Cuán segura es tu clave?

La fábrica de contraseñas seguras

SECUENCIA DIDÁCTICA 3

INFORMACIÓN SECRETA Y COMUNICACIÓN SEGURA

Mensajes privados, no tan privados

Enviamos mensajes un poco más

seguros

Protocolos para ocultar información

SECUENCIA DIDÁCTICA 4

MALWARE, SITIOS FALSOS Y OTRAS ACCIONES SOSPECHOSAS

Diferenciamos sitios inseguros

Resulta cada vez más importante saber usar una computadora, un teléfono inteligente o cualquier otro dispositivo computacional. Sin embargo, para ser un **ciudadano digital** consciente, no basta solo con aprender a usar *software*. Es necesario, además, entender cómo funciona y conocer las situaciones peligrosas e inesperadas que puedan presentarse. De este modo, se pueden evitar fraudes, infecciones y abusos, entre otros problemas que, en conjunto, se denominan de “seguridad informática”.

En este capítulo se presentan temas de seguridad informática asociados a las redes sociales, la identidad digital y la autenticación de los datos. Se ofrece una introducción a los métodos de comunicación segura y la encriptación de mensajes. Además, se habla de engaños comunes en el uso de Internet. Por último, se plantean consejos para analizar de forma crítica la seguridad a la hora de descargar y ejecutar programas.



Secuencia Didáctica 1

REDES SOCIALES E IDENTIDAD DIGITAL

La mayoría de las personas tiene alguna idea acerca de qué son las redes sociales y, aun si no son capaces de definirlas, es posible que formen parte de alguna. El concepto de red social no involucra necesariamente computadoras ni sistemas informáticos, pero en este capítulo se hará referencia solo a las de este tipo.

En esta secuencia didáctica se proponen actividades que permiten comprender algunas de las características de las redes sociales y la identidad digital. Al finalizarla, los estudiantes redactarán recomendaciones dirigidas a niños de su edad para que estén en condiciones de hacer más seguros sus perfiles en las redes sociales.

.....

OBJETIVOS

- Conocer las características generales de las redes sociales.
- Comprender algunos riesgos a los que se está expuesto cuando se forma parte de una red social.
- Proveer herramientas que permitan generar conciencia crítica sobre las acciones que corresponde tomar en diferentes situaciones de riesgo.

.....

Actividad 1

La publicación en redes sociales

GRUPAL (3)

OBJETIVOS

- Conocer el funcionamiento de las redes sociales y sus características.
- Identificar los principales riesgos que existen en las redes y distinguir información apropiada y no apropiada para compartir en ellas.

MATERIALES

- Ficha para estudiantes

DESARROLLO

La primera actividad de esta secuencia consiste en presentar el problema de la seguridad informática. Comenzamos con un debate general para introducir el tema, preguntando a los estudiantes: “¿Qué es una red social? ¿Cuáles conocen o usan? ¿Cómo es su perfil en la red? ¿Qué hacen en las redes?”.

Se espera que ofrezcan distintos tipos de respuestas. En relación con su perfil de usuario, es posible que digan que ingresan sus datos personales reales o que inventan algunos de ellos, ponen una imagen que los representa, etc.

¿Qué es la identidad digital?

Es el conjunto de informaciones publicadas en Internet sobre cada uno de nosotros. Esto incluye: datos personales, imágenes, videos, noticias, comentarios, gustos, amistades, etc. Todos estos elementos componen la imagen que los otros tienen de nosotros en la web y, de alguna forma, determinan nuestra reputación digital.

La identidad digital es dinámica y compleja. Puede construirse sin que se corresponda exactamente con la realidad. Sin embargo, lo que se hace bajo esa identidad tiene consecuencias en el mundo real y viceversa. Por eso es importante ser consciente de ella y cuidarla de forma responsable.



En cuanto a las redes sociales, los estudiantes pueden responder que son páginas o sitios web, aplicaciones, juegos, etc. Quizás hasta podrían surgir respuestas más técnicas, tales como que son programas o sistemas informáticos. Además, probablemente mencionen que a través de ellas comparten opiniones, experiencias, logros y fotos con amigos y familiares de modo público, o, incluso, juegan en red.

Los estudiantes pueden nombrar redes sociales tales como Facebook, Instagram, Snapchat, Google+ y Twitter, entre otras. Algunas no tan obvias son Youtube y WhatsApp. Mencionamos que, a través de estas redes, se comparten fotos, audios, videos, juegos, noticias, opiniones, enlaces, etc. En caso de que haya estudiantes que no hayan tenido experiencia previa, les mostramos los principales elementos de las redes sociales: qué tipo de información se comparte, dónde y cómo se accede a esa información y cómo se socializa, a fin de que puedan responder las preguntas planteadas en la ficha.

¿Qué entendemos por red social?

Un sistema informático programado por personas, que se ejecuta en un conjunto de computadoras en alguna parte del mundo. Al conectarse a alguna de las redes, se puede establecer contacto con otras personas, compartir información, intereses, actividades y juegos, entre otras cosas.

Es importante escuchar y registrar las experiencias que los estudiantes tengan para compartir, para recuperar genuinamente sus saberes previos y hacerlos participar activamente de los contenidos de la clase. Luego preguntamos: “¿Qué diferencias hay entre una red social y otra? ¿Qué es posible publicar o compartir en cada una de ellas?”.

Es posible que los alumnos respondan que algunas redes (por ejemplo, Instagram) están diseñadas para compartir o publicar fotos. Otras admiten texto, audio o videos. Sin embargo, las redes pueden cambiar con el tiempo y parecerse más o menos entre ellas. El nombre que se le da a la relación entre los usuarios también puede variar: *amigo/a*, *seguidor/a*, *contacto*, u otros.

Les pedimos a los estudiantes que formen grupos de 3 integrantes y les repartimos la ficha de actividades. La propuesta es que elijan 3 o 4 redes sociales y realicen un análisis comparativo según el tipo de información que se puede compartir en cada una de ellas. Por ejemplo, es posible completar la ficha indicando que en Facebook se puede compartir tanto texto o noticias, como videos y fotos en la parte del estado, mientras que en Instagram es más usual compartir fotos o videos y los textos deben ser cortos. WhatsApp se usa más para mensajería instantánea, aunque también permite compartir fotos, videos y audios.

No siempre las redes sociales son completamente específicas para un tipo de información; por ejemplo, Twitter fue creada para compartir mensajes cortos llamados *tuits* (adaptación del inglés, *tweet*, ‘pío’, es decir, la onomatopeya usada para imitar la voz de un pájaro), pero actualmente se comparten también imágenes o videos, noticias, conversaciones y mensajes, entre otro tipo de información.

Una diferencia importante entre Twitter y otras redes sociales es que toda la información que se publica en esta red es pública, a menos que se tenga cuidado en restringir el acceso. En cambio, la información publicada en otras redes sociales es, en general, solamente accesible a un grupo restringido de perso-

nas, a menos que se marque como pública. Aun así, existen detalles de privacidad un poco más sutiles. Por ejemplo, en algunas redes sociales, cuando se publica algo y un contacto permitido lo comenta, en general, los contactos de este último pueden ver la publicación aunque no sean contactos de quien la publicó inicialmente. Esto último también puede ocurrir al etiquetar a una persona en una publicación.

Otro punto a tratar en esta actividad corresponde al almacenamiento de la información. En la mayoría de las redes sociales, la información (fotos, videos, mensajes, etc.) queda almacenada en Internet en uno o más servidores y se puede acceder a ella a través de quien la compartió y otros usuarios, desde distintos dispositivos. En cambio, otras redes sociales, como Snapchat, normalmente eliminan contenido de sus servidores cada cierto tiempo. En WhatsApp los mensajes quedan guardados en el celular de los participantes de las conversaciones y, al cambiar de teléfono, desaparecen si no se realizan copias de seguridad. En Facebook, aunque se cambie el dispositivo de acceso, siempre será accesible todo el contenido que se haya creado.

Dinamismo de las redes

Hay redes de distinto tipo: generales (no están dirigidas a un tipo específico de usuario ni se refieren a un tópico concreto, sino que promueven la libre participación, centrándose en los contactos), profesionales, dedicadas a un tema, etc. Independientemente de esto, las redes sociales son dinámicas. Pueden cambiar en su funcionamiento, su propuesta o su forma de acceso. Por eso, recomendamos que estas actividades sean tomadas a modo de referencia y se las actualice en la medida en que sea necesario.

Proponemos a los estudiantes que investiguen dónde y por cuánto tiempo se almacena la información de las distintas redes sociales. Guiamos la investigación preguntando: “¿Se almacena en Internet o en los dispositivos de acceso tales como celulares, computadoras o *tablets*? ¿Desde qué dispositivos se puede acceder a la red social?”.

Buscamos que lleguen a la conclusión de que la información, en algunos casos, se almacena **localmente**, es decir, en el dispositivo que usan. En otros casos, se almacena **remotamente**, es decir, en los servidores de la red social en Internet. Resaltamos que, en la mayoría de los casos, ellos mismos son los dueños del contenido que comparten o publican. Esto les permite verlo, modificarlo y eliminarlo. Sin embargo, deben tener en cuenta que, en ciertas redes sociales, aun después de eliminada la información, alguien puede haber realizado una copia y conservarla.

En la siguiente tabla se muestran las características analizadas de cada una de las redes sociales que aparecen en la ficha para los estudiantes.

RED SOCIAL	¿QUÉ TIPO DE INFORMACIÓN SE PUEDE COMPARTIR?	¿DÓNDE SE GUARDA LA INFORMACIÓN QUE SE COMPARTE?	¿LA INFORMACIÓN ES PERMANENTE O TRANSITORIA?
Facebook	Texto, fotos, videos y enlaces a otros sitios de Internet.	En servidores de Facebook.	Permanente.
Twitter	Textos de 140 caracteres como máximo. En menor medida, fotos y videos.	En servidores de Twitter.	Permanente.
WhatsApp	Texto, audio, fotos y videos.	En los dispositivos de los que participan de la conversación.	Los mensajes desaparecen al cambiar de teléfono.
Snapchat	Texto, fotos y videos.	En los servidores de Snapchat.	Transitoria.
Instagram	Inicialmente, imágenes. En la actualidad, también admite videos cortos.	En los servidores de Facebook (ya que Instagram fue comprada por esta compañía).	Permanente, las imágenes. Transitorias, los videos.

CIERRE

Realizamos una breve reflexión acerca de la información que los estudiantes no compartirían en las redes sociales. Quizás expresen que no publicarían frases ofensivas, malas palabras, experiencias tristes o fotos vergonzosas, entre otras posibilidades. Plantearemos entonces la pregunta nexa para la siguiente actividad: “¿Qué problemas de seguridad les parece que pueden producirse a partir del uso de redes sociales?”.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

LA PUBLICACIÓN EN REDES SOCIALES



Seguramente formás parte de alguna red social. ¿Sabés quién puede mirar el contenido que compartís en ella y dónde se guarda esa información? Hoy vamos a trabajar sobre estos aspectos de las redes sociales.

1. Elegí cuatro redes sociales de la tabla y completá los casilleros.

RED SOCIAL	¿QUÉ TIPO DE INFORMACIÓN SE PUEDE COMPARTIR?	¿PARA QUÉ COMPARTIRÍAS INFORMACIÓN EN ESTA RED SOCIAL?	¿QUÉ INFORMACIÓN NO COMPARTIRÍAS EN ESTA RED SOCIAL?	¿DÓNDE SE GUARDA LA INFORMACIÓN QUE SE COMPARTE?	¿LA INFORMACIÓN ES PERMANENTE O TRANSITORIA (SE BORRA AUTOMÁTICAMENTE CADA CIERTO TIEMPO)?
Facebook					
Twitter					
WhatsApp					
Snapchat					
Instagram					

2. ¿Alguna vez intentaste borrar una foto compartida en alguna red social? ¿Pudiste?

3. ¿Qué pasa con los comentarios asociados a la foto borrada? ¿La foto realmente desaparece de Internet? ¿Qué ocurre si alguien la vio y bajó una copia a su computadora?

Actividad 2

Mentiras y problemas de ciberseguridad¹






GRUPAL (3)

OBJETIVOS

- Reflexionar sobre los conflictos y riesgos relacionados con el uso de las redes sociales por parte de niños y jóvenes.
- Adquirir estrategias para reaccionar ante alguna situación de cuidado en el uso de redes sociales.
- Abordar el tema de la seguridad informática desde una perspectiva interdisciplinaria que se aproxime a conceptos de las áreas de Ciencias Sociales y Tecnología.

MATERIALES

-  Noticias del diario sobre delitos relacionados con redes sociales
-  Ficha para estudiantes "Mentiras y problemas de ciberseguridad"
-  Ficha para estudiantes "Expertos en ciberseguridad"

.....

Esta actividad tiene dos fichas para los alumnos. En el desarrollo se indica el momento en que debe usarse cada una.

DESARROLLO

Proponemos adentrarnos en la temática de la seguridad informática a través de una noticia del diario. Es posible utilizar alguna de las que incluimos en esta actividad u otras más actuales. Pedimos a los estudiantes que se dispongan en grupos de tres integrantes, les entregamos las noticias y les damos 10 minutos para que las lean.

Luego distribuimos la primera ficha de actividades, "Mentiras y problemas de ciberseguridad", en la que se abordan los distintos tipos de información que se comparte en las redes sociales. Les pedimos que la completen.

Durante la puesta en común de las discusiones en los grupos, señalamos que es importante saber que determinada información no se debería compartir de manera pública en una red social (por ejemplo, datos personales que puedan ser sensibles, como la dirección, en ciertos casos la ubicación, contraseñas, nombres completos, etcétera). Tampoco es una buena idea compartir información acerca de cuándo se van de vacaciones o detalles sobre la vida íntima de otras personas. Por ejemplo, si alguien obtiene la contraseña y usuario de alguien más, podría usar esa cuenta y robar su identidad en la red social. Si alguien publica su ubicación, algún otro con malas intenciones podría saber sus movimientos y atacarlo, robarle, etc. Si se publica que se viaja por vacaciones, un potencial ladrón podría ver que la casa de quien viaja ha quedado desprotegida. El nombre completo puede utilizarse para buscar otro tipo de información, distinta de la publicada, usando otras técnicas o, incluso, servir para probarlo como contraseña.

También hay que prestar atención al modo en que se conforman las relaciones en las redes sociales, es decir, a quién se acepta como amigo, conocido o seguidor. Al respecto, una sugerencia puede ser no aceptar en una red social a alguien que no se conozca personalmente. Discutiremos con los grupos qué estrategias usan para decidir a quién aceptan.

Información pública o privada

La información se puede compartir de manera **pública** o **privada**. Al compartir información de manera pública, cualquier persona puede acceder a ella si busca lo suficiente. Un completo desconocido puede ver una foto compartida u otra información sensible. Por el contrario, si se comparte información de manera privada, será accesible solamente para un grupo reducido de miembros de la red social.

Algunas palabras que pueden llegar a surgir en ciertos casos son *cyberbullying* o *ciberacoso*, *grooming*, *sexting*. Se recomienda tratar el tema y que los mismos estudiantes lo investiguen.¹

Hay que hacerles notar a los estudiantes que una red social permite mentir y esconderse. La persona que usa una red social puede mentir acerca de la edad, el género o la apariencia. También puede hacerse pasar por otra persona; de hecho, un serio problema hoy en día tiene que ver con el **robo de identidad**. Este ocurre cuando una persona adquiere o utiliza información personal de otra sin estar autorizada, con la intención de cometer un fraude u otros delitos. Las dos formas más comunes en que esto sucede son obteniendo la contraseña del perfil de usuario de la víctima o creando una cuenta falsa con datos personales del usuario víctima, asumiendo su identidad.²

Es importante destacar en la conversación que, en caso de recibir un mensaje o invitación sospechosa, si no se sabe cómo proceder, lo recomendable es buscar ayuda de padres, docentes o alguna persona mayor responsable.

Todo lo propuesto hasta ahora pone de manifiesto que los usuarios de redes sociales dejan **rastros digitales** y, en muchos casos, exponen información sensible o privada.

Ciberacoso

Término que proviene del inglés *cyberbullying*. Es el uso de medios de comunicación digitales para acosar a una persona o grupo de personas, mediante ataques personales, amenazas, divulgación de información confidencial y/o falsa, entre otras estrategias.

Grooming

Se trata de una serie de conductas y acciones emprendidas deliberadamente por un adulto con el objetivo de ganarse la amistad de un menor de edad. Por lo general, crea una conexión emocional con el chico a fin de disminuir sus inhibiciones y poder abusar sexualmente de él.

Sexting

Se refiere al envío de fotos y videos de carácter sexual (erótico o pornográfico) por medio de dispositivos tales como celular, computadora, etc. La práctica del *sexting* implica riesgos psicológicos, legales e incluso de la integridad física de los participantes. Muchos de sus practicantes son menores de edad y no son conscientes de esos riesgos: es el deber de padres, madres y educadores advertírselos.

Rastros digitales

Se llama así a los datos y la información producidos por acciones y comunicaciones en la red, que pueden usarse para cometer delitos informáticos. Por ejemplo: "Me voy 15 días a Mar del Plata".

¹ Página recomendada para profundizar o investigar el tema: <https://www.argentinacibersegura.org>.

² Noticia sobre robo de identidad en González Pérez, L. (2 de junio de 2012). Redes sociales: crecen los casos de robo de identidad. *Clarín*. Obtenido de <https://goo.gl/mgAC8f>.

El uso de la computadora como medio de comunicación permite que cualquiera pueda ocultar su identidad; mentir sin ser descubierto es mucho más fácil de lo que sería posible en una conversación cara a cara. Internet posibilita hablar con gente cuya ubicación, género o edad se desconocen.

Algunos datos que se puede compartir con los estudiantes son los siguientes¹: 5000 casos de ciberacoso se denuncian cada año solamente en la Ciudad Autónoma de Buenos Aires; el 95% de las víctimas son menores. El 88% de los casos de delitos informáticos son realizados por adultos que inapropiadamente les comunican a los niños con quienes dialogan vía Internet que son especiales, sugieren ocultar información a sus padres y, en ocasiones, solicitan que alguno de los involucrados muestre sus partes íntimas o hacen referencia a partes íntimas y vínculos incómodos e inapropiados entre un adulto y un menor.

Una vez finalizada la discusión, se entrega la segunda ficha de la actividad y se les pide a los estudiantes que resuelvan las consignas.

CIERRE

Concluimos la actividad contándoles a los estudiantes que, para un mejor uso de las redes sociales y un mejor cuidado de la información personal que se comparte, se recomienda revisar las distintas opciones y configuraciones de privacidad de las redes sociales frecuentadas y realizar búsquedas sobre sí mismos en buscadores y redes sociales sin haber accedido con usuario y contraseña, para verificar qué información se expone públicamente y cuál no.

¹ Estos datos fueron extraídos de Listek, Vanesa. (6 de diciembre de 2016). Grooming: cada vez hay más casos de acoso a chicos en la Web. *La Nación*. Obtenido de: <https://goo.gl/tdTQkN>.

ANEXO: NOTICIAS SUGERIDAS PARA LA ACTIVIDAD



JUEVES
19 DE FEBRERO
2015

NOTICIAS EN LÍNEA

PARCIALMENTE NUBLADO
MÁX. 32° / MÍN. 24°

ACTUALIDAD POLÍTICA ECONOMÍA ESPECTÁCULOS **POLICIALES** EL MUNDO

DETENIDO POR CORRUPCIÓN DE MENORES A TRAVÉS DE RED SOCIAL

La Guardia Civil ha detenido a un hombre de 41 años por su implicación en delitos de acoso y amenazas a niñas con edades comprendidas entre los 12 y 15 años a través de una red social. Según se ha informado, hasta el momento los agentes han localizado a 24 víctimas.

La operación, bautizada Benhur, se inició cuando se tuvo conocimiento de la existencia de un perfil en una red social que se utilizaba para solicitar fotografías pertenecientes a menores con contenido inapropiado.

Su método consistía en crear un perfil en la red social haciéndose pasar por otra persona, para hacer amistad con jóvenes menores de edad, con las que iniciaba un intercambio de información con el objetivo de conseguir teléfonos y fotografías de partes íntimas para luego chantajearlas.

Noticia modificada para fines didácticos de *Globedia*, recuperada de: <https://goo.gl/x9qi2a>.



Noticia modificada con fines didácticos, recuperada de Uno: <https://goo.gl/pXcBxr>.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

MENTIRAS Y PROBLEMAS DE CIBERSEGURIDAD

Cuando usamos las redes sociales se presentan algunos peligros. ¿Cuáles son? ¿Qué tenemos que hacer en estos casos? Después de leer las noticias entregadas por tu docente, respondé las preguntas con los compañeros de tu grupo.

1. ¿Qué peligros relacionados con las redes sociales podés identificar en las noticias leídas?

2. ¿La misma persona puede tener más de una cuenta en una misma red social?

3. ¿Es más fácil mentir en persona o a través de una red social? ¿Por qué?

4. ¿Qué información es mejor no compartir en las redes sociales?



Ciberacoso (cyberbullying)

Es el uso de medios de comunicación digitales para acosar a una persona o un grupo, mediante ataques personales, amenazas, divulgación de información confidencial y/o falsa, entre otras estrategias.

Grooming

Se trata de una serie de conductas y acciones emprendidas deliberadamente por un adulto con el objetivo de ganarse la amistad de un menor de edad. Por lo general, crea una conexión emocional con el menor a fin de disminuir sus inhibiciones y poder abusar sexualmente de él.

TAREA PARA EL HOGAR

Revisá tus perfiles y la configuración de privacidad de las redes sociales que frecuentás y asegurate de que no compartís información privada de manera pública.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

EXPERTOS EN CIBERSEGURIDAD

¿Qué tipo de información es adecuado compartir en las redes sociales? ¿Cómo sabemos quiénes son realmente las personas que nos contactan a través de ellas? En esta actividad vas a trabajar sobre estos temas.



1. Un amigo de nuestra red social está por publicar varios posteos.
¿Podrías ayudarlo a no mandar información privada a demasiada gente?

Publicación | Foto/video | Video en vivo
Acontecimiento importante

¿Qué estás pensando?

Foto/video | Sentimiento/actividad | ...

Amigos | Publicar

¿Quién debería ver esto?

- Público**
Cualquiera dentro y fuera de Facebook
- Amigos** (checked)
Tus amigos de Facebook
- Amigos excepto...**
No mostrar a algunos amigos
- Solo yo**
Solo yo
- Amigos concretos**
Solo mostrar a algunos amigos

Ver todo

• ¿Cuál o cuáles de los siguientes estados no convendría publicar en una red social porque podría poner en riesgo la seguridad de quien los publica? Marcalos con una **X**.

- ☐ Estoy contento porque me fue bien en la prueba de Matemática.
- ☐ En mi cumple me regalaron un auto a control remoto :D.
- ☐ Mi perrito se perdió esta mañana. Si alguien lo ve, por favor, me avisa.
- ☐ Solo en casa y jugando a los videos toda la tarde y comiendo papas fritas.
- ☐ ¡Qué ganas de comerme una hamburguesa!

NOMBRE Y APELLIDO:

CURSO:

FECHA:

- ¿Por qué creés que el o los estados anteriores no deberían ser publicados?

- 2.** A nuestro amigo le llegó la siguiente notificación de alguien que no conoce:
“Juan Paco Pedro de la Mar te ha enviado una solicitud de amistad. Tienen 5 amigos en común”. ¿Qué debería hacer?

- ☐ Aceptar sin duda a este nuevo amigo, pues tienen amigos en común.
- ☐ Aceptarlo, pero luego preguntarle a algún amigo en común quién es esa persona.
- ☐ Rechazar la solicitud de amistad porque no lo conoce.
- ☐ Consultar con sus amigos en común quién es esa persona antes de tomar la decisión de aceptarlo o no.

- ¿Por qué?

- 3.** Si esta persona que envió la solicitud le mandara a nuestro amigo mensajes con preguntas personales del tipo: **“¿Cuántos años tenés? ¿Dónde vivís? ¿Cómo es tu casa? ¿Cómo se llaman tus papás?”**, ¿estaría bien responderle? ¿Por qué?

- 4.** En el recreo te encontrás con Tomi y Luli, dos estudiantes menores que vos, que acaban de abrir por primera vez una cuenta en una red social. Orgullosos, te muestran lo que publicaron. Ahora es tu turno de ayudarlos a crear perfiles seguros.

.....

1 Adaptación de la actividad "Follow the Digital Trail" de Common Sense Education, disponible en <https://goo.gl/BMQbhb>.

NOMBRE Y APELLIDO:

CURSO:

FECHA:



Nombre: Tomi



Dónde vive:
Argentina



Actividad favorita:
atrapar insectos



Mascota:
perro Tobi



Comida favorita:
milanesa con puré



Foto favorita



Nombre:
María Luciana Pérez



Dónde vive: Av.
San Martín
432 2º D, Merlo,
San Luis



Usuario:
Luli_Luna
Contraseña:
papafrita



Fecha de
nacimiento:
02/02/2005



Secreto: mis papás
pelean todo el
tiempo



Foto favorita

a. Seguí las pistas de Tomi y Luli y completá el cuadro.

	TOMI	LULI
De quién se conoce su nombre completo		
De quién se puede encontrar la casa		
De quién se sabe su fecha de nacimiento		
Quién comparte un secreto en Internet		
A quién podés describir mejor con su foto favorita		

b. Respondé las siguientes preguntas.

- ¿Sobre quién pueden encontrar más información? ¿Por qué? _____
- ¿Cuál de los usuarios tiene un “rastro o huella digital” más grande? _____
- ¿Hay alguna cosa publicada por Luli en Internet que podría llegar a darle problemas? _____
- Si es así, ¿cuál es y por qué? _____

c. A partir de las preguntas anteriores marcá la opción más adecuada para decir qué es una huella o rastro digital.

- ☐ Una pista que los animales dejan al caminar
- ☐ Zapatos que comprás por Internet
- ☐ Información personal en Internet

d. ¿Qué información es segura para compartir?

- ☐ Tu fecha de nacimiento
- ☐ Tu primer nombre o tu usuario
- ☐ Tu dirección

Actividad 3

Decálogo del buen uso de redes sociales



GRUPAL (4)

OBJETIVOS

- Orientar a los estudiantes respecto del buen uso de las redes sociales.
- Desarrollar habilidades de transferencia de conceptos de ciberseguridad a las situaciones de la vida cotidiana.
- Organizar la información utilizada a lo largo de la secuencia en un texto que permita sistematizarla.

MATERIALES



Lápiz



Papel



Ficha para estudiantes

DESARROLLO

Invitamos a la clase a que armen grupos de 3 o 4 estudiantes. Les contamos que las autoridades educativas de la zona están preocupadas porque cada año registran muchos casos de acoso cibernético entre los estudiantes. Nosotros sabemos que estos ataques cibernéticos podrían prevenirse si los usuarios estuvieran más informados sobre los datos que es pertinente publicar en las redes sociales y sobre aquellos que jamás deberían ser publicados.

Las autoridades consideran que este curso es experto en seguridad informática y por eso envió una carta con una misión muy particular: elaborar un decálogo con las claves a tener en cuenta para el buen uso de las redes sociales. Esta suerte de guía será fundamental para todos los estudiantes y estará pegada en todas las carteleras de los pasillos de las escuelas. Les aconsejamos considerar:

- Qué elementos deben estar en el perfil de una red social.
- Qué elementos no deben estar en el perfil de una red social.
- Consejos a la hora de aceptar amigos.

RÚBRICAS

Al final de la ficha de los estudiantes, incluimos una rúbrica para que los alumnos sepan qué estaremos evaluando y qué criterios utilizaremos.

CIERRE

Concluimos que los problemas de ciberseguridad están mucho más presentes de lo que solemos creer. Esta es una tendencia que tuvo un gran crecimiento en la última década y se encuentra en plena expansión. Por eso, es muy importante aprender a cuidarnos y facilitar ayuda a quienes no saben cómo enfrentar este tipo de problemática, a familiares o amigos o a personas de nuestro entorno en general.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

DECÁLOGO DEL BUEN USO DE REDES SOCIALES

La escuela ha enviado la siguiente carta a los estudiantes:

Estimados estudiantes expertos en seguridad informática:

Nos contactamos con ustedes para solicitarles su colaboración en la confección de un afiche para el buen uso de las redes sociales entre niños y jóvenes.

Como expertos en seguridad informática, sabrán que miles de niños y jóvenes sufren de acoso cibernético cada año. Estos ataques cibernéticos podrían prevenirse si los usuarios estuvieran mejor informados sobre qué datos es pertinente publicar en las redes sociales y cuáles jamás deberían publicarse.

Necesitamos un decálogo que brinde consejos útiles sobre el uso adecuado de las redes sociales. Sería ideal que pudieran acompañarlo con dibujos, para facilitar la lectura de los estudiantes más pequeños.

Atentamente,

Las autoridades escolares

1. Un decálogo es una lista de 10 principios, normas o reglas. El decálogo que vos y tus compañeros elaboren tiene que contener:

- Elementos que deben estar en el perfil del usuario.
- Elementos que NO deben estar en el perfil del usuario.
- Consejos a la hora de aceptar amigos.
- Dibujos que ayuden a la comprensión de cada uno de los puntos.

EVALUACIÓN

La tabla que se encuentra a continuación describe las tareas y la forma de calificar que tu docente considerará para evaluar cómo vos y tus compañeros resuelven las actividades. Es recomendable leerla antes de comenzar la escritura del decálogo.

CALIFICACIÓN	ES NECESARIO TRABAJAR MÁS	BUENO-MUY BUENO	EXCELENTE
Formato del decálogo	No genera 10 principios sobre el buen uso de redes sociales.	El grupo genera 10 principios o normas, pero la escritura no es clara.	El grupo genera 10 principios, normas o reglas sobre el buen uso de las redes sociales.
Ciberseguridad	No se recuperan conceptos y nociones de ciberseguridad.	Se recuperan algunos conceptos y nociones de ciberseguridad.	El grupo incorpora todos los conceptos y nociones de ciberseguridad aprendidos, tales como información de perfil, chat con extraños, aceptación de amigos, tipos de lenguaje en la comunicación y solicitudes apropiadas e inapropiadas.



Secuencia Didáctica 2

AUTENTICACIÓN CON CONTRASEÑAS

En el famoso cuento “Alí Babá y los cuarenta ladrones”, los malhechores usaban la frase secreta “ábrete sésamo” para abrir las puertas de la cueva donde guardaban los tesoros robados. A lo largo de la historia, distintos tipos de toques, apretones de mano y saludos secretos se han usado como contraseñas entre grupos de personas.

Hoy en día, prácticamente todos necesitamos contar con alguna suerte de palabra secreta o clave para acceder a otro tipo de tesoro: la información valiosa. Ya se trate de redes sociales, juegos o plataformas de compras en línea, lo cierto es que la mayoría de los servicios suele requerir un nombre de usuario y una contraseña.

Existen distintas formas de autenticación de una contraseña, pero el uso de las alfanuméricas es el mecanismo más difundido. En esta secuencia didáctica vamos a estudiar distintos factores que deben tenerse en cuenta para generar contraseñas lo más seguras posibles.

.....

OBJETIVOS

- Identificar cuáles son las características de las claves seguras.
- Crear claves seguras utilizando las características aprendidas.

.....

Actividad 1

¿Cuán segura es tu clave?

DE A DOS

OBJETIVOS

- Sensibilizar a los estudiantes a partir de un juego sobre hábitos de seguridad informática para elegir una contraseña.
- Reconocer algunas acciones de seguridad en el uso de contraseñas.

MATERIALES

-  Ficha para estudiantes

DESARROLLO

Arrancamos la actividad diciendo a la clase: "Levanten la mano quienes usan contraseñas, por ejemplo, en redes sociales, juegos o para bloquear dispositivos como el celular". Es probable que la mayoría lo haga. Si hay estudiantes que nunca crearon una contraseña, les pedimos que piensen una. Luego, les indicamos que levanten la mano si tienen algunos de los siguientes hábitos relacionados con la seguridad de las claves:

- | | |
|---|---|
| 1. Usás tu apodo como clave. | 5. Solo vos y tus padres o tutores conocen tu clave. |
| 2. Tu clave incluye texto, números y símbolos. | 6. Tu clave es una palabra que aparece en el diccionario. |
| 3. Tu clave tiene menos de 8 caracteres (letras, dígitos y otros símbolos). | 7. Te resulta difícil recordar tu clave. |
| 4. Dejás que tus amigos sepan tu clave. | 8. Tu clave es un número telefónico. |

Algunos puntos corresponden a prácticas seguras (2 y 5) y otros no (1, 3, 4, 6, 7 y 8). Preguntamos: "¿Qué diferencia una contraseña segura de otra que no lo es? ¿Qué tipo de contraseña imaginan que pueden ser más segura?". Es posible que respondan que las contraseñas que tienen al menos 8 caracteres y combinan números, letras en mayúscula y minúscula y otros símbolos son más seguras. Esto es así porque los programas de computación pueden combinar caracteres y hacer muchos intentos para adivinar una clave en segundos. Mientras más larga y compleja sea la clave, más difícil será para una máquina atacante adivinarla.

A continuación, les contamos el caso del abuelo don Braulio, quien ha recibido una tarjeta de débito electrónica por primera vez en su vida. Necesita usar la tarjeta, pero antes debe crear una contraseña. ¿Cómo lo podemos ayudar? Repartimos la ficha para estudiantes y les pedimos que la completen trabajando de a dos.

En la segunda consigna de la ficha, se sugiere que el usuario aplique una serie de reglas para crear una contraseña. La más adecuada de las opciones es la (ii) ya que cumple con todas las reglas: tiene dos mayúsculas, más letras que dígitos y tres símbolos (#@BelBob3r-2688).

CIERRE

Esta actividad revela que muchos de los usuarios, incluyendo adultos, tienen pocos saberes sobre cómo generar contraseñas seguras. Elegir como claves palabras que son fácilmente reproducibles o fáciles de adivinar no es un buen hábito de seguridad informática. Para concluir, sugerimos reflexionar sobre la diferencia de hábitos para crear contraseñas y analizar con toda la clase la temática de la seguridad en claves remarcando qué hábitos son seguros y cuáles no.

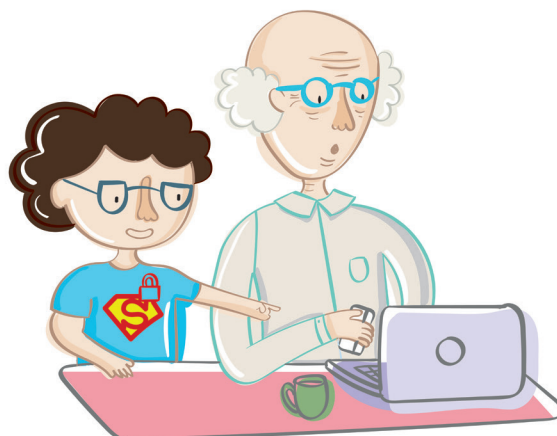
NOMBRE Y APELLIDO:

CURSO:

FECHA:

¿CUÁN SEGURA ES TU CLAVE?

El abuelo Braulio Martínez nació el 29/02/1950 y su DNI es 10.474.391. Recibió por correo una tarjeta para hacer los trámites del banco desde su casa, pero debe crear una contraseña y no entiende mucho de eso. ¡Ayúdalo a elaborarla!



1. Don Braulio probó crear las siguientes contraseñas, pero el sistema le dijo que no eran seguras:

Braulio	braulio290250	braulio1234	elprincipito
Braulio1950	1234	bmartinez	sodaestereo
10.474.391	juanaazurduy	Br@ul10	oidmortales

- ¿Por qué creen que son consideradas inseguras?

2. El banco le solicitó al abuelo que creara la contraseña aplicando estas reglas:

- La contraseña debe contener al menos dos letras mayúsculas.
- La contraseña debe tener más letras que dígitos.
- La contraseña debe contener al menos tres símbolos (que no sean letras ni números).

Mirá con tus compañeros las contraseñas propuestas e indicá cuál es la más adecuada.

- ☐ HloD@mb2953?
- ☐ #@BelBob3r-2688
- ☐ R5#X&v73r68?!
- ☐ *h9n3ytR33*
- ☐ BrauLio1966

- ¿Qué otras posibilidades de contraseña se te ocurren? Da 3 ejemplos.

Actividad 2

La fábrica de contraseñas seguras



DE A DOS

OBJETIVOS

- Reconocer claves seguras y no seguras.
- Crear claves seguras fáciles de recordar.

MATERIALES



Ficha para estudiantes

DESARROLLO

Iniciamos la actividad, preguntando: "Cuando ingresan a un sitio en línea que requiere una contraseña, ¿qué tipo de contraseña pueden pedirles para que esta sea segura?". Pueden responder que algunos sitios solo aceptan como contraseña aquellas que tienen al menos 8 caracteres y combinan números, letras en mayúscula y minúscula y otros símbolos.

Retomamos lo visto en la actividad anterior, en la que señalamos que los programas de computación pueden combinar caracteres y hacer muchos intentos para adivinar una clave en segundos. Mientras más larga y compleja sea la clave, más difícil será para una máquina atacante adivinarla.

Si la clave fuera de 3 caracteres y se usaran solo números, sería bastante fácil adivinarla. Bastaría con intentar mil veces. ¿Por qué? Porque desde el 000 al 999 hay 1000 claves posibles. Se sugiere ejemplificar esta situación usando claves de 1 y 2 dígitos ya que la operación que surge es la potenciación. Y para los casos de 1 y 2 dígitos es más simple observar que pasa de 10 a $10 \times 10 = 100$ claves posibles respectivamente.

A través de más ejemplos se puede notar que:

- Para claves de 4 dígitos hay $10^4 = 10.000$ posibles claves.
- Para claves de 5 dígitos hay $10^5 = 100.000$ posibles claves, y así sucesivamente.

Esto muestra que, aumentando el número de caracteres de la clave, crece exponencialmente la cantidad de intentos necesarios para adivinarla. La insistencia en el uso de símbolos, números y letras es principalmente para que la contraseña elegida sea más difícil de adivinar. Una clave que, además de números, incluya letras y símbolos aumentará mucho la cantidad de combinaciones que va a tener que probar el programa de quien esté tratando de descubrirla.

Usando letras en mayúscula, minúscula, dígitos y 5 símbolos en claves de 8 caracteres, la cantidad de claves posibles se eleva a $(27 + 27 + 10 + 5)^8 = 513.798.374.428.641$ (más de 500 billones). Usando solo letras en minúscula da $(27)^8 = 282.429.536.4816$ (algo más de 280.000 millones) claves y usando solamente dígitos, $(10)^8 = 100.000.000$ (100 millones). Muchas personas usan nombres, fechas, siglas u otro tipo de palabras como clave porque son más fáciles de recordar. Es importante que una clave sea fácil de recordar para evitar anotarla y que alguien la vea. Pero también es importante pensar en una contraseña que sea difícil de adivinar.

Teniendo en cuenta cómo las computadoras procesan la información para armar combinaciones y así descubrir una clave, entregamos la ficha a los estudiantes y les pedimos que formen parejas y la completen. En la ficha proponemos seguir los pasos de una receta para fabricar contraseñas seguras.

TAREA PARA EL HOGAR

Al final de la ficha, proponemos un juego optativo para que los estudiantes realicen en sus hogares: el ahorcado. La idea es que las primeras tres palabras se jueguen de forma clásica y luego se incorpore una variante. Esta cuarta palabra a adivinar se tratará de una palabra a la que se le aplicarán las mismas reglas que se utilizaron en la fábrica de contraseñas seguras.

¿Qué buscamos con esta variante? Que el otro participante del ahorcado no logre descifrar la palabra. Entonces, después de varios intentos fallidos, se le explicará que a la palabra se le aplicaron una serie de reglas que la vuelven más difícil de adivinar.

CIERRE

Para cerrar lo visto sobre seguridad de contraseñas, recomendamos las siguientes prácticas:

- No usar solamente letras o números.
 - No usar palabras reconocibles, tales como nombres propios, palabras del diccionario o términos de televisión, canciones o similares, aun si terminan con números.
 - No usar palabras en idiomas extranjeros populares.
 - No usar información personal como fechas, números de DNI o de teléfono.
 - No escribir contraseñas en papel como recordatorio.
-

LA FÁBRICA DE CONTRASEÑAS SEGURAS

Te presentamos una receta para fabricar contraseñas seguras. ¡Crea contraseñas que sean fáciles de recordar, pero difíciles de adivinar! Los pasos a seguir son:

1. Pensá una frase cualquiera, por ejemplo: "Somos lo que hacemos para cambiar lo que somos".
2. Anotá cada una de las iniciales de las palabras, una al lado de la otra. Siguiendo nuestro ejemplo, quedaría así: slqhpclqs.
3. Sustituí letras por números. Por ejemplo, la letra h por el número 4 y la q por el símbolo @. Ahora nuestra frase quedaría así: sl@4pcl@s.
4. Colocá en mayúscula al menos una letra, como, por ejemplo, la P. Esto nos da: sl@4Pcl@s

**¡ATENCIÓN!**

No escribas ni imprimas tus contraseñas en un papel.

1. Aplicá la receta (eligiendo dos o más sustituciones) a las siguientes frases:

"Susanita tiene un ratón, un ratón chiquitín, que come chocolate y turrón". _____

"Oíd, mortales, el grito sagrado: libertad, libertad, libertad". _____

"Cuando bailo un bailecito mi pañuelo es una pluma, va volando entre las nubes acariciando la luna". _____

2. ¿Cuáles son buenas prácticas del uso y la creación de contraseñas? Leé las oraciones y escribí SÍ o NO según corresponda en cada caso:

Cambiar tu contraseña si sospechás que alguien más entró en tu cuenta. ☐

Usar una palabra del diccionario como contraseña. ☐

Decir a tus padres tu contraseña. ☐

Si creés que alguna de tus contraseñas no es segura, cambiarla pronto. ☐

3. ¿Qué tenemos que hacer cuando vemos que alguien va a escribir su contraseña en una compu o un celular?

☐ Mirar hacia otro lado.

☐ Anotar la contraseña en un cuaderno o celular.

☐ Decirle tu propia contraseña para mostrar que son buenos amigos.

☐ Mirar de cerca y avisarle que no está ocultando la contraseña.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

TAREA PARA EL HOGAR: ¡A JUGAR AL AHORCADO!¹

Pedile a algún familiar que juegue con vos al ahorcado. En las primeras tres veces, buscá que adivine las palabras que aparecen en la primera columna de la tabla. A medida que va adivinando, completá las otras columnas de la tabla. ¡Atención! Tu familiar no debe ver las palabras hasta el final del juego.

PALABRA A ADIVINAR	¿ADIVINÓ?	SI GANÓ, ¿EN CUÁNTOS INTENTOS?	¿CUÁNTAS LETRAS ADIVINÓ?
Zombiz			
Tomate			
Rinoceronte			
3Er3i#u3			

Antes de pasar a la cuarta palabra del juego, decile que ahora se pueden poner expresiones con letras, números y símbolos en lugar de palabras.

En la tabla pusimos un ejemplo, 3Er3i#u3, que obtuvimos utilizando la fábrica de contraseñas seguras.

1. Elegimos la frase: "Por el río Paraná iba navegando un piojo".
2. Anotamos cada una de las iniciales de las palabras, una al lado de la otra: perpinup.
3. Sustituimos letras por números. Por ejemplo, la letra p por el número 3 y la n por el símbolo #. Ahora nuestra frase quedó así: 3er3i#u3.
4. Colocamos en mayúscula al menos una letra, en este ejemplo, la E. Esto nos da: 3Er3i#u3.

¡Vos podés usar otra frase! Dale varias oportunidades a tu familiar para que adivine. Si termina ahorcado, dale más chances. Al finalizar el juego, contale la importancia de elegir contraseñas seguras. Compartí con él las reglas aplicadas a la frase y explicale cómo funciona la fábrica de contraseñas seguras para que confirme que una buena contraseña es difícil de adivinar.

CONTRASEÑAS QUE NO

Una de las contraseñas más usadas es **qwerty**. Esta no es una contraseña segura. ¿Se te ocurre por qué tanta gente la usa? Ayuda: intentá escribirla en el teclado de tu computadora. Otras contraseñas muy inseguras son: **contraseña**, **1234**, **1111**, **123456**, **12345678**. Si una contraseña es muy usada, es probable que sea insegura.²



.....

¹ Ahorcado (juego), (s.f.). Wikipedia. Obtenido de <http://goo.gl/NkQE7a>.

² No vale copiar: las 25 contraseñas más usadas, (13 de enero de 2017). *Día a Día*. Obtenido de <http://goo.gl/cUnv5u>.



Secuencia Didáctica 3

INFORMACIÓN SECRETA Y COMUNICACIÓN SEGURA

Guardar o enviar información de manera secreta fue –y sigue siendo– un gran desafío en la historia de la humanidad. Se han creado muchos mecanismos para asegurar que nadie pueda leer o alterar la información original antes de que llegue a su destino. Uno de los métodos más usados es la **encriptación**. Se usa cuando queremos acceder a una red social, a un cajero bancario, cuando nos conectamos al wifi que tiene clave o enviamos mensajes de texto con el celular.

Un **sistema de encriptación** consiste en dos procedimientos: uno que se encarga de encriptar datos conocidos como **texto claro** (en inglés, *plaintext*) en textos que parecen sin sentido, es decir, **texto cifrado** o **codificado** (en inglés, *ciphertext*); y otro que se ocupa de descifrar el texto cifrado para obtener el mensaje original.

Al completar esta secuencia didáctica, los estudiantes van a conocer algunos métodos simples de encriptación y la forma en que pueden ser vulnerados. A su vez, podrán comprender la importancia de su uso en Internet y otras redes de computadoras.

.....

OBJETIVOS

- Comprender principios de encriptación.
- Mostrar que en Internet se usa encriptación.
- Comprender que un mensaje enviado por una red insegura puede ser observado por terceros.

.....

Actividad 1

Mensajes privados no tan privados

TODA LA CLASE

OBJETIVOS

- Conocer qué es un canal inseguro.
- Comprender que en Internet hay canales inseguros.

MATERIALES

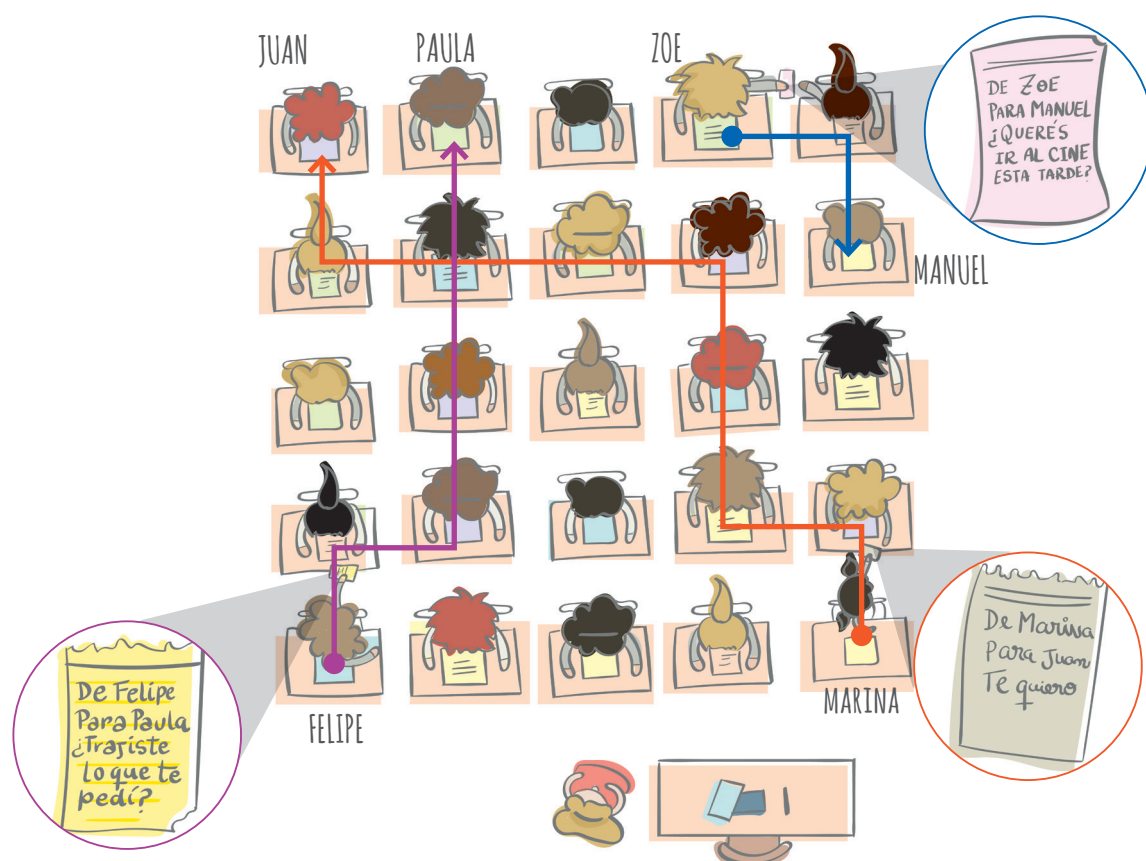
- Hojas de tamaño A6 o similar
- Ficha para estudiantes

DESARROLLO

Cuando enviamos un mensaje por chat o mail, hacemos comentarios en una red social o completamos nuestro usuario y contraseña en alguna pantalla de registro, esa información viaja a través de la red para llegar a destino. En algún momento nos da alguna clase de respuesta como “mensaje enviado”, “comentario guardado” o “acceso exitoso”.

En esta actividad vamos a mostrar el movimiento de la información a través de un **canal inseguro**. Iniciamos la actividad repartiendo la ficha a los estudiantes o escribiendo la consigna en el pizarrón. Varios estudiantes tienen que mandar, simultáneamente, mensajes secretos a algún compañero, con las siguientes condiciones:

- El mensaje debe ser privado.
- El mensaje debe tener escrito el nombre de quien lo manda y del destinatario.
- La hoja que contiene el mensaje no debe doblarse ni colocarse en un sobre.
- El mensaje deberá llegar a destino pasando de mano en mano por los compañeros que estén entre el remitente y el destinatario, pero sin que estos se levanten del asiento.



Al término del juego preguntamos a la clase si creen que algún compañero en el medio podría haber leído o modificado el contenido de algún mensaje. Explicamos que esto violaría la **privacidad** del mensaje, ya que dejaría de ser secreto.

Luego guiamos la conversación formulando preguntas tales como: “¿Llegaron los mensajes? ¿Cómo se puede saber que los mensajes que llegaron fueron enviados por la persona que dice el papel?”. Buscamos, de este modo, que concluyan que no pueden asegurarse del todo de que fue así.

Continuamos preguntando: “¿Cómo se puede saber que el mensaje no lo leyó alguien más en el camino?”. No se puede, ya que leer no deja rastros. “¿Cómo se puede asegurar que nadie, en el camino, cambió alguna parte del mensaje?”. Tampoco se puede estar seguro. Alguien podría haber roto la nota y creado una nueva con el mismo remitente y destinatario. O podría haber agregado o borrado una parte del mensaje. Por ejemplo, alguien podría cambiar una nota que dice “TE QUIERO” y escribir un “NO” al comienzo, de modo que llegaría a destino el mensaje “NO TE QUIERO”. Preguntamos: “¿Se les ocurre alguna idea para que el mensaje no pueda ser leído ni modificado?”. Aquí, por ejemplo, podrían surgir ideas como la de poner el mensaje en un sobre.

CIERRE

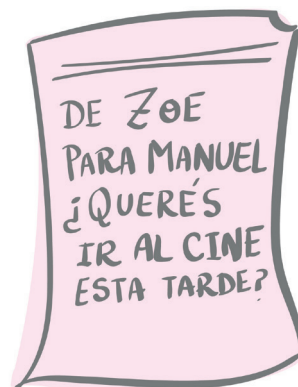
Para finalizar, señalamos que, en este caso, el **medio o canal de comunicación** son los mismos alumnos, y que se trata de un **canal inseguro** porque los mensajes viajan a través de las propias manos de quienes los hacen circular y estos pueden ver, entender y modificar los mensajes fácilmente. En el caso de Internet, ya sea desde un navegador o desde una aplicación, también existen canales inseguros. Para transformar un canal inseguro en un canal seguro se usa la encriptación. En la próxima actividad se aprenderán técnicas sencillas de encriptación.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

MENSAJES PRIVADOS NO TAN PRIVADOS



1. a. Escribí un mensaje privado para un compañero que esté ubicado lejos. El mensaje debe estar escrito en una hoja de papel con letra mayúscula, de imprenta y clara. Por ejemplo: "JUAN ME CONTÓ UN CHISTE EN EL RECREO" o "ESTA TARDE FESTEJO MI CUMPLE". Es importante que escribas en el mensaje el nombre del remitente (quién envía el mensaje) y el del destinatario (quién debe recibirlo). Atención: no vale doblar el mensaje ni ponerlo en un sobre.

b. Tenés que hacer llegar tu mensaje al destinatario haciendo que pase de mano en mano. ¡No vale levantarse del asiento ni caminar!

c. Cuando el mensaje haya llegado a tu compañero, respondé las siguientes preguntas.

● ¿Llegó el mensaje que enviaste?

● ¿Podés saber si el mensaje fue leído por alguien más en el camino?

● ¿Podés asegurar que nadie, en el camino, cambió alguna parte del mensaje?

● ¿Cómo podés hacer para asegurarte de que el mensaje no sea leído ni modificado en el camino?

Actividad 2

Enviando mensajes un poco más seguros



DE A DOS

OBJETIVOS

- Aprender estrategias simples para enviar mensajes secretos y saber cómo descifrarlos.
- Entender cómo funciona un ataque a un sistema de encriptación por fuerza bruta.

MATERIALES



Ficha para estudiantes

DESARROLLO¹

En esta actividad vamos a trabajar algunos algoritmos de encriptación que pertenecen a la clase **cifrados de sustitución simple**. Esta clase de técnicas de encriptación se llama *de sustitución* debido a que la parte central del proceso consiste en el reemplazo de una letra por otra. El primer método de encriptación que se plantea, conocido como **cifrado César** o **cifrado de desplazamiento**, itiene más de 2000 años! Su nombre se debe al líder político y militar romano Julio César, a quien se le atribuye su uso.

Reemplazos de cifrado César desplazando una letra

A	B	C	D	E	F	G	H	I	J	K	L	M	N
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	

Este es un ejemplo de texto al que se le aplicó esta técnica: IPMB NVÑEP. Todas las letras fueron desplazadas una posición hacia adelante en el alfabeto. Por lo tanto, el proceso para descifrar el mensaje consiste en que cada letra del mensaje secreto o encriptado sea reemplazada por la letra anterior de acuerdo al orden alfabético.

En el caso de la primera palabra del mensaje, hay que reemplazar la letra I por la H, la P por la O, la M por la L y la B por la A. Así se logra descifrar la primera palabra: "HOLA". Siguiendo el mismo proceso con la segunda palabra se obtiene el mensaje: "HOLA MUNDO". Si al desplazarnos se acaba el alfabeto, se vuelve a comenzar desde el principio, por lo que la Z da como resultado la A.

Una variante de la técnica presentada consiste en que se realice otra cantidad de desplazamientos (mayor que 1). A continuación se muestra cómo serían los reemplazos desplazando las letras dos, tres y diez posiciones.

¹Parte del texto de esta sección fue traducido y adaptado de: Newton, H. y Bell, T. (s.f.). "Coding - Encryption: Substitution Ciphers", Computer Science Education Research Group. New Zealand. Field Guide. Disponible en <https://goo.gl/DzCAk8>.

Reemplazos de cifrado César desplazando dos letras

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B

Reemplazos de cifrado César desplazando tres letras

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Reemplazos de cifrado César desplazando diez letras

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J

Para que dos personas puedan comunicarse con este cifrado, es necesario que ambas sepan cuál es el número de desplazamientos que tienen que hacer: una para encriptar desplazando letras hacia adelante y la otra para descifrar desplazando letras hacia atrás. En este caso, la cantidad de letras que deben moverse es una suerte de contraseña que hay que conocer tanto para cifrar como para descifrar.

Repartimos la ficha de la actividad a los alumnos y les solicitamos que se dispongan en parejas. Les explicamos en qué consiste el método de sustitución y les aclaramos que hay una breve descripción en la ficha. Les pedimos que resuelvan el primer punto, en el que deben cifrar algunos mensajes aplicando distintos desplazamientos.

ME GUSTA EL HELADO $\xrightarrow{1 \text{ letra}}$ NF HVTUB FM IFMBEP

QUIERO JUGAR AL FÚTBOL $\xrightarrow{3 \text{ letras}}$ TXLHUR MXJDU DÑ IXWERÑ

VAMOS AL PARQUE $\xrightarrow{2 \text{ letras}}$ XCÑQU CN RCTSWG

A continuación, les solicitamos que resuelvan el punto 2, en el que deben llevar a cabo el proceso inverso, es decir, descifrar mensajes encriptados.

GN EQOQEKÑKGOVQ GU RQFGT $\xrightarrow{2 \text{ letras}}$ EL CONOCIMIENTO ES PODER

WBNPT, BSHFÑUJÑB $\xrightarrow{1 \text{ letra}}$ VAMOS, ARGENTINA

ÑWMBRZDKB ÑC NRFÑBDRNY $\xrightarrow{10 \text{ letras}}$ ENCRYPTAR ES DIVERTIDO

Preguntamos: "¿Qué pasa si no sabemos u olvidamos la contraseña o la cantidad de desplazamientos que hay que realizar?". Señalamos que, teniendo en cuenta que el mensaje está encriptado usando el cifrado César, podemos desplazar una letra. En caso de obtener algo ilegible que no represente un mensaje, seguimos probando desplazar dos o más letras hasta obtener un mensaje claro. A este tipo de técnica de prueba se la llama **fuerza bruta**. Consiste en probar todas las posibles contraseñas; en este caso, todos los desplazamientos posibles.

Pedimos a los estudiantes que resuelvan el punto 3, en el que deben aplicar fuerza bruta para descifrar el mensaje. La cantidad de desplazamientos adecuada es 10. Una pista que se les puede dar a los estudiantes es hacerles notar que en el texto cifrado aparece BB, y que en español hay pocas combinaciones posibles de dos consonantes iguales seguidas en una misma sílaba: LL o RR. O sea, los únicos dos reemplazos que tiene sentido probar son 10 y 17.

VR ZÑBBY CÑ FYUFRY UYMY $\xrightarrow{17 \text{ letras}}$ MI PERRO SE VOLVIÓ LOCO

El último tema que se trabaja, en el punto 5 de la ficha, es un cifrado de sustitución simple mucho más general. Consiste en hacer una tabla o mapa en el que a cada letra del alfabeto se le asigne otra sin un desplazamiento fijo, sino usando reemplazos arbitrarios, distintos para cada letra. En la ficha se propone el siguiente esquema de reemplazos.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
H	B	Z	D	L	F	C	U	R	J	K	O	A	N	Ñ	M	P	Q	G	S	T	E	V	W	X	Y	I

Reemplazos arbitrarios para cifrar mensajes.

Los estudiantes deben cifrar mensajes usando la codificación propuesta.

ME GUSTA LA PIZZA \longrightarrow AL CESTH OH PRIIH

MI PAPÁ TIENE BARBA \longrightarrow AR PHPH TRNL BHGBH

CIERRE

Para cerrar la actividad, destacamos que, cuando se usa encriptación, si alguien intercepta el mensaje, no podrá entenderlo fácilmente. Además, comentamos que las técnicas que se vieron en esta actividad son muy básicas. Una computadora puede decodificarlas con velocidad. Por eso existen técnicas más complejas que no han sido quebradas, es decir, que no pueden ser decodificadas en un tiempo corto.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

ENVIAMOS MENSAJES UN POCO MÁS SEGUROS



Existe una técnica para enviar mensajes secretos muy conocida llamada de **cifrado César** o **cifrado de desplazamiento**. En esta técnica se sustituye una letra por otra según el desplazamiento elegido. Por ejemplo, podemos desplazar la letra una sola posición: la A se reemplaza por la B, la B por la C y así sucesivamente hasta llegar a reemplazar la Z por la A. Mirá cómo reemplazar cada letra del alfabeto según esta técnica.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A

Reemplazos de cifrado César desplazando la letra una posición.

Usando este desplazamiento de una posición, a partir del texto original "SOY NICO Y TE INVITO A MI CUMPLE" se consigue el texto codificado "TPZ ÑJDP Z UF JÑWJUP B NJ DVNOMF".

Con este método también se podrían hacer desplazamientos de más de una posición. Por ejemplo, moviendo la letra 3 posiciones se reemplazaría la A por la D, la B por la E, la C por la F, de modo que la tabla de reemplazos quedaría de esta manera:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Reemplazos de cifrado César desplazando una letra tres posiciones.

1. Codificá los siguientes mensajes usando esta técnica.

ME GUSTA EL HELADO

Usá desplazamiento de 1 letra.

QUIERO JUGAR AL FÚTBOL

Usá desplazamiento de 3 letras.

VAMOS AL PARQUE

Usá desplazamiento de 2 letras.

CIFRADO CÉSAR

El método de cifrado César o cifrado de desplazamiento, tiene más de 2000 años y su nombre se debe al líder político y militar romano Julio César, a quien se le atribuye su uso.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

2. Decodificá los siguientes mensajes. Recordá que para descifrar el desplazamiento se hace al revés.

GN EQQEKÑKGOVQ GU RQFGT

Usá desplazamiento de 2 letras.

WBNPT BSHFÑUJÑB

Usá desplazamiento de 1 letra.

ÑWMBRZDKB ÑC NRFÑBDRNY

Usá desplazamiento de 10 letras.

3. ¡Ahora no sabemos cuántas posiciones se desplazaron las letras en este cifrado! Intentá descifrar el siguiente mensaje. Una pista: en el mensaje encriptado aparece BB en lugar de dos consonantes que forman parte de una sílaba. ¡En español esto solo puede ser LL o RR!

VR ZÑBBY CÑ FYUFY UYMY

ATAQUE DE FUERZA BRUTA

En criptografía, se denomina ataque de fuerza bruta a la forma de recuperar una clave probando todas las combinaciones posibles hasta encontrar aquella que permite el acceso.

4. ¿Se te ocurre otra técnica de encriptación que no sea de desplazamiento? Proponela.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

5. Existen otras formas de cifrados de sustitución que no usan desplazamiento. Por ejemplo, se puede crear un mapa o tabla que asocie cada letra a otra (o incluso la misma) de una manera simple. Esta técnica resulta más difícil de adivinar. Veamos un ejemplo:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
H	B	Z	D	L	F	C	U	R	J	K	O	A	N	Ñ	M	P	Q	G	S	T	E	V	W	X	Y	I

Reemplazos arbitrarios para cifrar mensajes.

Podemos cifrar la frase: "ME GUSTA EL HELADO" de esta forma "AL CESTH LO ULOHDM".
¡Ahora es tu turno! Usá la tabla de sustitución para encriptar estas frases.

ME GUSTA LA PIZZA

MI PAPÁ TIENE BARBA

6. En este tipo de encriptación, la clave o contraseña que deberán tener quienes quieran usarla será toda la tabla de sustitución del alfabeto. Para vos, ¿será más fácil o más difícil aplicar **fuerza bruta** sin tener la tabla de sustitución?

Actividad 3

Protocolos para ocultar información¹



GRUPAL (6)

OBJETIVOS

- Presentar una técnica para ocultar información sin usar encriptación.
- Diferenciar este tipo de técnicas de los sistemas de encriptación.

MATERIALES

Papel

Lápiz

Un cuaderno

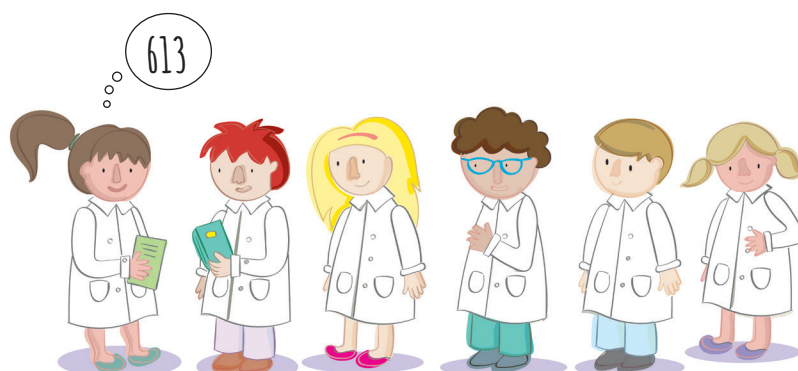
DESARROLLO

Las técnicas criptográficas permiten compartir información con otras personas manteniendo un nivel de privacidad importante. Para usarlas, es preciso acordar previamente entre quienes intercambian mensajes cómo se encriptan los datos. Si esta información llega a manos no deseadas, la privacidad de los datos se ve amenazada. En esta actividad vamos a plantear una situación donde se necesita compartir información pero no es posible usar encriptación. Esta técnica se conoce como **protocolo para ocultar información**. Estos protocolos intercambian información sin revelar datos privados a los participantes.

Proponemos realizar una actividad con un grupo de estudiantes que debe calcular la cantidad de hermanos de todos los integrantes del grupo sin que ninguno tenga que revelar a los otros cuántos hermanos tiene. También se puede trabajar con el promedio de altura, peso, edad, es decir, algún dato personal de tipo numérico.

Proponemos a la clase el siguiente desafío: “Queremos averiguar cuántos hermanos tienen en total los estudiantes de un grupo elegido al azar, pero sin que ninguno de ellos revele cuántos hermanos tiene. ¿Podemos obtener esta información? ¿Cómo?”. Escuchamos las sugerencias de la clase.

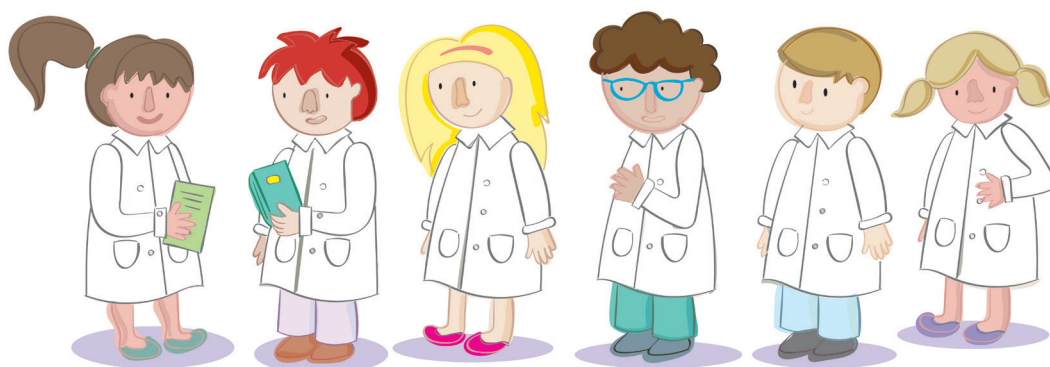
Elegimos a 6 estudiantes para ejemplificar la propuesta. Les pedimos que se pongan en fila, le damos un cuaderno y un lápiz al primero de ellos y le pedimos que elija secretamente un número cualquiera de 3 dígitos y lo escriba en la primera hoja del cuaderno. A modo de ejemplo, vamos a suponer que eligió el número 613.



Hacemos que el primer estudiante saque la primera página, sume su cantidad de hermanos al número secreto elegido y escriba el resultado en la segunda página del cuaderno. Si el primer estudiante tiene 3 hermanos, entonces escribe 616 en la segunda página del cuaderno. El cuaderno pasará al segundo estudiante del grupo, quien verá solamente el número allí escrito.

¹ Adaptación de la actividad "Sharing Secrets" de CS Unplugged, disponible en <https://goo.gl/s6EgNW>.

Luego, el segundo estudiante arranca la primera página, suma al número allí escrito su propia cantidad de hermanos, lo escribe en una nueva página y entrega el cuaderno al tercer estudiante. Este proceso se repite hasta que el cuaderno llegue al sexto y último estudiante.



Finalmente, devolvemos el cuaderno al primer estudiante. Este tiene que restar al número que está en el cuaderno el número que eligió en primera instancia. Siguiendo con el ejemplo, supongamos que hace la resta $624 - 613 = 11$. Este número es la suma de la cantidad de hermanos de todos los alumnos del grupo. Indicamos que, si todos destruyen los papeles que usaron, nadie podrá obtener las cantidades individuales de los otros.



CIERRE

Reflexionamos con los alumnos sobre la diferencia entre la técnica de este ejercicio y la encriptación de cifrado César. En ambos casos se consiguió que cierta información privada no fuera accesible para terceros. Sin embargo, la forma de conseguirlo fue distinta. Al usar encriptación, resultó indispensable que todos los que formaban parte del intercambio de información compartieran claves para poder cifrar y descifrar mensajes. En el caso de este ejercicio, se consiguió mantener información bajo custodia con una idea ingeniosa: toda esa información se ocultó detrás de una clave que conocía únicamente un participante del grupo.



Secuencia Didáctica 4

MALWARE, SITIOS FALSOS Y OTRAS ACCIONES SOSPECHOSAS

Como ciudadanos digitales, nos encontramos expuestos a distintos tipos de engaños. Por ejemplo, en los teléfonos inteligentes es muy común que aparezcan mensajes que recomiendan bajar cierto *software* para mejorar el rendimiento del dispositivo. ¿Cuál es el origen de este tipo de mensajes? ¿Qué pasa si instalamos los programas que recomiendan? Muchas trampas circulan a diario en el mundo virtual y es importante estar prevenidos.

El **malware** (del inglés *malicious software*, 'software malicioso') es un programa que ejecuta acciones malintencionadas en un dispositivo sin el consentimiento de su dueño. Existen muchas variantes: virus, gusanos y troyanos, entre otros. Difieren en el modo en que se produce la infección, se mantienen ocultos o se propagan. En esta secuencia didáctica vamos a trabajar sobre cómo detectar los engaños más típicos que se dan en el universo digital.

.....

OBJETIVOS

- Aprender qué es el malware.
- Comprender los riesgos de descargar *software* de sitios inseguros.
- Analizar de forma crítica un mensaje, una alerta, una publicidad o una aplicación.

.....

Actividad 1

Diferenciamos sitios inseguros



DE A DOS

OBJETIVOS

- Identificar un enlace perteneciente a una página oficial o un sitio fraudulento.
- Comprender la importancia de establecer conexiones seguras.
- Introducir el concepto de malware.

MATERIALES



Ficha para estudiantes

DESARROLLO

Jóvenes y adultos solemos recorrer grandes cantidades de sitios web cada vez que usamos dispositivos conectados a Internet. Las redes sociales, por ejemplo, permiten intercambiar gustos, opiniones, noticias y enlaces de forma masiva. Algunos de estos enlaces buscan estimular nuestra curiosidad para que ingresemos a sitios o aplicaciones web que están fuera de la red social.

Peguntamos a los estudiantes: "¿A quién le gusta visitar sitios que muestran trucos sobre cómo resolver juegos? ¿Quién descarga aplicaciones en el celular? ¿Cómo saben si una como Snapchat, Twitter o Instagram es segura?".

Es importante diferenciar entre una aplicación o un sitio seguro y otros que no lo son. Distinguir advertencias y notificaciones genuinas de otras falsas resulta vital para proteger nuestros datos y dispositivos. Así como Internet acerca el conocimiento y las culturas y facilita el intercambio de información, también permite que personas malintencionadas armen trampas para infectar nuestros dispositivos.

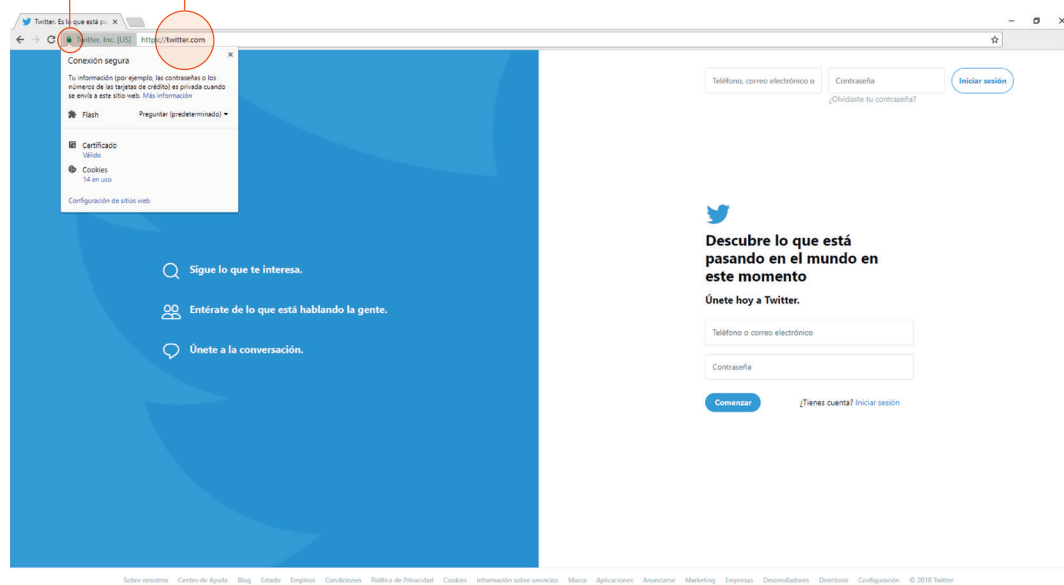
Entregamos la ficha de la actividad a los alumnos. Pedimos que encuentren las diferencias entre las dos capturas de pantalla de páginas que se muestran y que respondan las preguntas del punto 1.

Las diferencias importantes son tres:

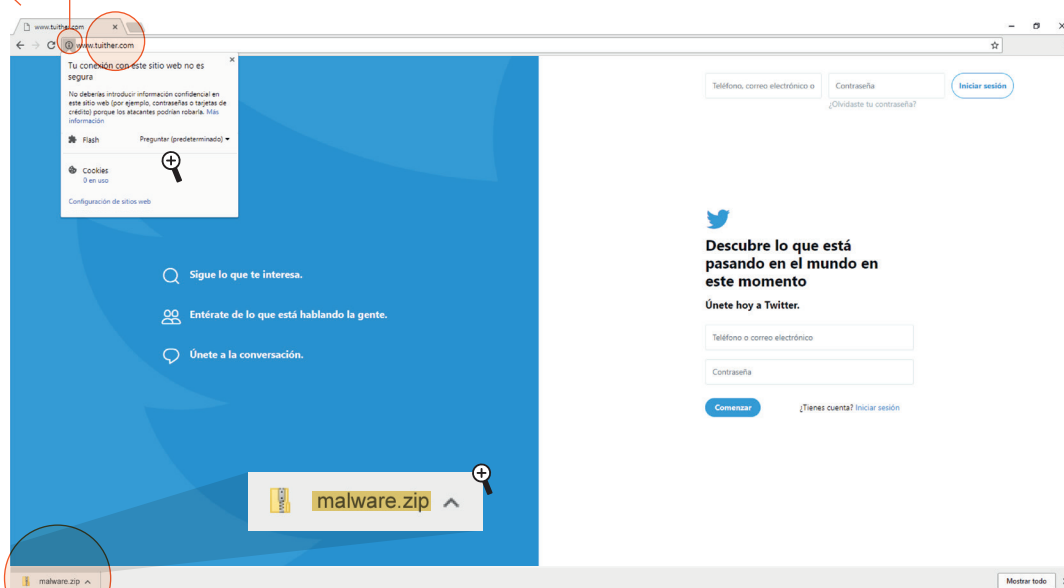
- Las direcciones de las páginas (URL) son distintas.
- En la segunda página aparece el símbolo de exclamación "!", en amarillo mientras que en la primera se muestra el candado verde.
- En la segunda se inició la descarga automática de un archivo malware.zip, mientras que en la primera, no.

Candado verde

Las direcciones de las páginas (URL) son distintas



Símbolo de exclamación "i"



Se inició la descarga automática de un archivo malware.zip

Las rutas o URLs son usadas para acceder a una página o recurso de Internet. Por ejemplo, <http://mail.google.com> es la dirección de Gmail, el servicio de correo electrónico de Google. En esta actividad nos vamos a centrar en aquellas que tienen el texto comprendido entre "http://" y la siguiente barra diagonal "/". Una URL consta de una secuencia de letras, números y algunos símbolos. Toda URL debe leerse tomando las palabras separadas por puntos de atrás para adelante. La última palabra antes de la barra "/" es lo que se conoce como el **dominio de nivel superior**. En el ejemplo, sería "com", que indica que se trata de un sitio comercial. Existen otros dominios de nivel superior como "edu", "ar", "org", etc.

com	Dominio de nivel superior
google	Dominio de nivel secundario (subdominio de .com)
mail	Dominio de tercer nivel (subdominio de google.com)

Jerarquía de un nombre de dominio.

Siguiendo la jerarquía de los componentes de una dirección web, al dominio de nivel superior lo sigue el dominio de nivel secundario. En el ejemplo que estamos analizando, el dominio de nivel secundario corresponde a la empresa Google.

Por último, hay un tercer nivel de dominio, el de mail. En este caso, se trata del servicio de correo electrónico que brinda la compañía. En las URLs `maps.google.com` y `translate.google.com`, "maps" y "translate" son los dominios de tercer nivel, y corresponden a los servicios de mapas y traducción brindados por la empresa Google. La importancia de prestar atención al orden radica en que la palabra ubicada más a la derecha tiene mayor importancia y se refiere a algo más grande.

Teniendo en cuenta esto, si quisiéramos acceder a un sitio o servicio que dependiera de la aplicación Snapchat y nos encontráramos estas dos URL: "algo.snapchat.com" y "snapchat.algo.com", ¿cuál deberíamos elegir? Deberíamos ingresar desde el primer sitio, porque indica que el dominio es snapchat.com. En cambio, el dominio "algo.com" es un dominio totalmente ajeno a Snapchat. Esta dirección indicaría que hay una empresa llamada "algo" que contiene la aplicación Snapchat, lo cual resulta al menos sospechoso.

En la siguiente figura se observa un candado verde que se encuentra a la izquierda de una dirección web. El candado indica que la comunicación es segura. Es decir, que usa encriptación. Al hacer clic allí aparecerá información importante sobre la comunicación y quien la verifica. Cuando no está presente el candado verde, suele aparecer solo el símbolo "i" o un candado rojo, lo cual indica que la conexión no es segura ya que no usa encriptación.



Dirección de un sitio bajo una conexión segura.

Muchas veces se envía o se recibe información privada que debería estar protegida. Tanto al realizar una compra por Internet como al escribir una contraseña de acceso a una red social, hay intercambio de datos entre el usuario y el sitio o la aplicación. El envío de información siempre debería estar protegido por mecanismos de encriptación. Además de prestar atención a la dirección URL del sitio al que se ingresa, es importante verificar que las conexiones sean seguras.

CUESTIÓN DE ÍCONOS

La forma de mostrar que la conexión de un sitio o aplicación no es segura varía de acuerdo al navegador que se utilice: Chrome, Explorer, Firefox, entre otros. En un navegador puede aparecer un mundo gris, un candado tachado o una cruz. La forma de representar la inseguridad puede variar ipero siempre se indica y es fundamental prestarle atención!

Por último, la tercera diferencia muestra un sitio que realiza una descarga automática de archivos. Esto podría ser peligroso, pues este tipo de acciones permite en muchos casos que las personas ejecuten un malware. Aun cuando la descarga parezca una foto, una canción, un juego o un video, es posible que se trate de un engaño y que al intentar abrir el archivo, este realice otras acciones. Estas acciones, que pueden ser silenciosas e imperceptibles para el usuario, van desde la pérdida o robo de información hasta la corrupción del sistema.

Les preguntamos a los estudiantes: “¿Es posible que una página no sea lo que dice ser? ¿Puede verse igual a la de una marca, aplicación o empresa conocida, pero ser falsa?”. La respuesta es afirmativa. Remarcamos que es importante revisar el dominio (la URL) y comprobar que la conexión sea segura y esté verificada.

“¿Es posible que una noticia o un mensaje que circule o llegue por Internet no sea cierto?”. Sí. Desde hace mucho tiempo existen, por ejemplo, cadenas de mensajes que prometen premios o que aparentan ser cadenas de recaudación serias para personas con enfermedades. Muchas de ellas pueden ser muy peligrosas porque pueden incluir enlaces que, al seguirlos, habilitan la descarga directa o indirecta de malware.

Un ejemplo concreto se muestra en la consigna 2 de la ficha para el alumno. Se trata de una cadena que circula por aplicaciones de chat y redes sociales en la que se promete premiar con juegos gratis a quienes la compartan y accedan al enlace <http://mijuegofavorito.freee.com>.



Mensajes engañosos que pueden corromper nuestros datos y dispositivos

Este enlace lleva a una página fraudulenta que descarga automáticamente un virus. Hacemos notar que se trata de un dominio muy sospechoso porque termina en *.freee.com* y no tiene relación con “mijuegofavorito”. ¿Es posible que un mensaje o mail no haya sido enviado por quien figura en el remitente? Sí. En muchos casos, esto se usa para hacer *phishing* y robar contraseñas, como en el ejemplo de la ficha.

Para reforzar lo trabajado en la actividad, escribiremos en el pizarrón las direcciones web **quinto-grado.colegio-argento.edu** y **colegio-argento.quinto-grado.edu**. Preguntaremos a los estudiantes cuál les parece que tiene más probabilidad de ser la página de quinto grado del colegio Argentó. Deberían notar que el nombre del colegio es más importante que el nombre del curso y, por lo tanto debería, ir a la derecha, con lo que se concluye que la primera URL debería ser la correcta.

CIERRE

Reflexionamos con los alumnos sobre la existencia de impostores que, para confundir a usuarios desprevenidos, usan sitios con nombres lo más parecidos posible a los de sitios genuinos. Repasamos que, siempre que se pueda, hay que revisar *(i)* a dónde apunta el enlace, botón, mensaje al que se está por hacer clic; *(ii)* si el sitio pide datos sensibles como contraseñas y tiene sentido que así sea, en cuyo caso solo se ingresará dicha información si la conexión es segura; *(iii)* si el sitio solicita descargar y/o ejecutar un archivo, acción que en general se debe evitar, salvo que sea un sitio de suficiente confianza y la situación lo amerite; y *(iv)* si lo que está diciendo, ofertando o proponiendo el sitio es genuino.

NOMBRE Y APELLIDO:

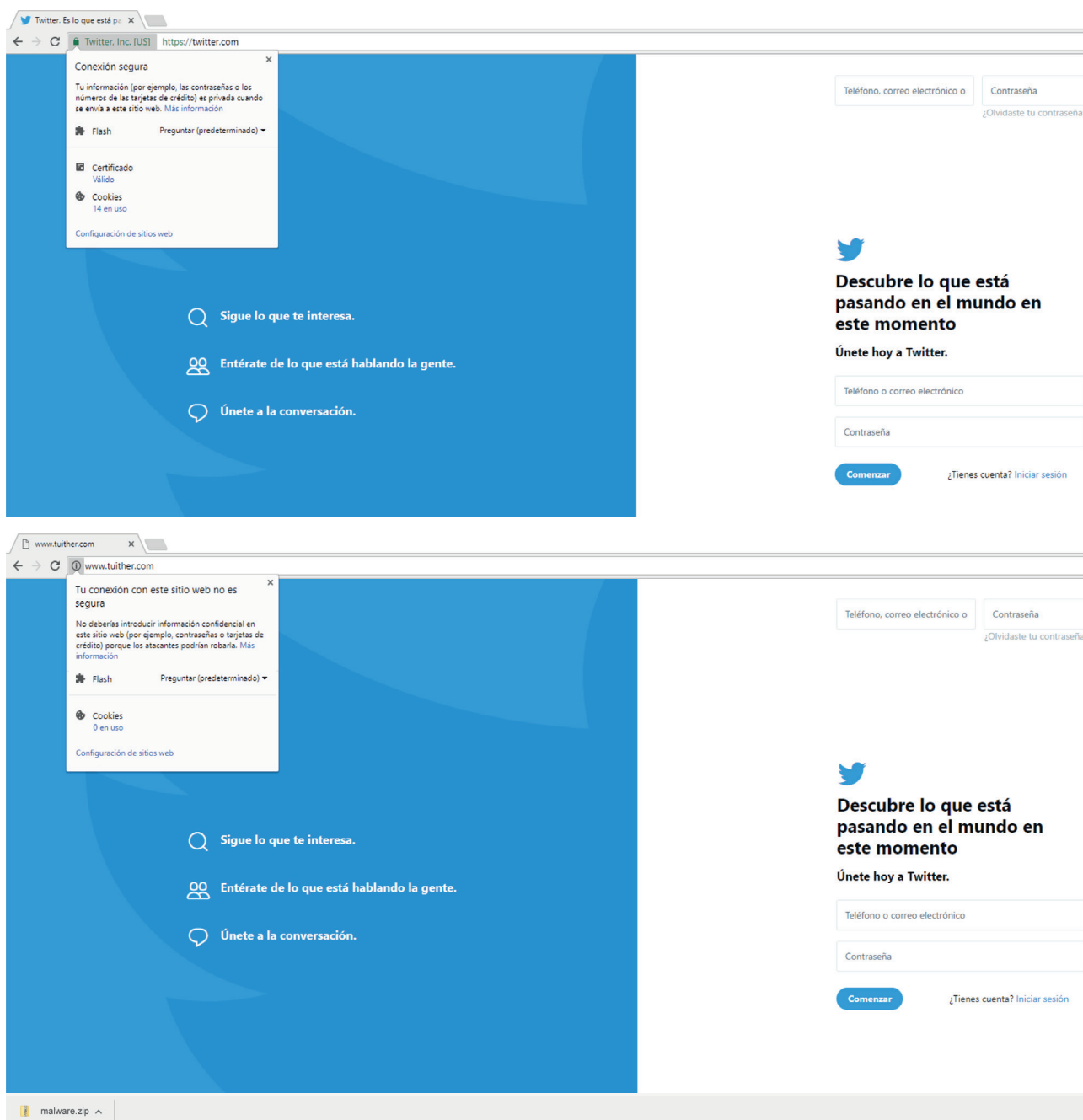
CURSO:

FECHA:

DIFERENCIAMOS SITIOS INSEGUROS

Al navegar por Internet, muchas veces encontramos ofertas, promociones y sugerencias que son sospechosamente ventajosas. ¿Cómo podemos saber si se trata de un mensaje auténtico o de un engaño? En esta actividad vamos a trabajar sobre algunas maneras de detectar trampas comunes.

1. Mencioná todas las diferencias que encuentres entre estas dos imágenes.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

- ¿Qué diferencias encontraste?

- ¿Son importantes estas diferencias? ¿Por qué?

- 2.** Suponé que sos parte de esta red social. ¿En cuál de las dos pantallas escribirías tu usuario y contraseña? ¿Por qué?

- 3.** Imaginá que la página no te pide descargar algo, pero que el candado verde no está presente; es decir, estás en la página correcta de ingreso donde se te pide usuario y contraseña, pero no está el candado verde. ¿Será peligroso ingresar el usuario y la contraseña?

ROBO DE IDENTIDAD

El robo de identidad en redes sociales es un delito que crece. Por ejemplo, un ataque posible consiste en desarrollar una página de acceso idéntica al sitio o aplicación que se pretende atacar con el objetivo de guardar los nombres de los usuarios y las contraseñas ingresados por personas que no se hayan percatado del fraude.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

4. Imaginá ahora que te llegan los siguientes mensajes al celular.



• ¿Serán confiables estos enlaces? ¿Por qué?

¿Qué es *phishing*?

Se conoce como **phishing** a un ataque que consiste en mandar un enlace para validación de una cuenta de usuario, aplicación o mail. Este enlace conduce a una página falsa donde se solicitan datos privados, como contraseñas, a fin de validar la cuenta, lo que permite el robo de esa cuenta.

ALGORITMOS Y PROGRAMAS

SECUENCIA DIDÁCTICA 1

ALGORITMOS

Algoritmos en la vida cotidiana

Plantamos una semilla

Aviones de papel

SECUENCIA DIDÁCTICA 2

PROGRAMAS

¡A dibujar!

Creamos una cuenta en Scratch

¡Algo sorprendente!

Es posible que la idea de programa nos resuene en algún lado. Pero, antes de hablar de esto, tenemos que referirnos a algo más simple y menos conocido: los algoritmos. Un **algoritmo** es una secuencia de instrucciones o un conjunto de reglas para hacer algo; por ejemplo, la lista de instrucciones necesarias para ir de casa a la escuela: girar a la izquierda, caminar tres cuadras, girar a la derecha, etc. En principio, las instrucciones están escritas para las personas, no para las computadoras. Un **programa** está compuesto por una serie de algoritmos escritos en un lenguaje de programación, de forma tal que una computadora pueda interpretarlo.

Las actividades de este capítulo nos permitirán familiarizarnos con la noción de algoritmo y de programa.



Secuencia Didáctica 1

ALGORITMOS

En esta secuencia didáctica los estudiantes relacionarán la noción de algoritmo con actividades cotidianas. El propósito es traducir actividades de la vida diaria en secuencias de instrucciones. Además, esperamos que los estudiantes sean capaces de reconocer el producto resultante de seguir una serie de pasos.

En este caso, no trabajaremos con la computadora ni escribiremos programas. Los estudiantes se familiarizarán con la tarea de detallar instrucciones, comprenderán la importancia que tiene la forma en que se encuentran ordenadas y distinguirán entre pasos necesarios, pasos irrelevantes y pasos erróneos.

.....

OBJETIVO

- Introducir la noción de algoritmo.

.....

Actividad 1

Algoritmos en la vida cotidiana¹



GRUPAL (4)

OBJETIVOS

- Mostrar formas en las que usamos algoritmos en nuestra vida cotidiana.
- Descomponer una actividad compleja en actividades más simples.
- Organizar sucesos secuenciales en un orden lógico.

MATERIALES

Papel

Lápiz

Ficha para estudiantes

DESARROLLO

Comenzamos la clase preguntando a los estudiantes: “¿Qué cosas hicieron a la mañana para prepararse para venir a la escuela?”. Algunas respuestas posibles son: “Me bañé”, “puse los útiles en la mochila”, “fui a la parada de colectivo”, entre otras. A medida que los alumnos vayan respondiendo, escribimos las respuestas en el pizarrón y les colocamos números para indicar un orden lógico en que estas acciones sucederían. Si los estudiantes nos dan respuestas desordenadas, iniciamos una discusión para consensuar un orden.

Esta es una buena oportunidad para señalar que, en algunos casos, el orden es importante, mientras que en otros no lo es. Por ejemplo, una persona podría bañarse primero y luego desayunar, y otra podría hacerlo al revés, lo que no implicaría un problema de orden. Sin embargo, no sucede lo mismo si consideramos las acciones de vestirse y de ir a la parada del colectivo; aquí el orden de estas dos acciones es relevante.

A continuación, hablaremos sobre la forma en que hacemos las cosas todos los días y cómo podemos realizar una descripción paso a paso del modo en que las llevamos a cabo. Por ejemplo, los alumnos pueden describir acciones tan simples como preparar el desayuno, cepillarse los dientes o ponerse los zapatos.

Repartimos la ficha de la actividad a los alumnos y les solicitamos que formen grupos de cuatro integrantes. En la ficha se muestra como ejemplo un algoritmo para ponernos las zapatillas.

Algoritmo para ponernos las zapatillas

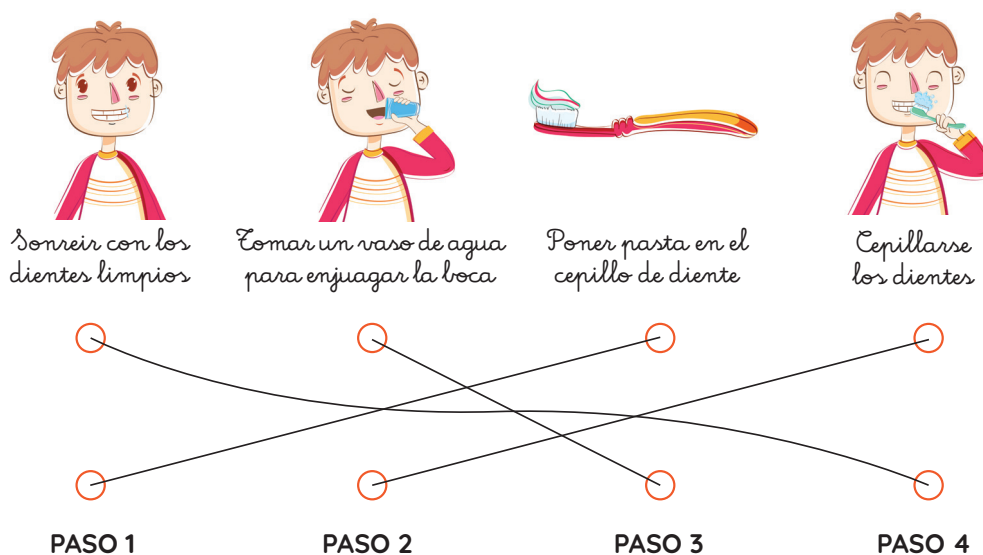


La primera consigna es pedir a cada equipo que elija una tarea cotidiana y enumere los pasos necesarios para llevarla a cabo. Luego, le pedimos a cada grupo que exponga los pasos para realizar la tarea elegida, pero sin mencionar a los demás a qué actividad corresponde. El resto de la clase debe adivinar cuál fue la actividad que eligió cada grupo.

¹ Adaptación de la actividad “Daily Algorithms” de Code.org, disponible en <https://goo.gl/8G9cU1>.

En la segunda consigna de la ficha, los estudiantes deben reconocer acciones a través de imágenes que presentan instrucciones desordenadas de una tarea. En este caso, se trata de un chico que se lava los dientes. Después de escribir la acción representada por cada imagen, deben conectar mediante líneas la imagen con el orden que le corresponda, de modo que el algoritmo para lavarse los dientes siga una secuencia lógica.

Algoritmo para lavarse los dientes



CIERRE

Escribimos en el pizarrón la palabra **algoritmo**. Luego, les explicamos a los alumnos que un algoritmo es una secuencia ordenada de instrucciones para alcanzar un objetivo. Mostramos, entonces, que ellos han escrito algoritmos para llevar a cabo actividades cotidianas. Aquí es importante hacerles notar que, en algunos casos, el orden de las instrucciones de los algoritmos puede cambiar sin que esto afecte el resultado, mientras que en otros esto no es posible.

NOMBRE Y APELLIDO:

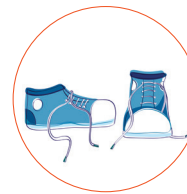
CURSO:

FECHA:

ALGORITMOS DE LA VIDA COTIDIANA

Para completar alguna tarea, muchas veces seguimos instrucciones paso a paso. Estas instrucciones, cuando están ordenadas, son algoritmos. Mirá, por ejemplo, un algoritmo para ponernos las zapatillas, que se encuentra a la derecha.

1. Elegí alguna tarea que te resulte familiar y escribí un algoritmo para llevarla a cabo.



Buscamos las
zapatillas



Metemos los
pies dentro de
las zapatillas



Nos atamos
los cordones

2. Las imágenes están desordenadas. Describí qué pasos sigue el chico en cada una. Después, uní estas imágenes con una línea según el orden que corresponda. Tenés que conseguir un algoritmo para lavarse los dientes. La primera ya está hecha a modo de ejemplo.



Sonreír con los
dientes limpios



PASO 1



PASO 2



PASO 3



PASO 4

La palabra **algoritmo** es una combinación de la palabra latina *algorismus* (por Al-Khwarizmi, un matemático persa del siglo IX) y la palabra griega *arithmos*, que significa 'número'.

Actividad 2

Plantamos una semilla¹



GRUPAL (4)

OBJETIVOS

- Organizar sucesos secuenciales en un orden lógico.
- Reconocer los pasos correctos, los innecesarios y los erróneos en un algoritmo.

MATERIALES

- Papel
- Tijera
- Pegamento
- Botellas de plástico o macetas
- Semillas
- Tierra
- Agua
- Ficha para estudiantes

DESARROLLO

Una de las actividades que los estudiantes suelen hacer durante el año es plantar una semilla. Vamos a tomar esa práctica para trabajar la noción de algoritmo. Para eso, proponemos trabajar con una hoja con 9 imágenes, cada una de ellas asociada a una instrucción.

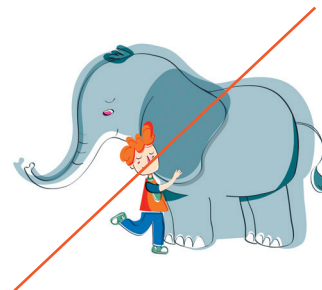
Opciones de pasos para plantar una semilla



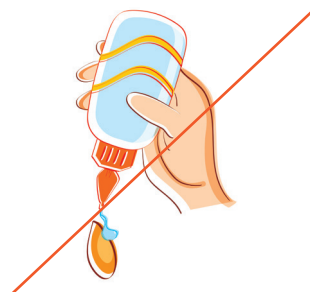
Poner la maceta al sol



Poner la semilla dentro de un hoyo en la tierra



Abrazar a un elefante



Poner pegamento en la semilla



Llenar la maceta con tierra



Hacer un hoyo en la tierra de la maceta



Regar con agua



Cubrir la semilla con tierra



Regar la maceta con gaseosa

¹ Adaptación de la actividad "Real-Life Algorithms | Plant a Seed" de Code.org, disponible en <https://goo.gl/xj4viG>.

Luego de repartir la ficha de la actividad a los estudiantes, comenzamos a discutir los distintos pasos y les preguntamos si todos ellos son necesarios para plantar una semilla. Hay 3 acciones que no lo son: abrazar a un elefante, poner pegamento en la semilla y verter gaseosa en la maceta. A continuación, les pedimos que recorten las imágenes y que eliminen los pasos innecesarios. Deben ordenar y pegar las 6 instrucciones que son necesarias en una hoja en blanco, tal como se muestra a continuación.



Llenar la maceta con tierra



Hacer un hoyo en la tierra de la maceta



Poner la semilla dentro de un hoyo en la tierra



Cubrir la semilla con tierra



Regar con agua



Poner la maceta al sol

Una vez que completen la tarea, les preguntamos si todos los pasos descartados son del mismo tipo. ¿Lograríamos plantar la semilla si antes de regarla abrazamos a un elefante? Aunque es innecesario, todavía podemos alcanzar el objetivo. En cambio, si pusiéramos pegamento en la semilla, la tarea fracasaría porque la semilla no podría germinar. ¿Qué pasaría si echáramos gaseosa en una maceta?

Les solicitamos a los grupos que intercambien sus algoritmos para ver si alguno tiene errores y requiere corrección. Por último, les pedimos a los estudiantes que planten una semilla siguiendo los pasos del algoritmo.

CIERRE

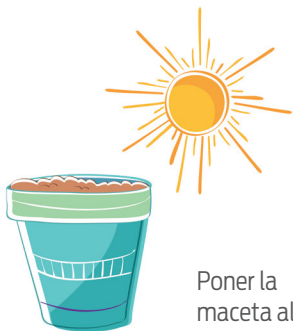
Remarcamos la diferencia entre pasos innecesarios y pasos erróneos. Por otro lado, comentamos que, para resolver un problema, no existe un único algoritmo. Como se vio en el ejercicio, si hubiésemos incluido el abrazo al elefante, el algoritmo hubiese sido igualmente correcto para plantar la semilla. De hecho, la cantidad de algoritmos para llevar a cabo una tarea es infinita.

PLANTAMOS UNA SEMILLA



¡Creemos un algoritmo para plantar una semilla!
Recortá las imágenes por la línea punteada.

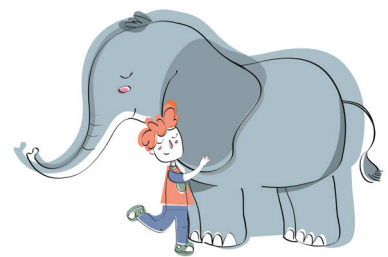
1. Trabajá junto con tus compañeros para elegir, ordenar correctamente y pegar en una hoja los pasos necesarios para conseguir un algoritmo para plantar una semilla.
2. ¿Qué pasos usaste para lograr el objetivo de plantar la semilla? ¿Alguno de los pasos de la hoja es innecesario? ¿Hay alguno que sea incorrecto?
3. Intercambiá tu solución con la de un compañero. ¿Hicieron el mismo algoritmo? ¿Encontrás algún error?



Poner la
maceta al sol



Poner la
semilla dentro
de un hoyo
en la tierra



Abrazar a
un elefante



Poner
pegamento
en la semilla



Llenar la
maceta
con tierra



Hacer un hoyo
en la tierra de
la maceta



Regar
con agua



Cubrir la
semilla
con tierra



Regar la
maceta
con gaseosa

Actividad 3

Aviones de papel¹



INDIVIDUAL

OBJETIVOS

- Reforzar la importancia del orden en la ejecución de instrucciones.
- Diferenciar entre acciones necesarias, opcionales y erróneas.

MATERIALES



Papel



Tijera



Pegamento



Clips

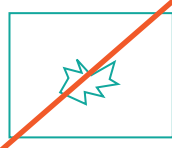


Ficha para estudiantes

DESARROLLO

Comenzamos por repartir a los estudiantes la ficha de la actividad, en la que se ven pasos desordenados para armar un avión de papel.

Opciones de pasos para hacer un avión de papel



Cortar el centro del papel



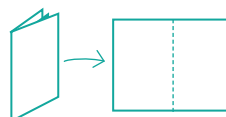
Doblar el papel
por la mitad



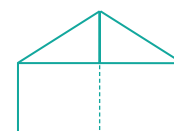
Hacer un bollo
con el papel



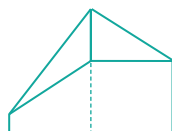
Agregar clip para darle mayor peso en la nariz



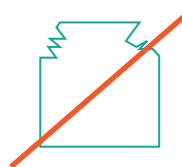
Desdoblar el papel



Doblar la esquina
al centro (der.)



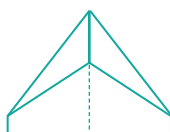
Doblar la esquina
al centro (izq.)



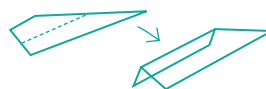
Romper bordes del papel



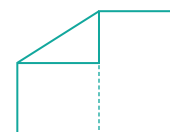
Doblar el papel
por la mitad



Doblar la esquina
al centro (der.)



Doblar hacia abajo
para formar alas
(ambos lados)



Doblar la esquina
al centro (izq.)

Hay 12 pasos, de los cuales 9 están relacionados con la actividad y 3 son innecesarios. Pedimos a los estudiantes que recorten los pasos para hacer un avión y los peguen en una hoja en blanco en un orden lógico, de forma tal de construir un algoritmo para llevar a cabo la tarea.

¹Adaptación de la actividad “Real-Life Algorithms | Paper Airplanes” de Code.org, disponible en <https://goo.gl/TtWKzx>.

Preguntamos a la clase: “¿Usaron todos los pasos?”. Si no lo hicieron, les consultamos: “¿Por qué?”. Tenemos que guiar la discusión para que los estudiantes puedan distinguir entre pasos optativos (innecesarios) y pasos erróneos. Se espera que puedan identificar que el paso de agregar un clip para dar mayor peso a la nariz del avión puede usarse, pero no es un paso necesario: aun cuando no se use, se obtendrá un avión de papel. Luego, les preguntamos: “¿Se les ocurren otros ejemplos en los que determinadas instrucciones sean optativas?”. Por otro lado, también deberían notar que cortar el centro del papel, hacer con él un bollo y romper sus bordes son instrucciones erróneas.

Les preguntamos si los pasos podrían realizarse en otro orden y si aun así conseguiríamos el mismo avión. Por ejemplo, ¿es necesario doblar primero la esquina izquierda, o se podría haber empezado por la esquina derecha? Entonces, ¿hay un único algoritmo para armar un avión de papel?

Les pedimos que intercambien sus propuestas y se fijen si identifican errores en los algoritmos recibidos. Si es así, ¿cuáles son estos errores? ¿El algoritmo del compañero es el mismo que ellos habían propuesto? Si la respuesta es no, les preguntamos si ese algoritmo, aunque diferente, sirve para armar un avión de papel. Finalmente, les pedimos que armen el avión siguiendo las instrucciones del algoritmo recibido y, si las normas del colegio lo permiten, los hagan volar por el aula.

RÚBRICAS

Al final de la ficha de los estudiantes, incluimos una rúbrica para que los alumnos sepan qué estaremos evaluando y qué criterios utilizaremos.

Como esta ficha tiene dos páginas y en la primera se les pide que recorten unas figuras, recomendamos fotocopiar o imprimir esta actividad en dos hojas diferentes. De esta forma, los estudiantes evitarán cortar parte de la tabla de la rúbrica.

CIERRE

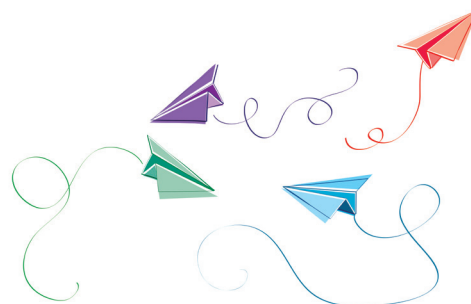
A modo de cierre, reflexionamos con los estudiantes sobre la posibilidad de modificar el orden de ciertos pasos sin alterar el resultado final. Por otro lado, es importante mencionar que hay ciertas acciones que pueden estar o no, y aun así, se alcanzaría el objetivo propuesto. En el ejemplo, la acción de agregar un clip para dar mayor peso a la nariz consigue un avión con mayor peso en la punta, pero si no se usa esta acción, de todas formas se obtiene un avión de papel.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

AVIONES DE PAPEL



¡Ahora vas a aprender a armar aviones de papel!

Para arrancar, recortá los 12 pasos desordenados de la imagen.

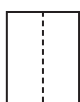
1. Ordená los recortes que se encuentran al pie de la página y pegalos en una hoja de modo que sirva de instructivo para hacer un avión.
2. Conversá con tus compañeros: ¿usaste todos los pasos? ¿Hay pasos que podrían no estar y aun así sería posible armar el avión? ¿Hay pasos que definitivamente no haya que usar?
3. A veces puede haber más de un algoritmo para la misma actividad. El orden de algunos pasos se puede cambiar sin alterar el resultado final. Usá las letras en las imágenes que siguen para crear dos algoritmos diferentes para hacer el mismo avión de papel.



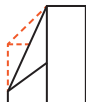
A



B



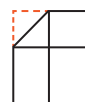
C



D



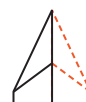
E



F



G



H

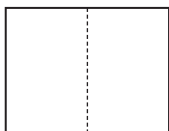
ALGORITMO 1



ALGORITMO 2



Cortar el centro del papel



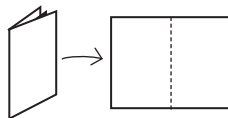
Doblar el papel por la mitad



Hacer un bollo con el papel



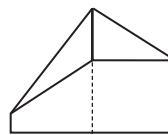
Agregar clip para mayor peso en la nariz



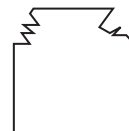
Desdoblar el papel



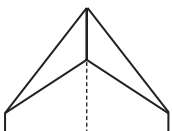
Doblar el papel por la mitad



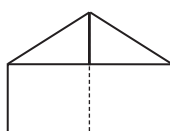
Doblar la esquina al centro (izq.)



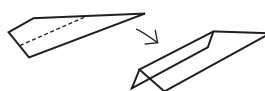
Romper bordes del papel



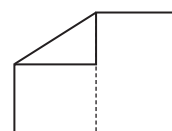
Doblar la esquina al centro (der.)



Doblar la esquina al centro (der.)



Doblar hacia abajo para formar alas (ambos lados)



Doblar la esquina al centro (izq.)

NOMBRE Y APELLIDO:

CURSO:

FECHA:

4. Hay muchos modelos distintos de aviones para armar, y también existen variantes del que acabamos de hacer. Podemos, por ejemplo, hacerle alerones en las alas o agregarle una cola. Escribí o dibujá en una hoja instrucciones para estas modificaciones y otras que se te ocurran y agregalas al algoritmo.

EVALUACIÓN

Estas son las tareas y la forma de calificar que tu docente considerará para evaluar tu desempeño.

CALIFICACIÓN	ES NECESARIO TRABAJAR MÁS	BUENO-MUY BUENO	EXCELENTE
División de actividades en instrucciones	El estudiante tiene problemas al dividir una actividad compleja en instrucciones simples.	El estudiante puede dividir una actividad compleja en instrucciones simples, pero incluye instrucciones innecesarias, erróneas o de muy diferente relevancia.	El estudiante puede dividir una actividad compleja en instrucciones simples, todas relevantes para realizar la tarea.
Reconstrucción de actividades a partir de instrucciones	El estudiante no puede reconocer la actividad resultante de ejecutar una serie de instrucciones simples.	El estudiante reconoce a veces la tarea resultante de llevar a cabo una serie de instrucciones simples.	El estudiante reconoce fácilmente la actividad resultante de llevar a cabo una serie de instrucciones simples.
Reconocimiento de instrucciones necesarias, innecesarias y erróneas	El estudiante confunde instrucciones necesarias, innecesarias y erróneas.	El estudiante reconoce la diferencia entre instrucciones necesarias, innecesarias y erróneas, pero comete errores al identificarlas en una tarea.	El estudiante distingue sin errores la diferencia entre instrucciones necesarias, innecesarias y erróneas.
Noción de algoritmo	El estudiante no comprende la noción de algoritmo.	El estudiante comprende la noción de algoritmo, pero tiene problemas al escribir algoritmos para tareas de la vida diaria.	El estudiante comprende la noción de algoritmo y puede describir fácilmente actividades de la vida diaria en forma de algoritmos.



Secuencia Didáctica 2

PROGRAMAS

Solemos expresar los algoritmos de forma tal que a las personas les resulten sencillos de leer y comprender. No existe un lenguaje único para hacerlo: se pueden usar dibujos o el idioma español, entre otros. Los lenguajes suelen ser muy ricos y expresivos: por ejemplo, dan la posibilidad de usar recursos tales como metáforas u onomatopeyas. Además, muchas veces el uso de palabras polisémicas da lugar a diversas interpretaciones: por ejemplo, al leer la frase “me voy a anotar en una carrera”, ¿sabemos si se refiere a una carrera universitaria o a una prueba de atletismo?

Al trabajar con sistemas digitales, la polisemia se vuelve un problema. Las computadoras siguen indicaciones al pie de la letra, pero no tienen la posibilidad de dilucidar el sentido de una frase por el contexto ni de interpretar el tono con el que fue dicha. Por eso, para darles órdenes, usamos **lenguajes de programación**, que usan instrucciones que siguen una sintaxis muy rígida y cuya semántica es inequívoca.

En esta secuencia didáctica pasaremos de la idea de algoritmo a la de **programa**. Comenzaremos trabajando con una actividad sin computadora. Luego, crearemos una cuenta de Scratch. Finalmente, haremos una experiencia lúdica en un entorno de programación.

.....

OBJETIVOS

- Exponer la dificultad de traducir problemas reales a programas.
- Evidenciar la necesidad de lenguajes formales para trabajar con computadoras.
- Poner en práctica la comunicación de ideas a través de códigos y símbolos.

.....

Actividad 1

¡A dibujar!¹



DE A DOS

OBJETIVOS

- Diferenciar las nociones de algoritmo y programa.
- Escribir algoritmos y codificarlos en un lenguaje de programación simbólico.
- Ejecutar los programas y verificar si tienen errores.

MATERIALES

Lápiz

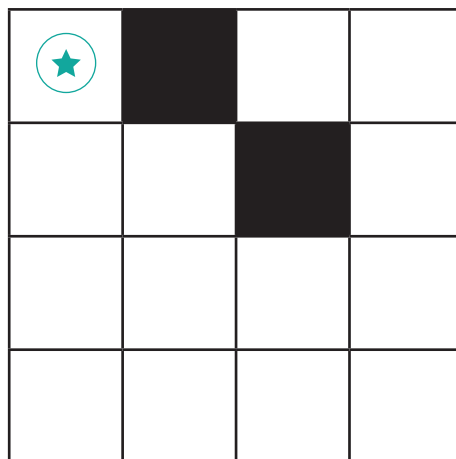
Papel cuadriculado

Ficha para estudiantes

DESARROLLO

En esta actividad pasaremos de la idea de **algoritmo** a la de **programa** y explicaremos las diferencias entre ambas. A lo largo de su desarrollo, los estudiantes deben dar instrucciones para que la “máquina de dibujar” haga dibujos sobre cuadrículas de 4 x 4. En primer lugar, expresarán algoritmos usando un lenguaje coloquial. Luego lo harán mediante un conjunto acotado de instrucciones precisas definidas en un inequívoco **lenguaje de programación**.

Para iniciar la clase, dibujamos en el pizarrón una imagen como esta.



Luego explicaremos a los estudiantes las reglas de la máquina de dibujar.

REGLAS DE LA MÁQUINA DE DIBUJAR

- La máquina puede pintar un cuadrado por vez.
- Solo puede moverse de un cuadrado a otro contiguo: arriba, abajo, izquierda o derecha.
- Siempre comienza desde el cuadrado ubicado en la esquina superior izquierda, marcado con una estrella.

Una vez que los estudiantes hayan comprendido las reglas, les pedimos que, de a dos, piensen y escriban algoritmos para que la máquina de dibujar realice el dibujo (esto es, pintar las dos casillas que se muestran en la imagen).

¹ Adaptación de la actividad : “Graph Paper Programming” de Code.org, disponible en <https://goo.gl/K5GUWf>.

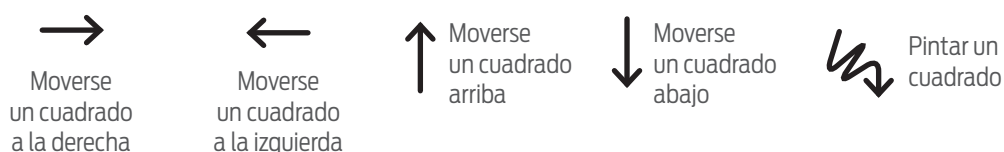
Cuando los grupos hayan escrito sus algoritmos, les solicitamos que los lean al resto de la clase. Es probable que surjan diversas respuestas con distinto vocabulario. Por ejemplo, alguna pareja de alumnos podría proponer las instrucciones “mover el lápiz a la derecha, pintar el cuadrado, mover el lápiz a la derecha, moverlo para abajo y pintar el cuadrado”. Otra podría plantear “ir a la derecha, rellenar de negro, ir a la derecha otra vez, ir para abajo y volver a rellenar de negro”. Es interesante escribir en el pizarrón propuestas que hayan sido formuladas usando distintas expresiones o palabras.

Al escribir las frases ponemos número a las instrucciones de cada propuesta. Si hay algoritmos que no son correctos –es decir, que no consiguen que la máquina pinte las casillas adecuadas–, discutimos con los alumnos los errores y los corregimos entre todos. Además, si los algoritmos proponen distintos recorridos para la máquina de dibujar, podemos aprovechar la oportunidad para remarcar que existen distintos algoritmos para alcanzar un objetivo y que no necesariamente uno es mejor que otro.

Preguntamos a la clase: “¿Qué diferencia hay entre las propuestas que están dando? No todas están expresadas del mismo modo. Si tuvieran que darle instrucciones a una computadora para que hiciera los dibujos, ¿los podría hacer de la forma en la que indicaron las propuestas? ¿Por qué?”.

Guiamos la discusión de forma tal de poner de manifiesto que, en general, una máquina no es capaz de interpretar instrucciones expresadas en lenguaje natural (como el español). Las computadoras hacen exactamente lo que les pedimos, pero solo comprenden una serie acotada de instrucciones muy precisas, que en su conjunto forman un lenguaje de programación. En el caso de la máquina de dibujar, usaremos un lenguaje de programación que consta de las 5 instrucciones que se muestran a continuación.

CONJUNTO DE INSTRUCCIONES PARA LA MÁQUINA DE DIBUJAR

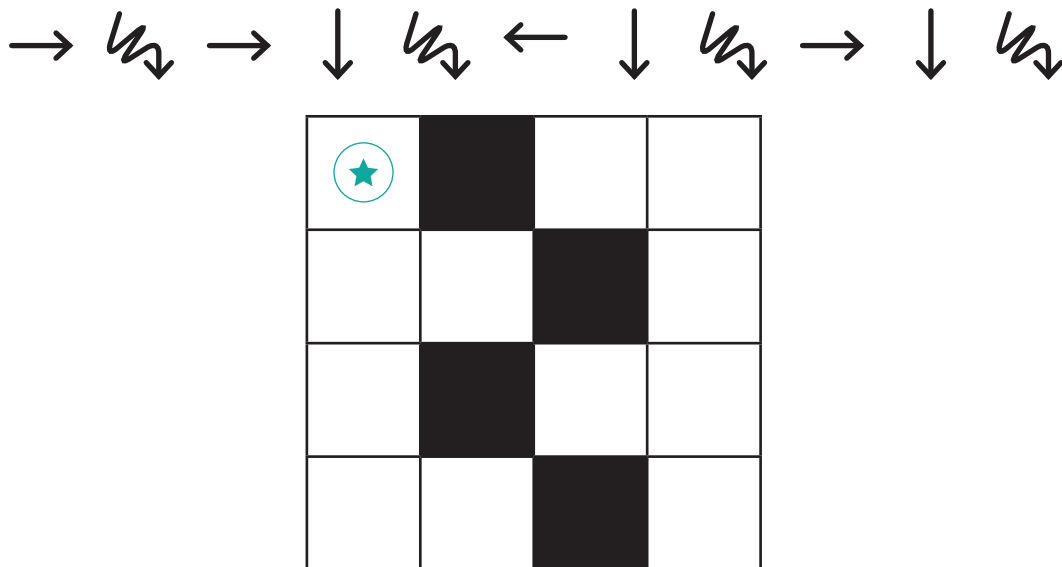


Ahora les pedimos a las parejas de estudiantes que usen el lenguaje de flechas y escriban un programa para dibujar el dibujo previamente expuesto. Un programa que consiga el objetivo –aunque no el único–, es el que se muestra en la imagen.

Programa escrito en el lenguaje de programación de la máquina de dibujar



Una vez que hayan completado la consigna pueden pasar a ejecutar programas. Les pedimos a las parejas que ejecuten el programa de la siguiente imagen, que también muestra el dibujo al que deberían arribar.



Finalmente, pedimos a los estudiantes que de manera individual escriban un programa para dibujar en la cuadrícula. Una vez que terminen de escribirlo, pueden intercambiar programas con un compañero y cada uno dibujar lo programado por el otro.

RÚBRICAS

Al final de la ficha de los estudiantes, incluimos una rúbrica para que los alumnos sepan qué estaremos evaluando y qué criterios utilizaremos.

CIERRE

Discutimos con los estudiantes la diferencia entre algoritmo y programa. En general, los algoritmos están escritos para ser de fácil comprensión para las personas, no para las computadoras. Se los suele escribir usando lenguajes expresivamente muy ricos –como el español, por ejemplo–, pero estos lenguajes suelen dar lugar a interpretaciones diferentes de una misma oración. Por el contrario, los lenguajes de programación tienen una sintaxis rígida y una semántica definida. Al igual que lo que sucede en el mundo de la música con una nota sobre un pentagrama, cada instrucción de un lenguaje de programación tiene una única interpretación posible; por eso usamos lenguajes de programación para dar instrucciones a una computadora.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

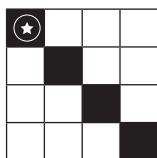
¡A DIBUJAR!

¡Vamos a programar una “máquina de dibujar”!
Para eso usaremos una grilla y este lenguaje de flechas.

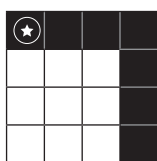
1. A partir del lenguaje de flechas, escribí programas para que la máquina haga los dibujos que muestran las grillas a continuación.

INSTRUCCIONES PARA LA MÁQUINA DE DIBUJAR

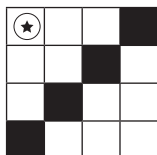
- Moverse un cuadrado a la derecha
- ← Moverse un cuadrado a la izquierda
- ↑ Moverse un cuadrado arriba
- ↓ Moverse un cuadrado abajo
- ↻ Pintar un cuadrado



Paso 1	Paso 2	Paso 3	Paso 4	Paso 5	Paso 6	Paso 7	Paso 8	Paso 9	Paso 10
Paso 11	Paso 12	Paso 13	Paso 14	Paso 15	Paso 16	Paso 17	Paso 18	Paso 19	Paso 20



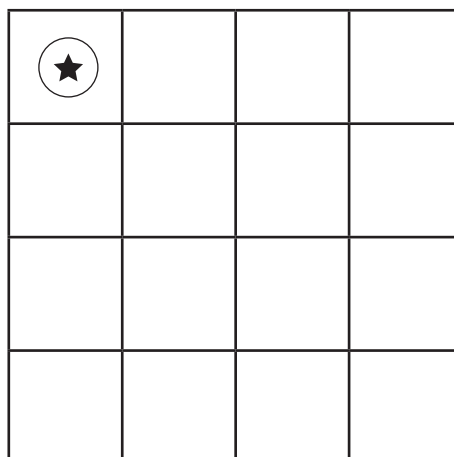
Paso 1	Paso 2	Paso 3	Paso 4	Paso 5	Paso 6	Paso 7	Paso 8	Paso 9	Paso 10
Paso 11	Paso 12	Paso 13	Paso 14	Paso 15	Paso 16	Paso 17	Paso 18	Paso 19	Paso 20



Paso 1	Paso 2	Paso 3	Paso 4	Paso 5	Paso 6	Paso 7	Paso 8	Paso 9	Paso 10
Paso 11	Paso 12	Paso 13	Paso 14	Paso 15	Paso 16	Paso 17	Paso 18	Paso 19	Paso 20

2. Leé el programa que se encuentra debajo ¿Qué dibujo debe hacer la máquina?

→ Paso 1	↻ Paso 2	→ Paso 3	↓ Paso 4	↻ Paso 5	← Paso 6
↓ Paso 7	↻ Paso 8	→ Paso 9	↓ Paso 10	↻ Paso 11	



NOMBRE Y APELLIDO:

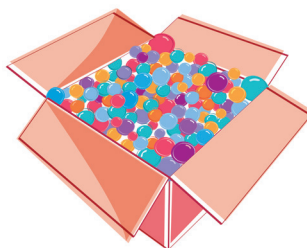
CURSO:

FECHA:

3. De la siguiente lista, ¿qué se parece más a un programa? ¿Por qué?



6 rosas en un florero



Una caja llena de bolitas



La partitura de una canción

EVALUACIÓN

Estas son las tareas y la forma de calificar que tu docente considerará para evaluar tu desempeño.

CALIFICACIÓN	ES NECESARIO TRABAJAR MÁS	BUENO-MUY BUENO	EXCELENTE
Comprensión de las consignas	El estudiante no sigue la imagen modelo y no entiende cómo usar las instrucciones para reproducirla.	El estudiante comprende cómo escribir instrucciones para definir algoritmos que reproducen la imagen modelo, pero comete errores.	El estudiante comprende cómo escribir instrucciones para definir algoritmos que reproducen la imagen modelo, y lo relaciona con la noción de algoritmo introducida en la secuencia anterior. Puede dar instrucciones con facilidad y sin errores.
Difrenciación entre algoritmos y programas	El estudiante confunde las instrucciones que pueden utilizarse en un algoritmo para la máquina de dibujar con su respectiva codificación en programas.	El estudiante distingue entre instrucciones usadas en algoritmos y su codificación en programas para la máquina de dibujar, pero comete errores al traducir de una a la otra.	El estudiante distingue entre instrucciones usadas en algoritmos y su codificación en programas para la máquina de dibujar, y es capaz de traducir fácilmente de una a la otra.
Escritura de programas en la máquina de dibujar	Dada una imagen, el estudiante tiene dificultades importantes en producir el programa asociado.	Dada una imagen, el estudiante es capaz de producir el programa asociado, con algunos errores menores.	Dada una imagen, el estudiante es capaz de producir el programa asociado, sin errores y con facilidad.
Ejecución de programas en la máquina de dibujar	Dado un programa, el estudiante tiene dificultades importantes en producir la imagen asociada.	Dado un programa, el estudiante es capaz de producir la imagen asociada, con algunos errores menores.	Dado un programa, el estudiante es capaz de producir la imagen asociada, sin errores y con facilidad.

Actividad 2

Creamos una cuenta en Scratch



DE A DOS

OBJETIVOS

- Explorar la comunidad en línea de Scratch.
- Explorar el entorno de Scratch.
- Cargar, guardar y ejecutar proyectos.
- Introducir instrucciones del lenguaje Scratch.

MATERIALES



Computadora



Conexión a Internet



Scratch

Sobres de papel
(uno para cada estudiante)

Ficha para estudiantes

DESARROLLO

A lo largo de esta actividad, los estudiantes ejecutarán por primera vez Scratch y empezarán a familiarizarse con su interfaz. Comenzarán creando sus cuentas de Scratch, que usarán a lo largo de todos los proyectos del manual. Luego, harán una primera experiencia de programación en computadoras.

Si se cuenta con acceso a Internet, es posible acceder a Scratch desde cualquier navegador. Basta dirigirse a <https://scratch.mit.edu> y crear una cuenta. Aconsejamos utilizar este método, ya que permite que los estudiantes tengan acceso a una gran cantidad de proyectos, además de darles la posibilidad de compartir sus trabajos con toda la comunidad de Scratch.

Si no se cuenta con acceso a Internet, es posible usar la aplicación Scratch Offline Editor, disponible para Windows, Linux y Mac en <https://scratch.mit.edu/scratch2download>.

El desarrollo de la actividad si se usa el entorno en línea es diferente que si se usa el editor sin Internet. Las interfaces de ambos son prácticamente idénticas, pero el entorno en línea requiere de la creación de una cuenta de usuario. Comenzamos indicando los pasos necesarios para realizar este proceso. Luego, seguimos con la actividad común a ambos entornos.

Creación de una cuenta para el entorno en línea

La creación de cuentas de usuario para la versión en línea de Scratch requiere de una dirección de correo electrónico. Como los estudiantes deben proporcionar una, sugerimos usar una cuenta propia o crear una dirección de correo electrónico para toda la clase. Hay que tener en cuenta que las notificaciones de cualquier comportamiento inadecuado en el sitio web de Scratch se enviarán al correo electrónico que esté registrado en la cuenta.

¿Qué es Scratch?

Scratch es un lenguaje de programación en bloques orientado a la enseñanza, principalmente mediante la creación de juegos. En Ciencias de la Computación, *scratching* quiere decir 'reutilizar código', lo que puede ser usado de forma beneficiosa y efectiva para otros propósitos, y fácilmente combinado, compartido y adaptado a nuevos escenarios.

De esta forma, cualquier usuario puede descargar proyectos públicos desarrollados por otros usuarios y trabajar sobre ellos.

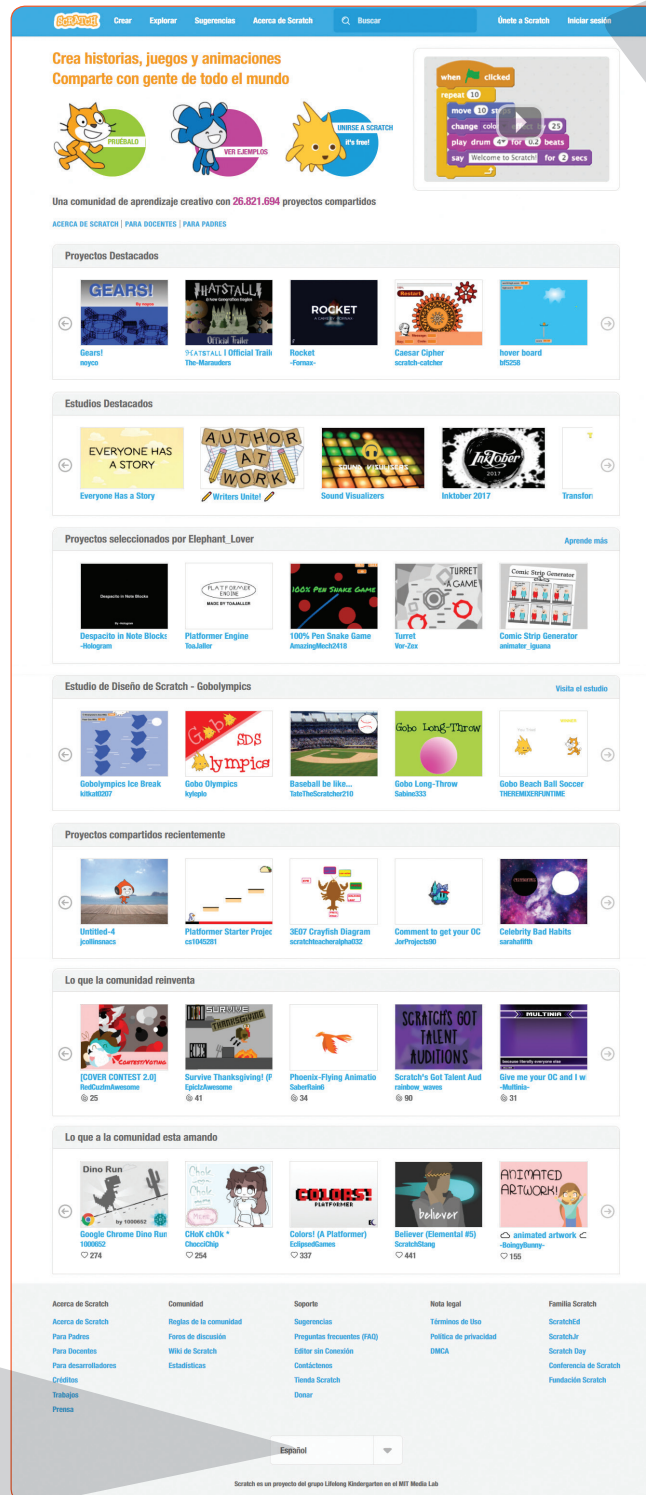


1. Ingresar al sitio web y seleccionar el idioma

Comenzamos pidiendo a los estudiantes que ingresen al sitio web de Scratch en <https://scratch.mit.edu>, y ayudamos a quienes tengan dificultades para llegar hasta ahí. Si al acceder se muestra la página en inglés, en la parte inferior seleccionamos el idioma español.

Español

Selección de idioma



Únete a Scratch

Iniciar sesión

2. Crear una cuenta nueva

Guiamos a los estudiantes para crear una cuenta. Indicamos hacer clic en *Únete a Scratch*, frase ubicada en la parte superior derecha de la página.

3. Seleccionar un nombre de usuario y una contraseña

Crear una cuenta es sencillo. Solo hay que seguir una serie de pasos. En primer lugar, se debe seleccionar un nombre de usuario y una contraseña, como muestra la siguiente imagen.

Únete a Scratch

Es fácil (y gratis) registrar una cuenta Scratch.

Elige un nombre de usuario en Scratch [No uses tu nombre real](#)

Elija una contraseña

Confirmar contraseña



1 2 3 4

[Siguiente](#)

4. Ingresar fecha de nacimiento, género y país

A continuación, hay que indicar fecha de nacimiento, género y país.

Únete a Scratch

Tus respuestas a estas preguntas se guardarán confidencialmente.

Por qué pedimos estos datos ?

Mes y año de nacimiento - Mes - - Año -

Género ☐ Masculino ☐ Femenino

País - País -



1 2 3 4

[Siguiente](#)

5. Indicar un correo electrónico

Por último, se indica una cuenta de correo electrónico, en la que se recibirá un pedido de confirmación de la dirección ingresada.

Únete a Scratch

Ingrese su dirección de correo electrónico y le enviaremos un mensaje de confirmación a su cuenta.

Correo electrónico

Confirmar su correo electrónico

☐ Recibe las últimas novedades del Scratch Team



1 2 3 4

[Siguiente](#)

6. Consejo para recordar la contraseña

Para recordar las contraseñas y mantener la privacidad de los estudiantes, les pedimos que anoten sus nombres de usuario y contraseñas en un papel. Luego, les pedimos que introduzcan ese papel en sobres individuales con sus nombres y los cerramos. Finalmente, guardamos todos los sobres en un lugar seguro dentro del aula.

Únete a Scratch

Bienvenido a Scratch!

Ahora ya estás registrado! Puedes comenzar a explorar y crear proyectos.

Si deseas compartir y comentar, simplemente haz clic en el correo electrónico que te enviamos.

Correo electrónico inválido? Cambia tu dirección de correo en [Configuración de la Cuenta](#).

¿Tienes problemas? [Por favor envía tu opinión](#)



1 2 3 4

[OK Vamos!](#)

Guiamos a los estudiantes para que sigan las indicaciones de la ficha y les damos un tiempo para que puedan registrarse, actualicen su página de perfil y exploren la comunidad en línea de Scratch. Los motivamos para que practiquen entrar y salir de sus cuentas. Para facilitar que encuentren y sigan los perfiles de otros, podemos crear una lista de los nombres de todos los estudiantes y sus respectivos nombres de usuario.

Podemos presentar a la clase las reglas de la comunidad de Scratch que se encuentran en el sitio web, para discutir sobre comportamiento respetuoso y constructivo durante el uso de la herramienta.

Actividad para ambos entornos de Scratch

Pedimos a los estudiantes que ingresen a Scratch. Aquellos que usen la **plataforma en línea**, deben entrar en <https://scratch.mit.edu/> desde el navegador. Si la página está en inglés, sugerimos cambiar el idioma a español, como ya se ha indicado. En la página de ingreso, deben hacer clic en *Iniciar sesión* e ingresar su nombre de usuario y contraseña.

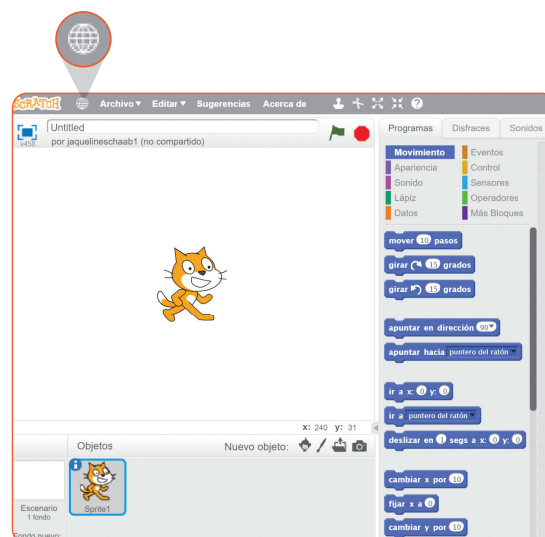


Opciones para ingresar en línea a Scratch

Luego de ingresar, veremos que el menú ha cambiado: donde decía *Iniciar sesión* aparece el nombre de nuestro usuario. Para crear un proyecto, se debe hacer clic en *Crear*, ubicado a la izquierda de la barra superior. Esto nos lleva al editor de Scratch. Al ingresar, veremos una pantalla como la que se observa a la derecha.

Si la interfaz aparece en inglés, hacemos clic en el ícono del mundo que se ubica al lado de la palabra *Scratch* (arriba a la izquierda) y cambiamos el idioma a español. Así podremos ver las opciones del menú y los bloques de instrucciones en nuestro idioma.

En el **Offline Scratch Editor**, basta ejecutar el programa para acceder directamente al editor.



Pantalla de inicio del entorno Scratch

Configuración del navegador

De acuerdo con la configuración de algunas máquinas, cuando se presiona por primera vez el botón *Crear* se va a una pantalla que dice *Active Adobe Player*. En ese caso, hay que permitir la activación para poder acceder al editor de Scratch.

La interfaz de Scratch tiene cuatro partes principales.

- Arriba a la izquierda, vemos un área con fondo blanco, en donde está el gato Scratch. Allí aparecerán los distintos personajes a los que daremos instrucciones con nuestros programas. A estos personajes se los llama en Scratch *objetos* cuando la interfaz está en español, o *sprites* si están en inglés.
- Abajo a la izquierda, se observa un área en la que podemos manejar los distintos objetos de nuestro proyecto. Cada objeto está asociado a un programa que contiene las instrucciones que dicho objeto debe seguir. Si seleccionamos un objeto, podemos ver las instrucciones asociadas a él en la parte derecha con fondo gris. En la figura de la página anterior está seleccionado el gato Scratch.
- En el centro se puede ver una lista de bloques de instrucciones. En la imagen de la página anterior estamos viendo instrucciones de color azul, agrupadas bajo la categoría *Movimientos*. Podemos elegir diferentes conjuntos de instrucciones haciendo clic en las etiquetas con distintos colores que se ven arriba de las instrucciones.
- Finalmente, en la parte derecha está el espacio donde armaremos nuestros programas.

Proponemos a los estudiantes que escriban programas y los prueben. Para hacerlo, deben seleccionar bloques y arrastrarlos a la zona gris. Al posicionar un bloque debajo del otro, vemos que se van encastrando: de esta forma, se va armando un programa. La imagen de la derecha muestra un posible programa.

Una vez armado el programa, pedimos a los alumnos que hagan clic sobre cualquiera de los bloques que lo componen y observen el efecto en el gato Scratch. Remarcamos que cada instrucción que posee un óvalo blanco tiene un valor predefinido. Si hacemos clic en el óvalo, podemos cambiar los valores numéricos.



Ejemplo de programa en Scratch

Proponemos a los estudiantes que experimenten con distintos números para que observen los cambios al ejecutar nuevamente el programa. Iniciamos una discusión para que los estudiantes expliquen qué está pasando y para que encuentren el significado de los distintos bloques usados en el programa.

Mostramos que haciendo clic continuo sobre una instrucción y moviendo el cursor podemos separar el programa en partes. Por otro lado, haciendo clic derecho sobre una parte vemos un menú de opciones cómo *duplicar* o *borrar*, como se muestra a la derecha. Existe también la opción *ayuda*, que abre una ventana con una descripción del significado del bloque. En la versión *offline* de Scratch estas descripciones están solo disponibles en inglés.



Programa separado en partes y opciones desplegadas al presionar el botón derecho del ratón

Para finalizar, indicamos a los estudiantes que usen la opción **Archivo > Guardar** para guardar el programa. Al hacerlo, deben elegir un nombre para el archivo. Es importante recordar en qué directorio se guarda el programa. Pedimos que utilicen la opción **Archivo > Nuevo** para comenzar nuevamente con un proyecto desde cero. Pueden utilizar la opción **Archivo > Abrir** para volver a trabajar en un proyecto previamente guardado.

Terminamos esta actividad cerciorándonos de que todos los estudiantes sepan cómo realizar las tareas indicadas hasta el momento, de modo que estén listos para explorar por su cuenta nuevos bloques de Scratch en la próxima actividad.

CIERRE

A modo de cierre, relacionamos el programa Scratch con los algoritmos trabajados en actividades anteriores. Antes trabajamos con descripciones paso a paso para realizar una tarea. Algo parecido sucedió en esta actividad: el gato Scratch siguió paso a paso cada una de las instrucciones que los estudiantes incluyeron en sus programas. Por otro lado, hacemos notar que, en la experiencia con esta herramienta, primero escribieron un programa y luego lo ejecutaron, y que estas dos instancias son diferentes.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

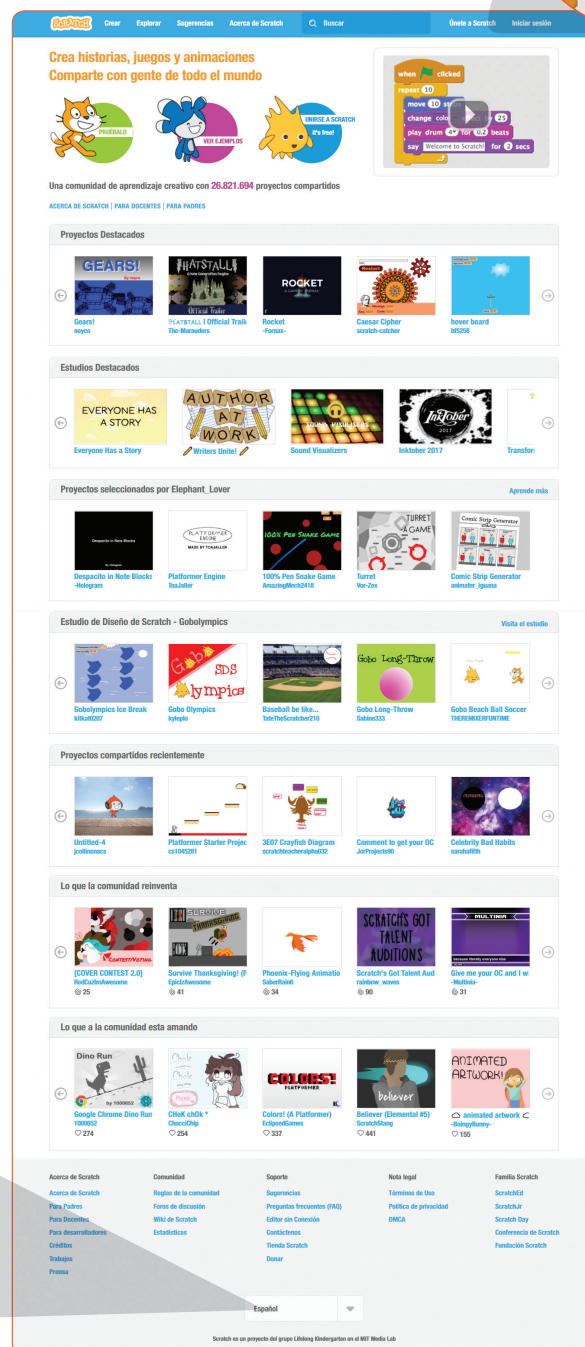
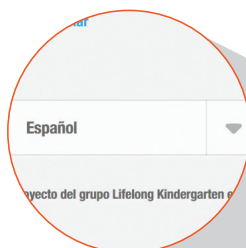
CREAMOS UNA CUENTA EN SCRATCH



¡Vamos a aprender a usar Scratch! Scratch es a la vez un lenguaje y un entorno de programación en el que podés desarrollar tus programas. Para eso, vas a necesitar abrir una cuenta en el sitio de Scratch en línea para poder crear, guardar y compartir tus proyectos.

¿Ya estás frente a una compu con Internet? Seguí estas instrucciones:

1. Abrió un navegador web y cargá la dirección de Scratch: <https://scratch.mit.edu>.
2. ¿La página está en inglés? Andá hasta el final de la página y desplegó el menú. Seleccioná el idioma español.
3. En la página principal, andá a la parte superior derecha de la barra y hacé clic en *Únete a Scratch*.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

4. Completá los siguientes pasos para registrar tu cuenta en Scratch.

PASO 1: ELEGÍ UN NOMBRE Y UNA CONTRASEÑA

Elegí un nombre de usuario y una contraseña. Es importante que el nombre de usuario no sea tu nombre real para que no puedan identificarte; así podés preservar tu privacidad en la web. Cuando elijas la contraseña, pensá alguna pista que te ayude a recordarla. Cuando termines, presioná *Siguiente*.



Únete a Scratch

Es fácil (y gratis) registrar una cuenta Scratch.

Elige un nombre de usuario en Scratch [No uses tu nombre real](#)

Elija una contraseña

Confirmar contraseña



1 2 3 4 

[Siguiente](#)

PASO 2: COMPLETÁ TUS DATOS PERSONALES

En esta pantalla tenés que indicar tu mes y año de nacimiento, si sos varón o mujer y tu país de residencia. Cuando termines, presioná *Siguiente*.



Únete a Scratch

Tus respuestas a estas preguntas se guardarán confidencialmente.

Por qué pedimos estos datos 

Mes y año de nacimiento - Mes - - Año -

Género ☐ Masculino ☐ Femenino

País - País -




1 2 3 4 

[Siguiente](#)

PASO 3: AGREGÁ UN CORREO ELECTRÓNICO

Para completar el registro de tu nueva cuenta, indicá una dirección válida de correo electrónico. Tu docente te dirá cuál tenés que usar.



Únete a Scratch

Ingrese su dirección de correo electrónico y le enviaremos un mensaje de confirmación a su cuenta.

Correo electrónico

Confirmar su correo electrónico

☐ Recibe las últimas novedades del Scratch Team



1 2 3 4 

[Siguiente](#)

PASO 4: ¡LO LOGRASTE!

¡Ya podés ingresar a la comunidad Scratch en línea!



Únete a Scratch

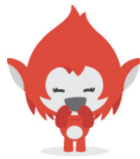
Bienvenido a Scratch!


Ahora ya estás registrado! Puedes comenzar a explorar y crear proyectos.

Si deseas compartir y comentar, simplemente haz clic en el correo electrónico que te enviamos.

Correo electrónico inválido? Cambia tu dirección de correo en [Configuración de la Cuenta](#).

¿Tienes problemas? [Por favor envía tu opinión](#)



1 2 3 4 

[OK Vamos!](#)

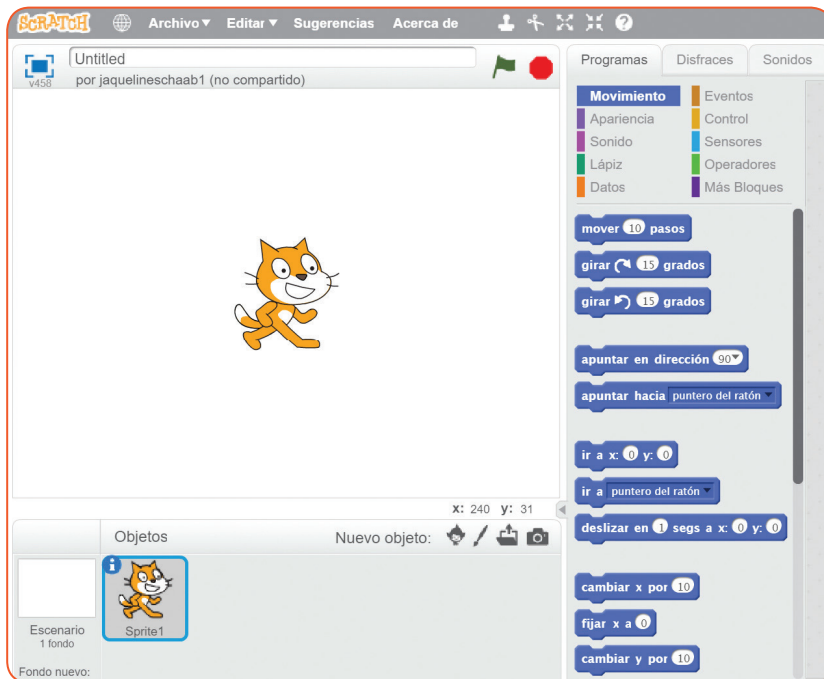
NOMBRE Y APELLIDO:

CURSO:

FECHA:

CREAMOS NUESTRO PRIMER PROGRAMA

Ahora, ¡a divertirnos! Creá y ejecutá tu primer programa siguiendo las instrucciones de tu docente para poder ingresar al editor de Scratch. ¿Ves una pantalla como esta?



5. Mirá las distintas partes de la pantalla. Con la ayuda de tu docente, construí el programa que se muestra a la derecha en el área de programación. Para hacerlo, tenés que buscar los bloques en la lista que se ve en el centro de la pantalla y arrastrarlos, de a uno por vez, hacia el área de programación. Observá que, si ponés un bloque debajo del otro, se encastran y van armando el programa.

Una vez que completaste el programa, ejecutalo haciendo clic en cualquiera de sus bloques. Mirá la ventana de la izquierda. ¿Qué está haciendo el gato Scratch?



6. Probá cambiar los números que aparecen en los bloques. Para eso, hacé clic en los recuadros blancos y escribí otro número. ¿Qué efecto tiene? ¿Podés cambiar los números para que el gato empiece y termine en el mismo lugar?

Actividad 3

¡Algo sorprendente!



DE A DOS

OBJETIVO

- Realizar una experiencia exploratoria con Scratch.

MATERIALES



Computadoras con acceso a Internet



Scratch



Ficha para estudiantes

DESARROLLO

Esta es una actividad abierta, en la que los estudiantes utilizarán el editor de Scratch para crear programas, sin un fin predeterminado. El objetivo es que se familiaricen con el uso de la interfaz y que comiencen a explorar algunos bloques por su cuenta.

Ayudamos a los estudiantes a abrir el editor de proyectos de Scratch (ya sea fuera de línea, en el entorno instalado en las PC, o en línea, si ingresamos al sitio web <https://scratch.mit.edu> y entramos en la cuenta que creamos previamente). Les pedimos a los alumnos que comiencen un proyecto haciendo clic en **Crear** en la parte superior de la página. Tengamos a mano la ficha de la actividad para guiar a los estudiantes durante sus exploraciones.

Damos a los estudiantes 10 minutos para explorar la interfaz de Scratch de manera libre. Los animamos con frases tales como: “Tienen 10 minutos para hacer que algo sorprendente le suceda al gato Scratch” o “tomen 10 minutos para explorar la interfaz, ¿qué observan?”. Alentamos a los alumnos a trabajar juntos, a ayudarse unos a otros y a compartir sus descubrimientos.

Pedimos a 3 o 4 voluntarios que compartan con todo el grupo las cosas que descubrieron. Después, hacemos preguntas que los inviten a seguir explorando el entorno: “¿Alguien sabe cómo agregar sonido? ¿Alguien averiguó cómo cambiar el fondo? ¿Quién pudo agregar al proyecto otro personaje, además del gato Scratch? ¿Alguien descubrió cómo obtener ayuda sobre los bloques?”.

RÚBRICAS

Al final de la ficha de los estudiantes, incluimos una rúbrica para que los alumnos sepan qué estaremos evaluando y qué criterios utilizaremos.

CIERRE

A modo de cierre, comentamos a los estudiantes que el entorno ofrece muchísimas opciones y que no deben preocuparse si no entienden algunas. Indicamos que, a lo largo del curso, irán aprendiendo a usar muchas de ellas para construir sus propios programas.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

¡ALGO SORPRENDENTE!

¡Tiempo de explorar Scratch! Hoy vale probar todo. Probá, cliqueá y después contanos.



1. Hacé clic en diferentes partes de la interfaz y fijate qué pasa. Jugá con diferentes bloques: arrástralos y soltalos en el área de programación. Experimentá haciendo clic en cada bloque para ver qué hace. En la imagen de abajo tenés un ejemplo para que te sirva de inspiración.



2. ¿Probaste agregar sonido? ¿Y cómo se hace para cambiar el fondo del gato? ¿Te fijaste si podés agregar otro personaje? Esta es tu oportunidad para seguir explorando Scratch.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

EVALUACIÓN

Estas son las tareas y la forma de calificar que tu docente considerará para evaluar tu desempeño.

CALIFICACIÓN	ES NECESARIO TRABAJAR MÁS	BUENO-MUY BUENO	EXCELENTE
Interfaz de Scratch	El estudiante no reconoce las distintas partes de la interfaz de Scratch y tiene problemas al navegar por el sitio.	El estudiante reconoce las partes principales de la interfaz de Scratch (área de ejecución, área de objetos, área de bloques y área de programación).	El estudiante reconoce las partes principales de la interfaz de Scratch (área de ejecución, área de objetos, área de bloques y área de programación) y las utiliza con facilidad. Ha explorado también otros aspectos de la interfaz por su cuenta.
Manejo de bloques	El estudiante tiene problemas eligiendo bloques, arrastrándolos y combinándolos en el área de programación.	El estudiante puede elegir bloques, arrastrarlos y combinarlos en el área de programación.	El estudiante puede elegir bloques, arrastrarlos y combinarlos en el área de programación. Es capaz de modificar parámetros y de reorganizar y borrar bloques.
Guardar, cargar y ejecutar programas	El estudiante tiene problemas creando, grabando, cargando o ejecutando programas.	El estudiante es capaz de grabar crear y ejecutar programas, con algunas dificultades.	El estudiante es capaz de grabar crear y ejecutar programas.
Crear programas	El estudiante crea programas al azar, sin comprender el efecto que tendrán durante la ejecución.	El estudiante es capaz de crear programas y puede relacionar los pasos del programa con los pasos durante la ejecución.	El estudiante es capaz de explicar qué realiza un programa antes de ejecutarlo, y construye programas con un objetivo previo.

03

EVENTOS, CONDICIONALES Y CICLOS

SECUENCIA DIDÁCTICA 1

EVENTOS

El control movitrónico

La máquina de dibujar 1.0

SECUENCIA DIDÁCTICA 2

CONDICIONALES

La carrera de los palos españoles

La máquina de dibujar 2.0

SECUENCIA DIDÁCTICA 3

CICLOS

Ay, ay, ay es tres veces ay

La máquina de dibujar 3.0

Los lenguajes de programación proveen distintas construcciones que permiten definir cómo es el flujo de ejecución de las instrucciones de nuestros programas. En este capítulo, en primer lugar, se presentan algunas nociones que posibilitan expresar cómo debe reaccionar un programa ante algún **evento** externo (por ejemplo, cada vez que se presione una tecla del teclado o un botón del ratón). A continuación, se trabaja sobre **sentencias condicionales** y sus usos en programación, que permiten que los programas tomen decisiones. Por último, se ejercita el reconocimiento de patrones y el uso de **ciclos** en los programas.



Secuencia Didáctica 1

EVENTOS

Un **evento** es un suceso que puede acontecer súbitamente, sin que sepamos *a priori* cuándo va a ocurrir. Al programar, tenemos la posibilidad de definir cómo queremos que nuestros programas reaccionen cuando se produce uno. Al hacerlo, llevamos a cabo lo que en programación se conoce como **manejo de eventos**.

Esta secuencia didáctica está compuesta por dos actividades. La primera se realiza sin computadora y tiene como objetivo diferenciar los eventos de su manejo. La segunda pone en práctica estas construcciones en un proyecto en Scratch. Este proyecto se retoma en secuencias didácticas posteriores de este capítulo.

.....

OBJETIVOS

- Introducir la noción de evento.
- Diferenciar eventos de manejo de eventos.
- Poner en práctica estas nociones en un proyecto de programación.

.....

Actividad 1

El control movitrónico¹

TODA LA CLASE

OBJETIVOS

- Introducir la noción de evento.
- Diferenciar un evento de las reacciones que produce.

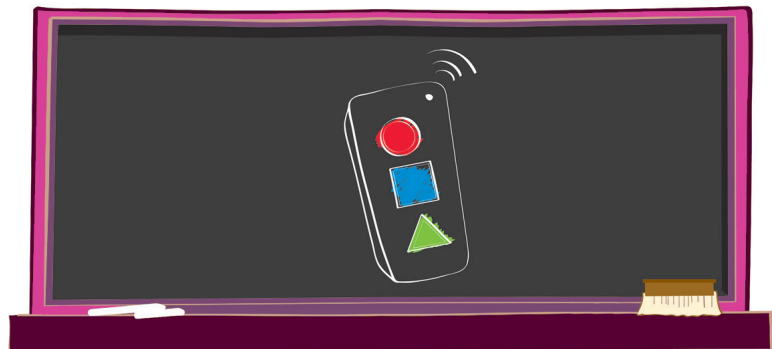
MATERIALES

 Tizas de tres colores diferentes

 Pizarrón

DESARROLLO

En esta actividad buscamos que los estudiantes comprendan qué entendemos por evento, y sean capaces de diferenciarlo de las acciones que un evento desencadena. Recomendamos tener dibujadas en el pizarrón las siguientes tres figuras.



El control movitrónico

Comenzamos preguntando a la clase: “¿Cómo hacen en sus casas para prender el televisor?”. Es probable que alguno responda que usa el control remoto; esto, si bien es cierto, también es inespecífico. Guiamos las preguntas hasta que los estudiantes identifiquen que para encender el televisor oprimen el botón de encendido del control remoto.

Preguntamos luego: “¿Qué sucede entonces?”. Seguramente algún estudiante dirá que se prende el televisor. Comentamos: “Es interesante que notemos las distintas cosas que suceden cuando queremos prender el televisor. En primer lugar, apretamos el botón de encendido del control remoto. Esa acción, que realizamos nosotros, genera una reacción por parte del televisor: se enciende. Apretar el botón es un evento. Prenderse, es la forma en que el televisor responde cuando se produce el evento”. Escribimos en el pizarrón la palabra **evento**.

Les contamos a los estudiantes que vamos a jugar al control movitrónico y les explicamos las reglas. El dibujo del pizarrón es un control remoto. Cuando apretemos el botón rojo, todos los estudiantes deben aplaudir; con el botón verde, todos extienden las manos hacia arriba; y con el botón azul, todos ponen ambas manos sobre la cabeza. Por ejemplo, si presionamos los botones en el orden rojo, azul, rojo, verde, los estudiantes deberían primero aplaudir, luego poner las manos sobre sus cabezas, aplaudir nuevamente y por último extender las manos hacia arriba.

Una vez comprendidas las reglas, empezamos a “apretar” los botones dibujados en el pizarrón. Cada vez que presionamos un botón, los estudiantes tienen que realizar la acción correspondiente. Deberán estar atentos, pues su respuesta al evento de presionar un botón tiene que ser rápida y adecuarse a las reglas del juego planteadas.

¹ Adaptación de la actividad “The Big Event” de Code.org, disponible en <https://goo.gl/CcclAr>.



Reacción de los estudiantes al presionar los botones en el orden rojo, azul, rojo y verde

Luego de un rato, dividimos al curso en dos grupos. A uno le cambiamos las reacciones previstas para cada botón. Por ejemplo, podemos indicar que si presionamos el botón rojo tienen que saltar una vez, si apretamos el verde tienen que taparse los ojos con las manos y si oprimimos el rojo, tienen que decir la palabra *movitrónico*.

Jugamos con los dos conjuntos de reglas: la mitad de la clase reaccionará de una manera y la otra mitad lo hará de otro modo. Disfrutamos del juego con los estudiantes.

CIERRE

Preguntamos a la clase: “¿Notan alguna similitud entre lo que estuvimos haciendo y lo que hablamos acerca de cómo encender un televisor?”. Guiamos la reflexión para hacerles notar que, al igual que con el control remoto del televisor, cuando presionamos los botones del control movitrónico provocamos eventos. Frente a tales eventos, los estudiantes respondieron. Finalmente, comentamos que esto sucede muy a menudo cuando usamos dispositivos digitales. Se puede mencionar, por ejemplo, que en los teléfonos celulares, cuando presionamos el ícono de una aplicación, se abre un programa en nuestro dispositivo.

Actividad 2




La máquina de dibujar 1.0

DE A DOS

OBJETIVO

- Escribir un programa que maneje eventos.

MATERIALES

-  Computadoras
-  Scratch
-  Ficha para estudiantes

DESARROLLO

En esta actividad comenzaremos un pequeño proyecto de programación con el fin de construir una máquina para dibujar, que será retomado en actividades posteriores de este capítulo. Una implementación completa se encuentra disponible en <https://goo.gl/6L9NL6>.

Trabajaremos con una pelota que, a medida que apretemos flechas en el teclado, será capaz de dibujar líneas en la pantalla. Dicho de un modo más técnico, programaremos cómo la pelota va a manejar los eventos que suceden cuando apretamos ciertas teclas. Desde una perspectiva más general, haremos lo que siempre hacemos al programar en Scratch: determinar cómo se comportan algunos objetos en respuesta a distintos eventos.

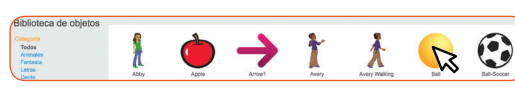
Empezamos pidiéndoles a los estudiantes que carguen Scratch. En caso de que usen el editor en línea, les indicamos que en primer lugar inicien una sesión y luego hagan clic sobre la opción *Crear* del menú superior.



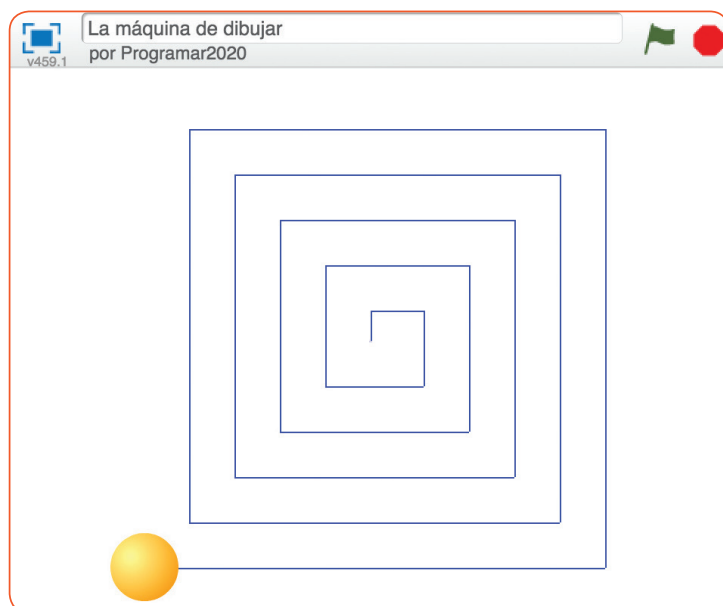
Para un mejor aspecto gráfico, es conveniente no usar el gato de Scratch como protagonista de esta actividad. En su lugar, sugerimos usar el objeto *Ball*, que tiene forma de pelota. Para ello, primero borramos al gato de la escena presionando el botón derecho del ratón sobre el gato de la galería *Objetos*.



A continuación, incorporamos un nuevo objeto de la biblioteca. Durante el resto de la actividad, usaremos como ejemplo el objeto *Ball* de la *Biblioteca de objetos*. Los estudiantes podrían querer usar algún otro objeto. Esto solo modificaría la visualización, pero la esencia de la actividad se mantiene.



Una vez incorporada la pelota en la escena, nos enfocamos en programar cómo debe reaccionar el objeto ante los eventos que consisten en presionar cada una de las flechas del teclado. Es importante tener siempre presente que, al programar, definimos las acciones que realizará el objeto de la galería que se encuentre seleccionado en ese momento.

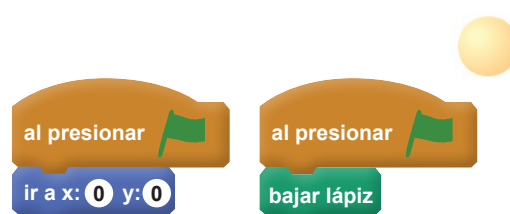


La máquina de dibujar 1.0

Les contamos a los estudiantes que vamos a hacer un programa para dibujar. Explicamos que la bola amarilla será la punta de un lápiz y el fondo de la pantalla, una hoja de papel blanco. Hay que conseguir que, al presionar una flecha en el teclado de la computadora, la bola se desplace en la dirección que indica la flecha y dibuje una línea. Repartimos la ficha y alentamos a los estudiantes a que exploren el entorno para descubrir cómo hacerlo. A medida que trabajan, nos acercamos a sus bancos y, de ser necesario, aclaramos las dudas que vayan surgiendo. Para resolver el desafío propuesto hay que tener en cuenta, por un lado, lo que queremos que suceda al comenzar a correr nuestro programa, y por el otro, debemos conseguir pintar usando la pelota.

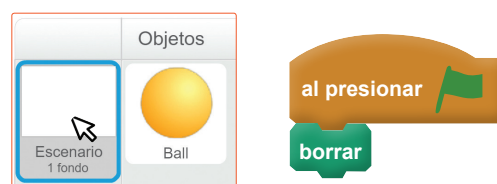
Es necesario tener presente todo el tiempo que la pelota debería comportarse como un lápiz; es decir, que cada vez que se mueva, dibuje una línea. En la categoría **Lápiz** está el bloque `bajar lápiz`. Una estrategia práctica consiste en usarlo junto con el bloque `al presionar bandera verde` de la categoría **Eventos**. En general, presionaremos la bandera verde para dar comienzo a una ejecución de nuestros programas. Por lo tanto, es un momento apropiado para que la pelota baje el lápiz y lo deje preparado para dibujar.

Sucesivas corridas de un programa Scratch van dejando rastro. Si usamos el programa para dibujar unas líneas y luego presionamos la bandera verde, ni la pelota volverá a su posición original ni se limpiará la hoja. Si queremos que eso suceda, tenemos que ocuparnos nosotros. Cada posición de un objeto en una escena de Scratch se identifica con dos coordenadas x e y . Estas representan posiciones respecto de un eje de coordenadas que ubica el centro de la pantalla en la posición $x=0$ e $y=0$. Los valores positivos de x identifican la parte derecha de la pantalla y los valores negativos, la parte izquierda. De forma similar, los valores positivos de y comprenden la parte superior de la pantalla y los valores negativos, la parte inferior. Para reposicionar nuestro objeto en el centro, debemos usar el bloque `ir a x:[] y:[]` y los valores 0 y 0 .



Inicialización de la pelota

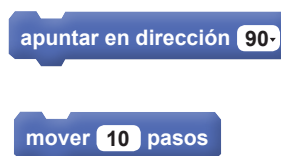
A continuación nos ocuparemos de limpiar el papel de dibujos residuales. Recordemos que, para programar, debemos seleccionar el objeto adecuado en la galería de objetos. En este caso, el fondo. Indicamos que al presionar la bandera verde se limpie todo lo que se encuentre dibujado en el papel. Para hacerlo, usamos el bloque `borrar` de la categoría *Lápiz*.

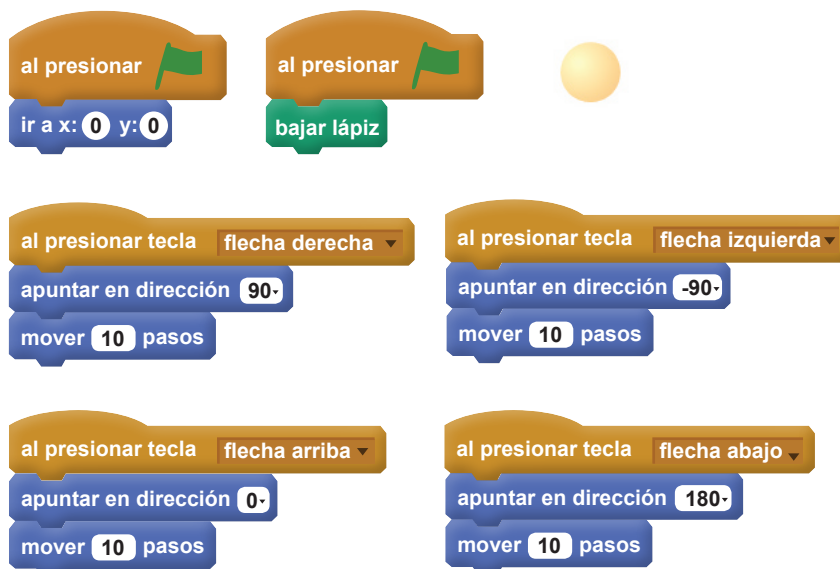


Inicialización del fondo

Ahora nos enfocamos en indicarle a la bola qué hacer cada vez que se presiona una flecha. Es importante tener en cuenta que los objetos de una escena de Scratch se encuentran apuntando a una dirección en todo momento. Las direcciones se representan con un número. En esta actividad, usamos valores para indicar únicamente direcciones horizontales y verticales. Utilizamos los valores 90 , -90 , 0 y 180 , que representan las direcciones derecha, izquierda, arriba y abajo respectivamente.

Una alternativa posible consiste en hacer que la pelota apunte en la dirección adecuada de acuerdo a cada flecha y a continuación desplazarla. Así como para indicar las direcciones usamos números, también los tenemos que utilizar para mover la bola, en este caso, para indicar la cantidad de pasos que dará. Variando este valor se obtienen líneas más largas y más cortas. El valor 10 del ejemplo produce desplazamientos de pequeña longitud. A continuación se muestra una posible solución.

Bloques de la categoría *Movimiento*



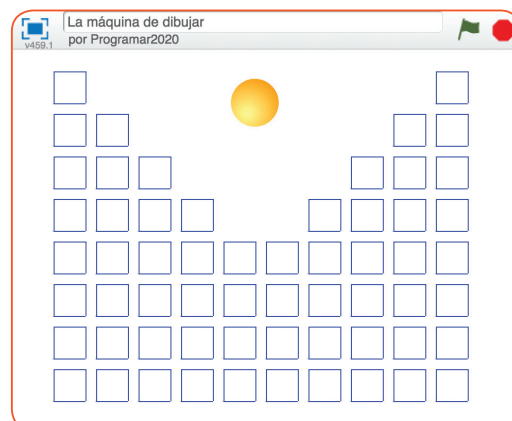
Propuesta de solución de las consignas 1 y 2 de la ficha

Con el programa propuesto, cada vez que movemos la pelota, queda un rastro de su paso. La consigna 3 de la ficha propone que los estudiantes consigan que la bola pueda desplazarse sin pintar. Puntualmente, que al presionar el 1 la máquina apoye la bola en el papel para poder dibujar y que al apretar el 2, la levante y se desplace sin dejar rastro. Para conseguirlo, alcanza con combinar los bloques `bajar lápiz` y `subir lápiz` con el bloque `al presionar tecla []` de la categoría *Eventos*.

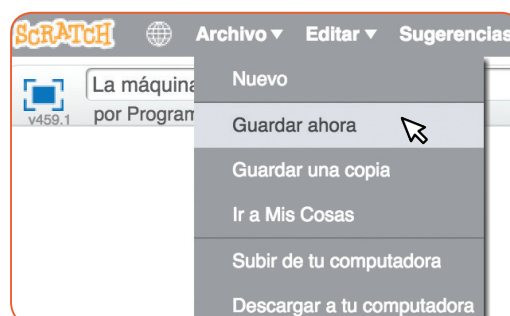


La pelota puede desplazarse sin dejar rastro

Invitamos a los estudiantes a que usen las máquinas programadas por ellos mismos. Se espera que comprueben experimentalmente que pueden usar la pelota para dibujar y desplazarse pintando y sin pintar.



Por último les pedimos que conserven sus programas, pues los retomaremos en otras actividades. Para eso es conveniente primero ponerles un nombre; esto puede hacerse en el bloque para texto que se encuentra en la parte superior de la pantalla. Luego, usando la opción *Guardar ahora* del menú superior del entorno se puede conservar el proyecto.



RÚBRICAS

Al final de la ficha de los estudiantes, incluimos una rúbrica para que los alumnos sepan qué estaremos evaluando y qué criterios utilizaremos.

CIERRE

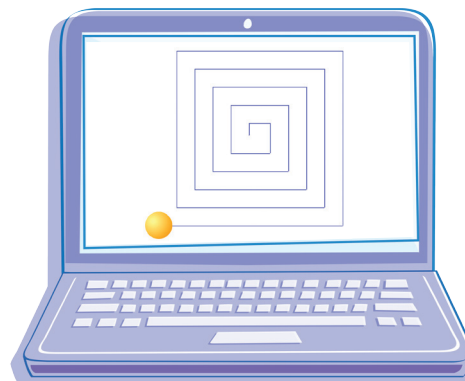
Repasamos con los estudiantes que, en este caso, hemos programado las respuestas de un objeto ante la ocurrencia de distintos eventos. Los eventos se producen al presionar ciertas teclas. Los estudiantes formularon instrucciones para que la pelota realice determinadas acciones cuando se producen los eventos.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

LA MÁQUINA DE DIBUJAR 1.0



¿Alguna vez usaste un programa para dibujar? ¡Ahora vas a hacer uno para que otros puedan hacerlo! Vamos a usar una pelota como lápiz y una hoja blanca como fondo.

1. Para construir la máquina de dibujar, lo primero que tenés que conseguir es que, cuando la pelota se mueva, dibuje una línea. Consejo: primero podés pensar qué eventos deberían provocar el movimiento de la pelota. Hacé una lista y escribilos abajo.

2. Ahora sería conveniente poder poner todo como al comienzo, ¿no? Para esto tenés que conseguir que el programa se reinicie: el papel debe quedar limpio y la pelota en el centro. ¿Cómo hiciste para lograrlo?

VOLVER A EMPEZAR

A lo largo de los proyectos de programación en Scratch, usaremos el evento **al presionar bandera verde** para reiniciar un programa. Es decir, para restablecer algunas de las condiciones iniciales.

al presionar



NOMBRE Y APELLIDO:

CURSO:

FECHA:

3. Por último la pelota también tiene que poder desplazarse sin pintar el papel. Hacé que se mueva dibujando después de que se presione la tecla y que se desplace sin dejar rastro luego de que se apriete la tecla . ¿Qué bloques usaste?
-

EVALUACIÓN

Estas son las tareas y la forma de calificar que tu docente considerará para evaluar cómo resolvés las actividades.

CALIFICACIÓN	ES NECESARIO TRABAJAR MÁS	BUENO-MUY BUENO	EXCELENTE
Reconocimiento de eventos	El estudiante no comprende la noción de evento y no puede reconocerlos o dar ejemplos.	El estudiante comprende intuitivamente la noción de evento, pero tiene problemas para dar ejemplos de pares evento-respuesta.	El estudiante comprende la noción de evento y puede dar ejemplos y reconocerlos sin problemas.
Objetos en Scratch	El estudiante no relaciona los objetos de un proyecto en Scratch con lo que sucede en la pantalla de ejecución.	El estudiante comprende el rol de los objetos en Scratch, pero tiene problemas para posicionarlos en la pantalla y asignarles instrucciones.	El estudiante es capaz de posicionar objetos en la pantalla y asignarles instrucciones.



Secuencia Didáctica 2

CONDICIONALES

En esta secuencia didáctica se introduce la noción de condicionalidad. Las **condiciones** son enunciados, o bien verdaderos, o bien falsos. Una vez comprendido esto, nos acercaremos al uso de las sentencias condicionales en programación. Los lenguajes nos permiten preguntar sobre una condición y generar una acción determinada de acuerdo con su verdad o falsedad.

En la primera actividad de la secuencia no usaremos computadoras. El objetivo es presentar la noción de sentencia condicional. En la segunda actividad, continuaremos el proyecto de la máquina de dibujar, incluyendo el uso de sentencias condicionales en el programa.

.....

OBJETIVOS

- Introducir la noción de condicionalidad.
- Presentar verdadero y falso como valores posibles de una condición.
- Usar sentencias condicionales en un proyecto de programación.

.....

Actividad 1

La carrera de los palos españoles¹



GRUPAL (4)

OBJETIVOS

- Reconocer que ciertos enunciados pueden ser verdaderos o falsos.
- Evaluar condiciones.
- Mostrar que las sentencias condicionales permiten expresar que algo debe suceder solo bajo ciertas condiciones.

MATERIALES

- Varios mazos de cartas españolas
- Varios grupos de 40 porotos

DESARROLLO

Comenzamos preguntando a la clase: “¿Es cierto que entre ustedes hay una persona que se llama Felipe?”. Solo por un momento, supondremos que sí hay alguien de nombre Felipe. Entonces, mientras lo señalamos, le decimos a la clase: “¿Qué hubieran respondido si no hubiese ningún Felipe en el curso?”². Probablemente alguno sugiera que, en tal caso, habría dicho que no había nadie con ese nombre en el curso.

Reflexionamos con los estudiantes acerca de que la frase “Felipe forma parte del curso” es cierta en algunos cursos y falsa en otros. Guiamos la discusión de forma tal de arribar a la conclusión de que hay sentencias que tienen esa propiedad: a veces pueden ser verdaderas y a veces pueden ser falsas. Mencionamos otros ejemplos, como “ahora está lloviendo” o “en este curso hay más chicos que chicas”, y analizamos con la clase el valor de verdad de las frases. Pensamos, además, algunas situaciones hipotéticas en las que cada frase sea cierta y otras en las que no lo sea. Escribimos **condición** en el pizarrón mientras les contamos a los estudiantes que una condición es un enunciado que, o bien es cierto, o bien es falso. Siempre posee un valor de verdad.

Les pedimos a los estudiantes que formen grupos de cuatro integrantes y repartimos un mazo de cartas y un conjunto de porotos a cada grupo. Les comentamos que van a jugar a la carrera de los palos españoles. Es un juego muy sencillo. Uno de los estudiantes mezcla el mazo durante un lapso breve y, a continuación, el jugador que se encuentra a su izquierda corta el mazo en dos³. Quien reparte descarta entonces la parte superior del corte, y luego va dando vuelta una a una las cartas del pilón que conservó. Cada vez que aparece una espada, el que mezcló gana un poroto; por cada basto, el poroto va para el primer compañero a su derecha; con un oro, para el que sigue hacia la derecha; y con una copa, el poroto es para el estudiante restante. Una vez que se acaban los naipes del pilón, cuentan cuántos porotos consiguió cada uno. Quien haya levantado más, gana.



¹ Adaptación de la actividad “Conditionals with Cards” de Code.org, disponible en <https://goo.gl/FyyJCK>.

² En caso de que no haya un Felipe en el curso, podemos formular la pregunta: “¿Qué hubiesen respondido si tuviesen un compañero llamado Felipe?”.

³ El corte es como el que se realiza al repartir las cartas en el truco. Sin embargo, en la carrera de los palos españoles, se descarta la parte superior del corte.

Luego de jugar un rato, preguntamos: “¿Podrían explicar qué determina quién suma un poroto cada vez que dan vuelta una carta?”. Es probable recibir respuestas tales como “la carta” o “el palo”. Comentamos con los estudiantes que, según el palo de la carta descubierta, uno u otro suma puntos. Entonces preguntamos: “Si tuvieran que escribir una regla que refleje la forma de ganar porotos de quien reparte, ¿cómo lo harían?”. Conducimos la charla para llegar a una frase del estilo: “Si sale espadas, quien reparte levanta un poroto” y la escribimos en el pizarrón.

Una vez consensuada la regla, charlamos con los estudiantes acerca de que, en cada mano, ellos evaluarán una **condición** cuyo **valor de verdad** determinó si el repartidor de la baraja recogía o no un poroto. Escribimos en el pizarrón las tres reglas restantes para sumar porotos.



Analizamos estas sentencias con los estudiantes, haciéndoles notar que todas poseen una estructura muy similar, que copiamos en el pizarrón.

SI CONDICIÓN, ENTONCES ALGO.

Les pedimos a los estudiantes que copien en una hoja las cuatro reglas e identifiquen en cada una de ellas cuál es la condición y qué es lo que ocurre cuando esta es verdadera. Deberían poder identificar que, en la primera regla, “la carta es de espadas” es una condición y “el jugador 1 gana un poroto” es lo que sucede cuando la condición se cumple. A continuación, tendrán que identificar las partes de las tres reglas restantes.

CIERRE

Comentamos con los estudiantes que todos los lenguajes de programación tienen construcciones que nos permiten evaluar condiciones y realizar ciertas acciones dependiendo de si lo evaluado es verdadero o falso. Esto es lo que nos permite incorporar a nuestros programas la toma de decisiones.

Actividad 2

La máquina de dibujar 2.0



DE A DOS

OBJETIVO

- Escribir un programa que incluya sentencias condicionales.

MATERIALES



Computadoras



Scratch

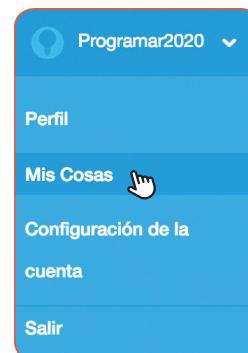


Ficha para estudiantes

DESARROLLO

Comenzamos la clase pidiendo a los estudiantes que carguen el entorno de Scratch y recuperen el proyecto de la máquina de dibujar. En caso de utilizar el editor en línea, en primer lugar tienen que iniciar sesión, y luego ir a la opción *Mis cosas*, que se encuentra en el menú del usuario, en la parte superior derecha del entorno.

Allí podrán seleccionar el proyecto. Por último, para ver sus programas, tienen que presionar el botón *Ver dentro*.



Con lo que programamos hasta el momento, la máquina de dibujar permite que la pelota salga parcialmente de los límites de la escena. Repartimos la ficha de la actividad y les contamos a los estudiantes que la pelota, si toca el borde, debe avisar de la situación con un sonido y reubicarse dentro de la escena. Repartimos la ficha y los alentamos a explorar el entorno en busca de los bloques adecuados para que sus programas alcancen el objetivo planteado. Para resolver el desafío hay que usar el bloque `si < > entonces` de la categoría *Control*. Este bloque permite chequear si una condición se cumple y, en caso de que así sea, especificar qué acciones debe realizar nuestro programa.



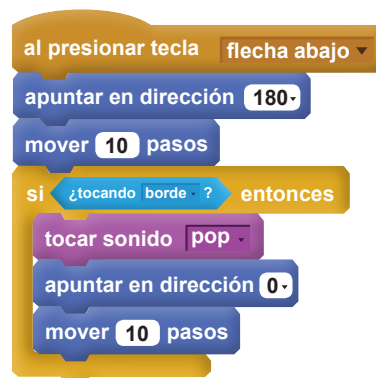
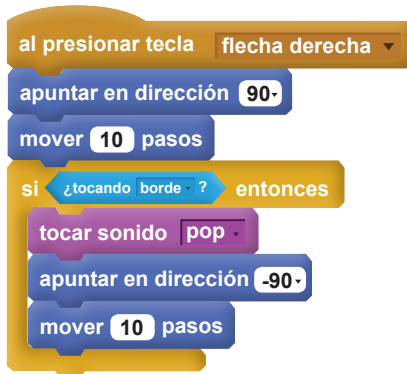
En esta actividad, una vez que movimos la pelota, la condición que debemos chequear es si el objeto alcanzó el borde de la escena. En la categoría *Sensores* se encuentra el bloque `¿Tocando []?`, que tiene `borde` como una de las opciones disponibles.



Primero, debemos conseguir que el programa emita un sonido de alerta si el objeto toca el borde. En la categoría *Sonido* está disponible el bloque `tocar sonido []` que sirve para este propósito.

tocar sonido pop

Finalmente, luego de dar la señal de alarma, hacemos retroceder la pelota 10 pasos para que vuelva a ubicarse completamente dentro de la escena. Notemos que, para hacer que retroceda, primero hay que apuntar la bola en el sentido opuesto al último movimiento. A continuación, observamos los cambios que debemos efectuar en el programa para alcanzar el objetivo planteado.



Instrucciones para que la pelota no salga de la escena

Una vez que los estudiantes hayan terminado sus programas, les recordamos que guarden la nueva versión del proyecto.

RÚBRICAS

Al final de la ficha de los estudiantes, incluimos una rúbrica para que los alumnos sepan qué estaremos evaluando y qué criterios utilizaremos.

CIERRE

Comentamos que los estudiantes usaron la sentencia condicional `si < > entonces` para resolver la actividad. Evaluaron la condición “la pelota está tocando un borde” e hicieron que sus programas siguieran ciertas instrucciones en caso de que fuera cierta. Scratch, además, ofrece otra forma para que los programas tomen decisiones. El bloque `si < > entonces si no` nos permite definir qué hacer tanto cuando una condición es verdadera como cuando es falsa.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

LA MÁQUINA DE DIBUJAR 2.0



La máquina de dibujar necesita algunos ajustes. ¡No puede ser que, una vez que la pelota llegue al borde, la dejemos seguir avanzando!

1. Lo primero que tenés que hacer es dar una señal de alarma cuando la pelota, al avanzar, se caiga fuera de la hoja blanca. ¿Te animás a incluir sonidos en tu programa? ¿Qué bloques usaste?

2. Además, no podemos darnos el lujo de dejar la pelota fuera de cuadro. Cada vez que detectes que la pelota se cae de la hoja, tenés que volver a posicionarla completamente dentro. ¿Qué instrucciones usaste?

UNA AYUDITA

Para resolver la actividad, explorá los bloques disponibles en las categorías **Control**, **Sonido** y **Sensores** de Scratch.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

EVALUACIÓN

Estas son las tareas y la forma de calificar que tu docente considerará para evaluar cómo resolvés las actividades.

CALIFICACIÓN	ES NECESARIO TRABAJAR MÁS	BUENO-MUY BUENO	EXCELENTE
Reconocimiento de eventos	El estudiante no comprende la noción de evento y no puede reconocerlos o dar ejemplos.	El estudiante comprende intuitivamente la noción de evento, pero tiene problemas para dar ejemplos de pares evento-respuesta.	El estudiante comprende la noción de evento y puede dar ejemplos y reconocerlos sin problemas.
Condicionales	El estudiante tiene problemas para reconocer la condición en la sentencia condicional y para diferenciar la condición de las instrucciones a ejecutar.	El estudiante entiende la estructura de la sentencia condicional, pero tiene algunos problemas al usarla en casos prácticos.	El estudiante comprende la noción de condicional y es capaz de entender su significado con facilidad.
Uso de bloques que contienen otros bloques dentro	El estudiante tiene problemas al componer bloques.	El estudiante puede componer bloques, pero tiene problemas para entender la forma en que los bloques se relacionan.	El estudiante puede componer bloques y comprende claramente su funcionamiento.



Secuencia Didáctica 3

CICLOS

Es habitual que, al escribir programas, nos encontremos frente a la necesidad de repetir varias veces una serie de instrucciones. Los lenguajes de programación ofrecen distintas construcciones que nos permiten expresar repeticiones sin necesidad de reiterar comandos en forma explícita. En general, nos referimos a estas construcciones como **ciclos**.

En la primera actividad de esta secuencia didáctica no usaremos computadoras. Los objetivos son dos: que los estudiantes reconozcan patrones repetidos en secuencias de instrucciones, y que puedan expresar estas secuencias en forma compacta. En la segunda actividad, retomaremos el proyecto de la máquina de dibujar, incluyendo el uso de ciclos para dibujar algunas figuras geométricas.

.....

OBJETIVOS

- Reconocer patrones.
- Expresar en forma compacta secuencias con instrucciones que se repiten.
- Usar ciclos en un proyecto de programación.

.....

Actividad 1

Ay, ay, ay es tres veces ay¹



DE A DOS

OBJETIVOS

- Reconocer patrones en secuencias.
- Introducir la noción de ciclo.
- Usar repeticiones para describir secuencias de instrucciones.

MATERIALES

Lápiz o bolígrafo

Ficha para estudiantes

DESARROLLO

Comenzamos la clase pidiendo un voluntario. El estudiante que se ofrezca deberá seguir una serie de instrucciones que le iremos dando. Le pedimos que se levante y nos quedamos esperando hasta que se pare. A continuación, le pedimos que se siente y nos mantenemos en silencio hasta que lo haga. Repetimos dos veces más ambas instrucciones de modo que el voluntario se pare y se siente tres veces en total. Entre pedido y pedido, es importante esperar hasta que el estudiante complete la acción indicada. Una vez hecho esto, liberamos al voluntario.

Le pedimos a la clase que nos diga todas las instrucciones que dimos, una por una, al estudiante que se prestó al juego. Es esperable que surja alguna respuesta que intercale pedidos para pararse con otros para sentarse. Mientras responden, vamos escribiendo las instrucciones en el pizarrón, una después de la otra.

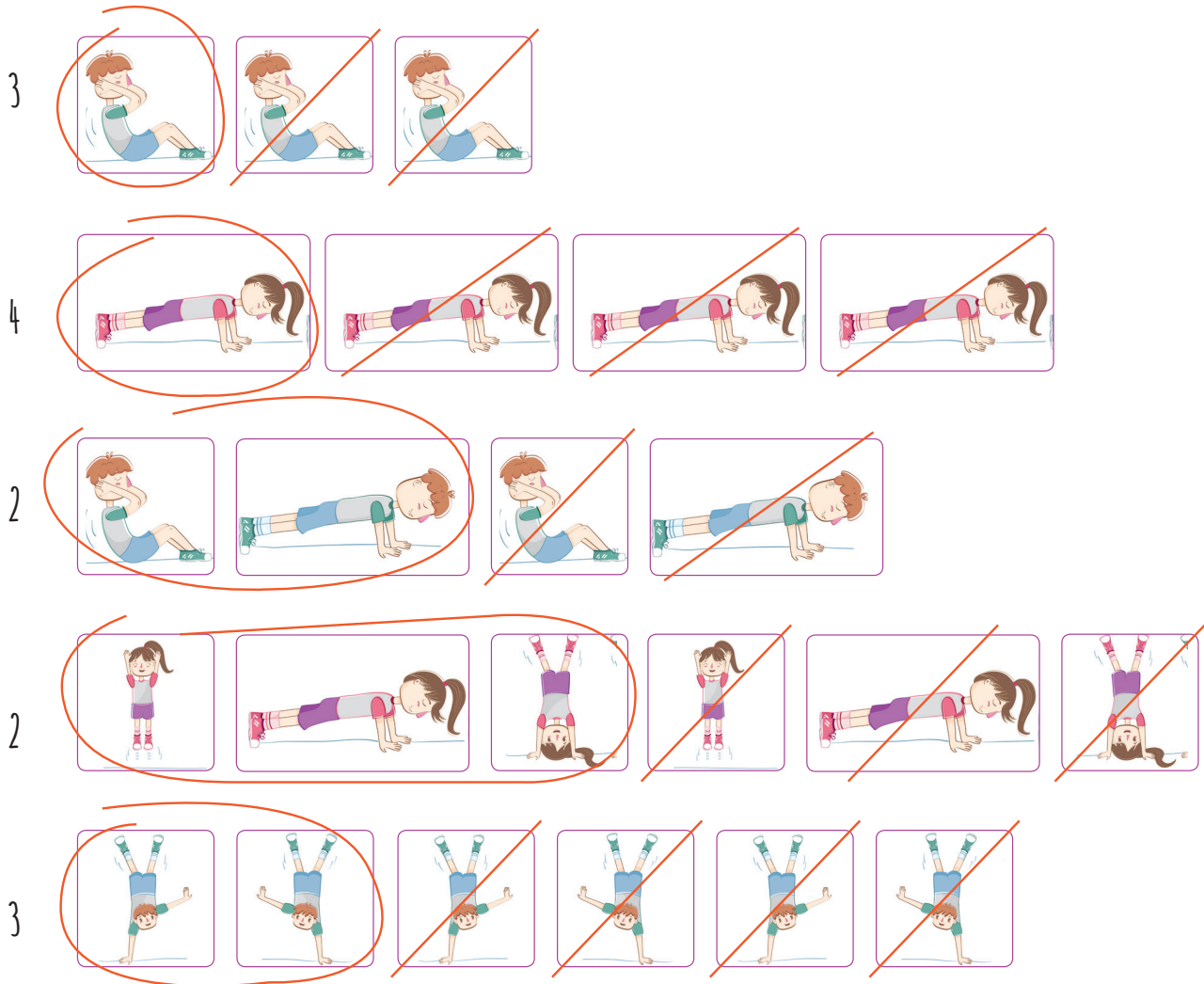
PARARSE SENTARSE PARARSE SENTARSE PARARSE SENTARSE

Preguntamos al grupo: “¿Cuántas veces pedimos que se pare el compañero?”. Tres. “¿Y cuántas veces pedimos que se siente?”. Tres también. “Si miramos todas las instrucciones, ¿encuentran algo que se repita? ¿Se les ocurre una forma más corta de decir lo que estas instrucciones expresan?”. Guiamos la reflexión de los estudiantes de forma tal de concluir que lo que hicimos fue repetir tres veces dos instrucciones: pararse y sentarse. Escribimos en el pizarrón un **3** a la izquierda del primer **Pararse**, dibujamos un círculo sobre las dos primeras palabras y tachamos desde el segundo **Pararse** en adelante.

3 (PARARSE SENTARSE) PARARSE SENTARSE PARARSE SENTARSE

Repartimos la ficha a los estudiantes y les pedimos que resuelvan la consigna en parejas. En la ficha se presentan varias rutinas aeróbicas, cada una compuesta por una serie de ejercicios que se repiten. Tal como hicimos en el pizarrón, los estudiantes deben redondear la primera aparición de la secuencia repetida, anotar el número de veces que se repite y tachar el resto de las indicaciones. A continuación se dan las soluciones.

¹ Adaptación de la actividad “Getting Loopy” de Code.org, disponible en <https://goo.gl/Z9dG7z>.



Rutinas de ejercicios de la actividad

CIERRE

Reflexionamos con los estudiantes sobre lo que hicieron en la actividad. En primer lugar, identificaron patrones en secuencias de instrucciones. Luego, expresaron en forma compacta lo descrito por la secuencia completa, en este caso, usando repeticiones. Les contamos que los lenguajes de programación también nos dan la posibilidad de expresar repeticiones en forma compacta.

NOMBRE Y APELLIDO:

CURSO:

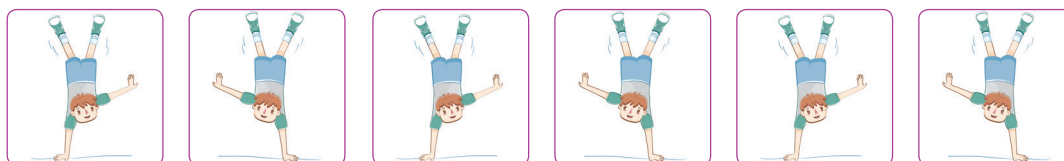
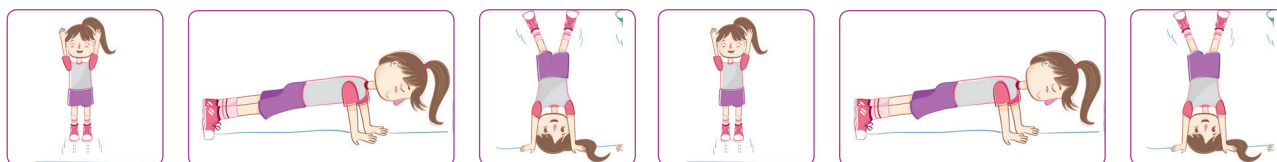
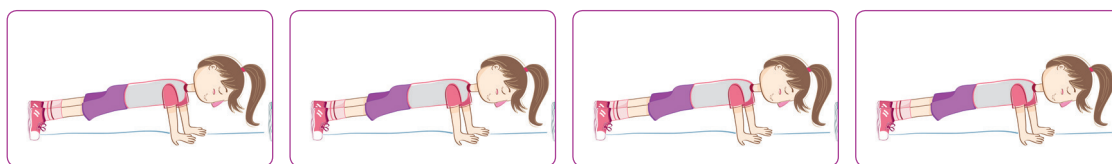
FECHA:

AY, AY, AY ES TRES VECES AY

Suele decirse que la actividad física tiene efectos positivos en la salud. Los entrenadores recomiendan diversas rutinas que nos permiten alcanzar la plenitud física. Acá abajo vas a encontrar varias rutinas de ejercicios. Sin embargo, vas a ver que las descripciones son bastante extensas. ¿Podés escribirlas de modo más breve? La primera ya está resuelta.

VIRUS Y PATRONES

¿Sabías que los antivirus exploran el contenido de los archivos buscando ciertos patrones que contienen algunos de los archivos infectados?



Actividad 2

La máquina de dibujar 3.0

DE A DOS

OBJETIVO

- Utilizar ciclos en un proyecto de programación.

MATERIALES

 Computadora

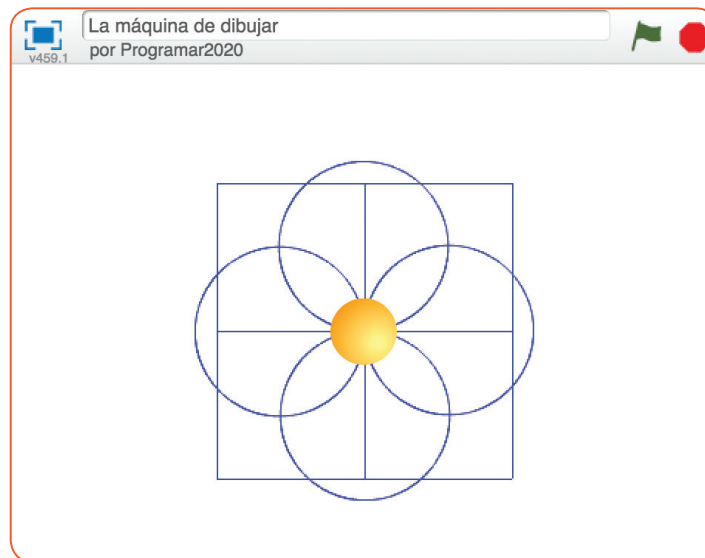
 Scratch

 Ficha para estudiantes

DESARROLLO

Comenzamos la actividad invitando a un estudiante a que dibuje un cuadrado en el pizarrón. Preguntamos a la clase : “¿Cómo dibujó su compañero el cuadrado?”. Se espera que contesten que lo hizo de a un lado por vez. Entonces, hacemos esta nueva pregunta: “¿Notan algo que se repita?”. Conducimos la conversación para que surja la respuesta: “Se repitió 4 veces dibujar un lado”.

Les pedimos que carguen el entorno de Scratch y recuperen el proyecto de la máquina de dibujar. Repartimos la ficha y les explicamos que ahora tendrán que incorporar la capacidad de dibujar cuadrados y círculos a la máquina. Concretamente, deben lograr que, cuando se presione la tecla 3, el programa reaccione dibujando un cuadrado y cuando se apriete el 4, un círculo.



La máquina de dibujar 3.0

Alentamos a los estudiantes a explorar el entorno y pensar cómo resolver el desafío. Si hiciera falta orientar la búsqueda, podemos sugerir revisar la categoría **Control**. Allí se encuentra el bloque `repetir ()`, que es de gran utilidad para resolver la actividad. Mientras los estudiantes están trabajando, nos acercamos a cada grupo para guiarlos si hiciese falta.

Dibujar un cuadrado consiste en dibujar sus cuatro lados. Teniendo en cuenta que los objetos en la escena de Scratch apuntan siempre en una dirección, una posibilidad es comenzar dibujando una línea en ese sentido. A continuación, tenemos que rotar la pelota 90° de forma que quede bien orientada para dibujar el siguiente lado. Si repetimos estas dos instrucciones cuatro veces, conseguimos dibujar los cuatro lados, es decir, el cuadrado completo. Para ilustrar la actividad usaremos un largo de lado de 100 pasos. Variando este número se obtienen cuadrados de otros tamaños.

En caso de que alguna pareja de estudiantes haya construido un programa sin usar el bloque `repetir ()`, les sugerimos que busquen un patrón que se repita en su propuesta. La secuencia de instrucciones `[mover (100) pasos, girar (90) grados]` aparece cuatro veces. Combinando una única aparición de estas instrucciones con el bloque `repetir ()` arribamos a la solución explicada en el párrafo anterior.

Es probable que los estudiantes no puedan deducir cómo dibujar un círculo. Si así fuera, sugerimos que esta parte de la actividad sea más asistida. En este caso, tenemos que movernos una distancia muy corta y rotar la pelota muy levemente. Estas dos instrucciones deben repetirse hasta que el giro haya alcanzado los 360°. A modo de ejemplo, podemos mover la pelota un paso, girarla un grado y repetir estas dos instrucciones 360 veces.



Instrucciones para dibujar un cuadrado

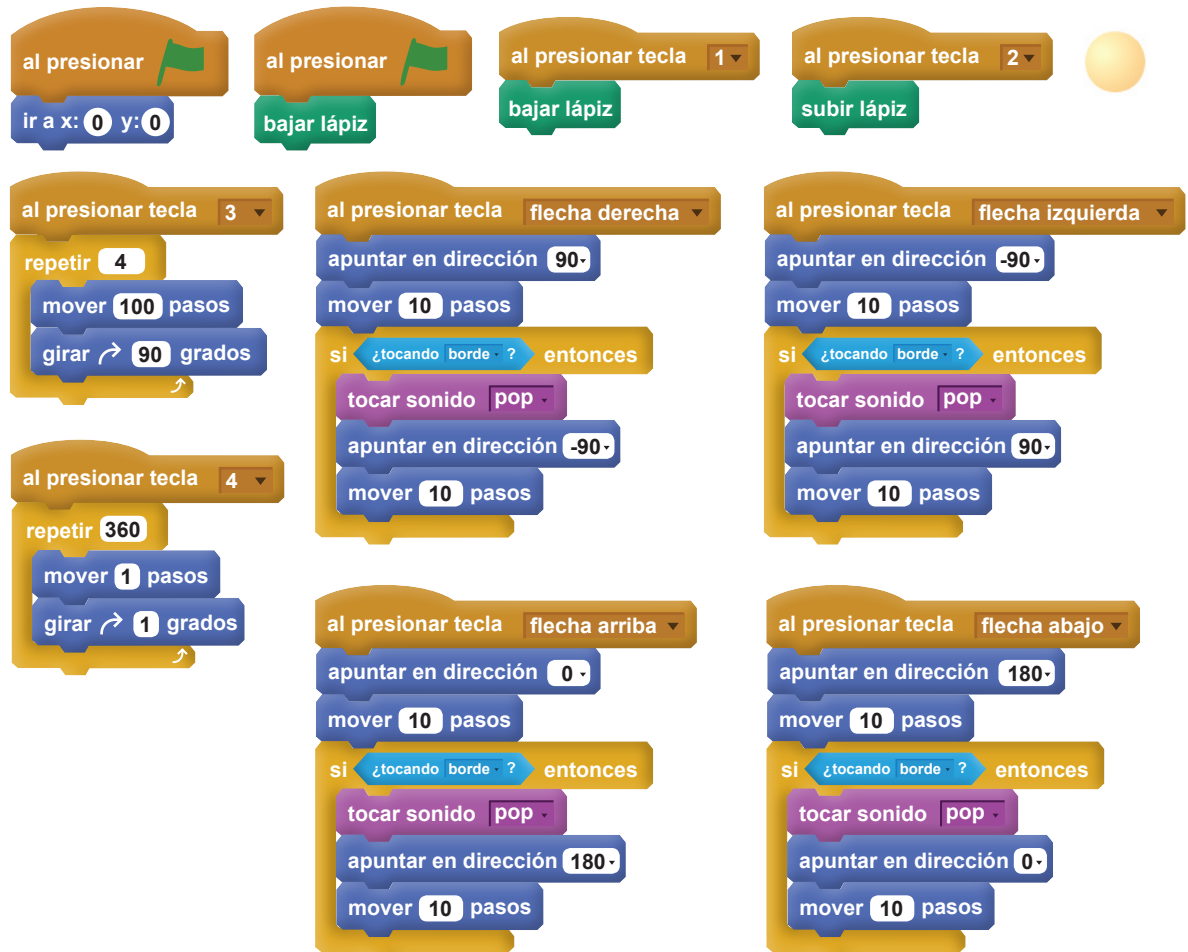


Posible propuesta de estudiantes para dibujar un cuadrado



Instrucciones para dibujar un círculo

Una solución completa de la nueva versión de la máquina para dibujar se observa a continuación.



Una solución completa para la máquina de dibujar 3.0

RÚBRICAS

Al final de la ficha de los estudiantes, incluimos una rúbrica para que los alumnos sepan qué estaremos evaluando y qué criterios utilizaremos.

CIERRE

Reflexionamos con los estudiantes sobre la conveniencia de expresar en forma compacta secuencias de instrucciones que se repiten. Si no existiese la instrucción `repetir ()`, estaríamos condenados a construir programas mucho más extensos y mucho más difíciles de comprender. Por ejemplo, si viésemos un programa que tiene 360 ocurrencias de las instrucciones `mover (1)` y `girar (1)`, ¿nos resultaría sencillo darnos cuenta a simple vista de que son instrucciones para dibujar un círculo?

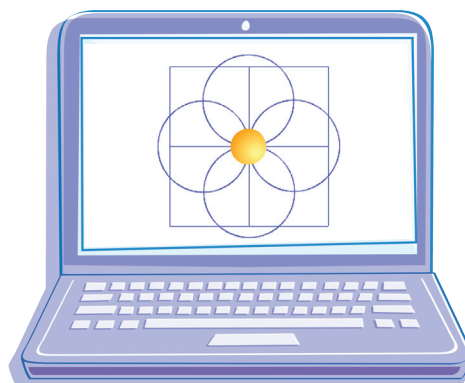
NOMBRE Y APELLIDO:

CURSO:

FECHA:

LA MÁQUINA DE DIBUJAR 3.0

¡Ya salió una nueva versión de la máquina de dibujar!
¿La novedad? Ahora dibuja figuras geométricas.



1. Tenés que lograr que al presionar la tecla la máquina dibuje un cuadrado. Cada lado debe medir 100 pasos. ¿Usaste algún bloque nuevo? ¿Cuál?

2. Al presionar la tecla , tenés que conseguir que esta versión del programa dibuje un círculo. ¿Cómo lo hiciste?

NOMBRE Y APELLIDO:

CURSO:

FECHA:

EVALUACIÓN

Estas son las tareas y la forma de calificar que tu docente considerará para evaluar cómo resolvés las actividades.

CALIFICACIÓN	ES NECESARIO TRABAJAR MÁS	BUENO-MUY BUENO	EXCELENTE
Reconocimiento de patrones	El estudiante no es capaz de identificar las similitudes entre situaciones diferentes que siguen un patrón.	El estudiante reconoce patrones simples.	El estudiante reconoce patrones fácilmente.
Ciclos	El estudiante no comprende la noción de ciclo. Tiene problemas al abstraer una secuencia de instrucciones repetidas en un ciclo.	El estudiante comprende la noción de ciclo, pero tiene problemas para abstraer instrucciones repetidas en un ciclo.	El estudiante comprende la noción de ciclo y puede abstraer instrucciones repetidas en un ciclo.
Reconocimiento de eventos	El estudiante no comprende la noción de evento y no puede reconocerlos o dar ejemplos.	El estudiante comprende intuitivamente la noción de evento, pero tiene problemas para dar ejemplos de pares evento-respuesta.	El estudiante comprende la noción de evento y puede dar ejemplos y reconocerlos sin problemas.
Uso de bloques que contienen otros bloques dentro	El estudiante tiene problemas al componer bloques.	El estudiante puede componer bloques, pero tiene problemas para entender la forma en que los bloques se relacionan.	El estudiante compone bloques y comprende claramente su funcionamiento.

PROCEDIMIENTOS

Al incrementar el tamaño de un programa, es imperativo hacerlo de manera ordenada y prolija. De lo contrario, la claridad inicial puede ofuscarse y nuestras creaciones se transformarán en objetos de *software* de difícil decodificación.

SECUENCIA DIDÁCTICA 1

INVASORES DEL ESPACIO, EL COMIENZO

Programamos la nave principal

Programamos las naves enemigas

Disparamos el proyectil

SECUENCIA DIDÁCTICA 2

PROCEDIMIENTOS Y PARÁMETROS

Procedimientos para dibujar

Estribillos como procedimientos

Programamos canciones en Scratch

SECUENCIA DIDÁCTICA 3

INVASORES DEL ESPACIO,

LA CONTINUACIÓN

Procedimientos en Space Invaders

Corregimos y testeamos el movimiento del proyectil

Eliminamos naves enemigas

Sumamos puntos

Una estrategia para lidiar con el problema de un programa de gran tamaño consiste en dividirlo en partes más pequeñas de forma tal que, al combinar las soluciones, se pueda resolver el problema original. En las actividades de este capítulo ponemos en práctica esta idea usando una construcción presente en la gran mayoría de los lenguajes de programación: los **procedimientos**.

En la primera secuencia didáctica introducimos el proyecto Invasores del espacio, que se desarrollará a lo largo del capítulo. Al hacerlo, se ejercitarán algunos conceptos trabajados en capítulos anteriores. En la segunda secuencia, se presenta la noción de procedimiento. Los conceptos se abordan primero en actividades sin computadora y, luego, se ponen en juego en un pequeño programa. En la tercera y última secuencia del capítulo, retomamos el desarrollo de Invasores del espacio, para obtener una versión muy similar a la del clásico juego de los años ochenta, Space Invaders.



Secuencia Didáctica 1

INVASORES DEL ESPACIO, EL COMIENZO

En esta secuencia didáctica los estudiantes comenzarán a programar un juego inspirado en el clásico de los años ochenta, Space Invaders. A partir del desarrollo del videojuego, profundizaremos algunos conceptos trabajados en los capítulos previos. El proyecto comenzará a crecer en complejidad y tamaño, lo que motivará la conveniencia de definir nuestros propios procedimientos. Una implementación completa del proyecto se encuentra disponible en <https://goo.gl/Swyy8R>.

.....

OBJETIVOS

- Ejercitar el manejo de eventos.
- Exponer debilidades de la programación sin procedimientos.

.....

Actividad 1




Programamos la nave principal

 DE A DOS

OBJETIVOS

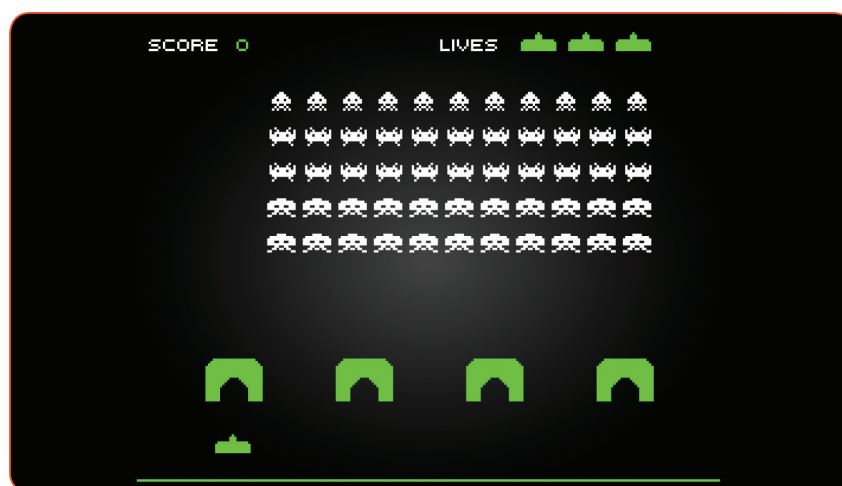
- Poner en práctica el manejo de eventos.
- Introducir las nociones de parámetro y argumento.
- Establecer el estado inicial de un programa.

MATERIALES

-  Computadora
-  Scratch
-  Juego Space Invaders
-  Ficha para estudiantes
-  Proyector (opcional)

DESARROLLO

Comenzamos la clase preguntando a los estudiantes: “¿Qué consolas de videojuegos conocen? ¿Sabían que cuando yo era chico no existían la Play 4 ni la Xbox? Es más, no existían la Play 3, ni la 2, ni tampoco la 1. ¿Conocen el famoso juego Space Invaders? ¿Les gustaría jugarlo y conocer cómo eran los videojuegos hace treinta años?”. Luego, invitamos a los estudiantes a probar durante 10 minutos el Space Invaders, para que conozcan la dinámica y se familiaricen con ella. Puede jugarse en línea, por ejemplo en <https://goo.gl/F4irUW>.



Pantalla principal del videojuego Space Invaders

A continuación, les pedimos que respondan las preguntas de la primera consigna de la ficha, donde tienen que identificar: (i) cuántos personajes hay en el videojuego; (ii) el personaje que utiliza el jugador; (iii) el objetivo del juego; (iv) el comportamiento de la nave verde; y (v) cómo lograr que la nave se mueva. La nave de color verde es la que utiliza el jugador, mientras que las de color blanco son las naves enemigas. El objetivo del juego es eliminar todas las naves enemigas. La nave verde, comandada por el usuario, se mueve únicamente al presionar las flechas izquierda y derecha del teclado. Además, usando la barra espaciadora se disparan proyectiles. Ayudamos a los estudiantes a relacionar estas acciones con el concepto de evento, visto en el capítulo anterior.

Como desarrollarán versiones propias del videojuego Space Invaders, no es necesario utilizar los mismos personajes del juego original. Sin embargo, en esta exposición sí se utilizan los personajes originales, solo en calidad de muestra. En caso de querer distribuir estas imágenes entre los estudiantes para que las usen en sus proyectos, pueden descargarse del proyecto disponible en <https://goo.gl/Swyy8R>. Una vez dentro, hay que seleccionar cada imagen en la galería de *objetos*, hacer clic en la solapa *Disfraces*, presionar el botón derecho del ratón sobre la imagen del objeto,

y seleccionar la opción *guardar a un archivo local*. De esta forma, se descargará un archivo con extensión .png (por ejemplo, “nave principal.png”). Luego, cuando los estudiantes tengan estos archivos en sus computadoras, podrán incorporarlos en sus proyectos haciendo clic sobre el ícono de carga de objetos del menú de la galería.

Para que los estudiantes puedan comenzar con la creación del videojuego, es necesario que empiecen por el punto a de la consigna 2. Allí se les pide que elijan el personaje que llevará a cabo el rol de la nave verde. Pueden elegir un personaje de la galería de *objetos* o usar alguna imagen propia. En caso de que el tamaño del personaje elegido sea demasiado grande o demasiado chico en relación con el tamaño de la escena, se pueden usar los controles de tamaño para redimensionarlo. Primero hay que hacer clic sobre el control, y luego algunos clics sobre el objeto que queremos cambiar de tamaño.



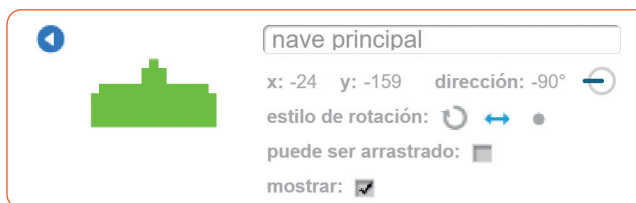
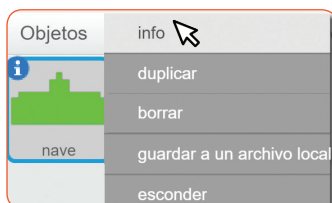
Nuevo objeto:

Exportación e importación de imágenes



Controles para redimensionar objetos

Es recomendable habituarse a ponerle un nombre a cada objeto que se agrega a la escena. Para hacerlo, se presiona el botón derecho del ratón sobre la imagen elegida en la galería de *Objetos* y se selecciona la opción *info*. Entonces vemos un panel en el que podemos editar el nombre. Al objeto agregado lo llamamos *nave principal*. Para volver a ver la galería completa, presionamos la flecha azul del panel.



Cuadro info de la nave principal

Al construir programas, es recomendable pensar en condiciones iniciales adecuadas: ¿dónde deben ubicarse los personajes que intervienen en la escena al comenzar?, ¿tienen que apuntar en alguna dirección en particular?, ¿deben estar visibles? Estas son algunas de las preguntas que podemos hacernos para pensar cómo establecer condiciones iniciales. Cuando comienza la ejecución del programa, tenemos la posibilidad de ordenar el mundo que estamos creando para que cumpla con las condiciones pensadas. Como en Scratch las ejecuciones comienzan cada vez que se presiona la bandera verde, manejaremos ese evento para inicializar el estado del programa. Una forma prolija de hacerlo es que cada objeto inicialice todo lo correspondiente a él mismo. Es decir que usaremos tantos bloques **al presionar bandera verde** como personajes intervengan en la escena. Por ahora, solo nos ocuparemos de la nave verde, pero repetiremos el procedimiento a medida que vayamos poblando de objetos nuestro juego. En este caso, alcanza con ubicar la nave en una posición inicial y hacerla visible. Una buena opción es posicionarla en el centro del borde inferior de la escena, concretamente usando el bloque **ir a x: (0) y: (-160)**.

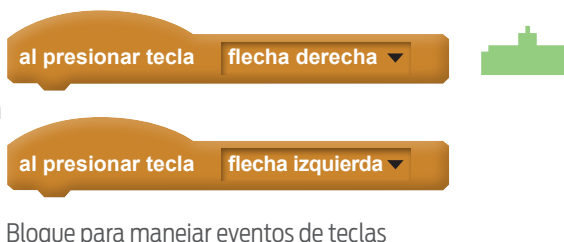


x: 0
y: -160

Inicialización de la nave principal

En el punto b de la consigna 2 se les solicita a los estudiantes que identifiquen los eventos que hacen que la nave verde se desplace; en este caso, hay que presionar las flechas derecha e izquierda del teclado.

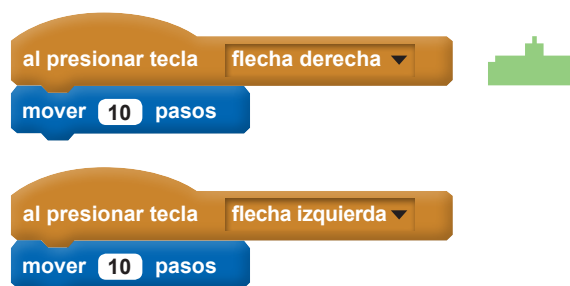
Luego de unos minutos, les preguntamos a los estudiantes si pudieron encontrar algún bloque que les permitiera manejar los eventos identificados. Deberían haber encontrado `al presionar tecla []`, en el cual `flecha derecha` y `flecha izquierda` son dos opciones. Se encuentra en la categoría *Eventos*.



Bloque para manejar eventos de teclas

Cuando los estudiantes hayan agregado el personaje e identificado los eventos, preguntamos: “¿Con estas instrucciones logramos que la nave se mueva? ¿Qué les parece?”. En el caso de contar con un proyector o televisor, mostramos en la pantalla lo que se describe a continuación. “Como podemos ver, nuestro personaje principal no se mueve. Por el momento, solo agregamos dos bloques que nos permiten identificar si ocurren dos eventos específicos: presionar la flecha derecha y presionar la flecha izquierda. Pero no agregamos instrucciones para manejar estos dos eventos”. Se les pide entonces a los estudiantes que resuelvan el punto c de la consigna 2 de la ficha, para que el personaje se pueda mover a la izquierda o a la derecha, dependiendo de la tecla que presionen.

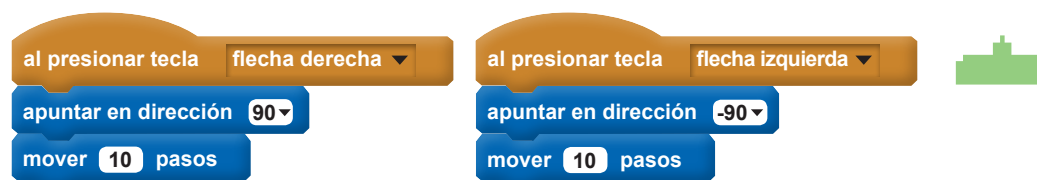
Luego de unos minutos, preguntamos a los estudiantes si pudieron resolver el desafío. Es probable que hayan encontrado el bloque `mover (10) pasos`, pero que los programas creados no funcionen de la manera esperada. Es posible que hayan tratado de resolver el desafío tal como se muestra en la figura de la derecha.



Solución incorrecta para el punto c de la ficha 1

En caso de contar con un proyector, pedimos a alguno de los estudiantes que conecte su computadora y muestre el programa de la nave verde. Luego, le solicitamos que lo ejecute y pruebe qué sucede. Tanto al presionar la flecha derecha como la izquierda, el personaje se mueve en ambos casos 10 pasos a la derecha. Reflexionamos con los estudiantes: “Si prestamos atención, veremos que hemos encastrado el mismo bloque para responder ante ambos eventos, que suceden cuando se presiona la flecha derecha o la izquierda. Por lo tanto, al oprimir cualquiera de ellas se ejecutará la misma instrucción. Recordemos que los eventos nos permiten identificar ciertos sucesos, pero la reacción del programa ante ellos tenemos que programarla nosotros”.

En la categoría *Movimiento* del panel de Scratch se encuentra el bloque `apuntar en dirección()`. El comportamiento de este bloque depende de un valor que se puede modificar; es decir, el bloque tiene un parámetro, y cada vez que lo usemos deberemos indicarle un valor al que llamamos argumento. Los valores que ofrece Scratch por defecto son - 90 (izquierda), 0 (arriba), 90 (derecha), y 180 (abajo). Cada uno representa una dirección específica. Para asegurarnos de que el personaje se mueva hacia una dirección en particular, antes de moverlo debemos hacerlo apuntar en la dirección deseada. Preguntamos a los estudiantes: “¿Si queremos que apunte a la derecha, qué argumento debemos elegir? ¿Y para que apunte a la izquierda?”. En la siguiente figura se muestra el programa para que la nave verde se mueva de la forma esperada



Programa correcto para el movimiento de la nave verde

CIERRE



Compartimos lo siguiente con los estudiantes: “En su vida cotidiana están rodeados de eventos. Por ejemplo ¿cómo se ejecutan Clash Royale, WhatsApp o Pokémon Go? Esperan que un evento específico suceda; en este caso, que los usuarios toquen el ícono de la aplicación”. Reflexionamos juntos sobre cómo, utilizando eventos, se logró que el personaje principal se moviera. Repasamos el modo en que identificamos diferentes sucesos y cómo programamos la reacción del programa cuando ellos ocurren. Les explicamos: “Además, se utilizaron los bloques `mover () pasos` y `apuntar en dirección ()` para lograr que la nave se moviera de la forma deseada. Estos bloques tienen un **parámetro** que nos permite indicar cuántos pasos y en qué dirección se mueve un objeto. Distintos valores producen un comportamiento diferente del programa. Los valores que se usan en concreto se llaman **argumentos**.”

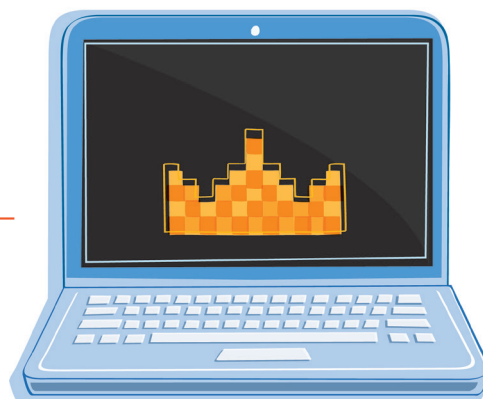
NOMBRE Y APELLIDO:

CURSO:

FECHA:

PROGRAMAMOS LA NAVE PRINCIPAL

¡Vamos a crear nuestra propia versión de Space Invaders! En esta actividad crearás el personaje principal y lograrás que se mueva hacia la derecha y hacia la izquierda cada vez que presiones las teclas flecha derecha  y flecha izquierda  respectivamente.



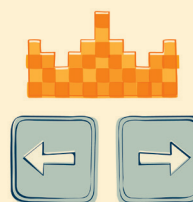
- 1.** Viajarás por el túnel del tiempo para aprender cómo eran los videojuegos de antes.
Jugá unos minutos al Space Invaders con tus compañeros y respondé:

a. ¿Cuántos personajes hay en el videojuego? ¿Cuál es el personaje que utiliza el jugador?

b. ¿Cuál es el objetivo del juego?

c. ¿Cómo es el comportamiento de la nave verde?

d. ¿Cómo lograrás que se mueva la nave verde?



VOLVIERON LOS EVENTOS

¿Qué sucede cuando presionás las flechas derecha e izquierda del teclado? ¿Y cuando apretás la barra espaciadora?

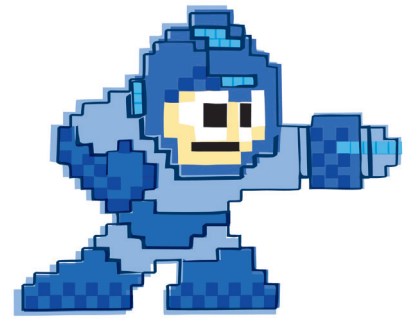
NOMBRE Y APELLIDO:

CURSO:

FECHA:

2. Creando un personaje

a. Elegí el personaje que va a cumplir el rol de la nave verde. En el caso de que quieras crear tus personajes pixelados, podés utilizar la aplicación web <https://www.piskelapp.com>. Por ejemplo, acá podés ver al famoso Megaman dibujado con Piskel.



b. ¿Cuáles son los eventos que hacen que la nave verde se mueva? Explorá Scratch para identificar un bloque que nos permita manejarlos. ¿Qué bloque encontraste?

c. Programá el personaje para que se mueva a la izquierda y a la derecha, según la tecla que se presione. Encargate, además, de que al comenzar una partida la nave esté visible y bien posicionada.

DATO A TENER EN CUENTA

Observá que el bloque `mover ()` `pasos` tiene un parámetro que nos permite establecer el número de pasos que queremos que avance nuestro personaje. El valor que efectivamente usemos se llama argumento.

mover 10 pasos

Actividad 2




Programamos las naves enemigas

 DE A DOS

OBJETIVOS

- Identificar patrones.
- Introducir ciclos que se ejecutan indefinidamente.
- Ejercitar el uso de ciclos.

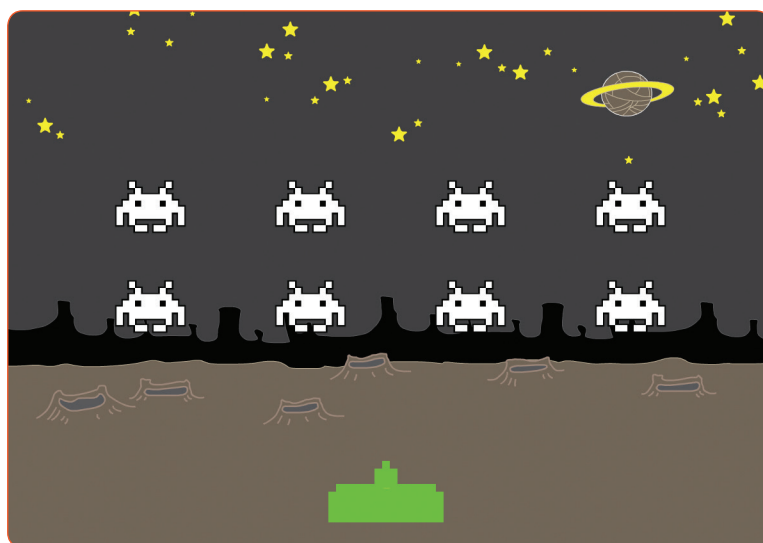
MATERIALES

-  Computadoras
-  Scratch
-  Ficha para estudiantes

DESARROLLO

En la actividad anterior, logramos que la nave verde se moviera a la derecha o a la izquierda según la tecla que se presionara. Para ello, se utilizaron bloques que permiten identificar eventos y, cuando alguno se produce, ejecutar una serie de instrucciones. También se trabajó el uso de bloques que contienen parámetros.

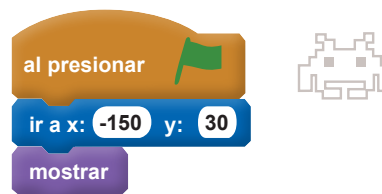
En esta actividad proponemos incorporar y programar las naves enemigas de Invasores del espacio siguiendo la ficha de la actividad. Una vez que los estudiantes hayan elegido sus personajes, como se pide en la primera consigna de la ficha, tienen que agregar dos filas de cuatro naves cada una y ubicarlas como se muestra en la figura que se encuentra a continuación. Les sugerimos que les cambien los nombres a *nave enemiga 1*, *nave enemiga 2*, y así hasta llegar a *nave enemiga 8*.



Disposición de las naves enemigas

Así como hicimos con la nave verde, también tenemos que establecer las condiciones iniciales de las naves enemigas. En primer lugar, debemos ocuparnos de posicionar los objetos, en este caso, las naves enemigas. Bajo **al presionar bandera verde**, colocamos el bloque **ir a x: () y: ()**, que se encuentra en la categoría *Movimiento*. Debemos tener en cuenta que cada nave debe ubicarse en un lugar diferente. A continuación se exhiben posibles valores iniciales para las coordenadas de cada nave enemiga.

(-150, 30)	(-50, 30)	(50, 30)	(150, 30)
(-150, -30)	(-50, -30)	(50, -30)	(150, -30)



Inicialización de las naves enemigas

Una vez establecidas las posiciones de las naves enemigas, debemos asegurarnos de que se vean. Lo hacemos usando el bloque **mostrar** de la categoría *Apariencia*.

Preguntamos a los estudiantes si recuerdan cómo se movían las naves enemigas. En caso de que no se acuerden, les sugerimos que jueguen de nuevo al Space Invaders. Les proponemos que programen el movimiento de las naves enemigas resolviendo la consigna 2 de la ficha. Para guiarlos en esta tarea, se recomienda plantear lo siguiente: “Seguramente, vamos a necesitar la secuencia de dos bloques que utilizamos en la actividad anterior para lograr que se mueva la nave verde. Dependiendo de la dirección y la cantidad de pasos que queremos que den las naves enemigas, modificaremos los argumentos de los bloques”.

A continuación, compartimos con los estudiantes la secuencia de instrucciones **apuntar en dirección (90)** seguida de **mover (10) pasos** y les consultamos si podemos usar estos mismos bloques para mover las naves enemigas.

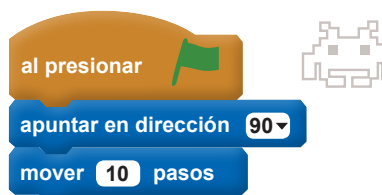


Bloques para mover objetos diez pasos hacia la derecha

Reflexionamos con los estudiantes: “Recordemos que en Scratch necesitamos identificar el evento que queremos que desencadene el movimiento de los objetos”. Proponemos que todas las naves enemigas comiencen a moverse apenas comencemos a jugar. Tendrán que usar, entonces, un segundo bloque **al presionar bandera verde**.

Una vez elegido el evento que da comienzo al movimiento de las naves enemigas, proponemos a los estudiantes que las programen para que se muevan hacia la derecha, como indica el punto 2.a de la ficha.

Luego de unos minutos, pedimos a los estudiantes que compartan de qué manera han resuelto el desafío planteado. Es muy probable que hayan incluido las instrucciones **apuntar en dirección (90)** y **mover (10) pasos**.



Solución de la consigna 2.a

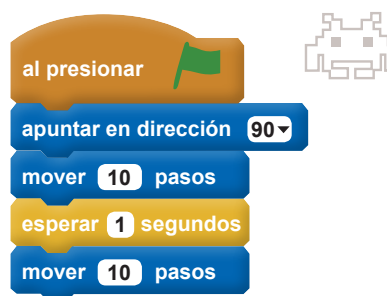
Se debe tener presente que cada vez que incorporemos instrucciones en el programa, son realizadas únicamente por el objeto que se encuentre seleccionado en la galería *Objetos*. Si los bloques se agregaron sobre una de las naves enemigas, hay una forma sencilla de copiarlos al resto de las naves blancas. Alcanza con arrastrarlos con el puntero del ratón a cada una de las naves enemigas de la galería. De esta forma, al ejecutar el programa, todas las naves enemigas se moverán simultáneamente.

En el juego original las naves se mueven, frenan un tiempo y luego se vuelven a desplazar. Les pedimos a los estudiantes que modifiquen sus programas para que las naves enemigas tengan ese comportamiento, resolviendo la consigna 2.b de la ficha. Para poder resolver el desafío se necesita el bloque `esperar (1) segundos`. Este bloque tiene un parámetro que determina la cantidad de segundos que un objeto debe esperar antes de ejecutar la siguiente instrucción. Para lograr que se mueva de manera similar al videojuego original, se pueden intercalar los bloques `mover (10) pasos` y `esperar (1) segundos`.

A continuación les proponemos a los estudiantes que programen al personaje para que se mueva cinco veces a la derecha, como indica la consigna 2.c de la ficha. Es muy probable que la mayoría de los estudiantes resuelva la actividad con un programa similar al de la figura de la derecha.



El código de una nave enemiga se copia al resto de las naves

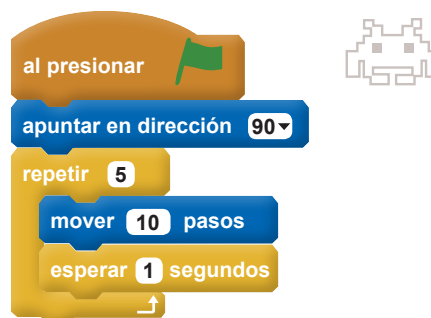


Solución de la consigna 2.b



Solución de la consigna 2.c

Observamos con los estudiantes que el patrón `mover (10) pasos, esperar (1) segundos` se repite. De esta manera, podemos recuperar el concepto de ciclo presentado en el capítulo anterior. Les pedimos que resuelvan la consigna 2.d en la que se les indica que modifiquen su solución para utilizar el ciclo `repetir ()`.



Solución de la consigna 2.d

Una vez que los estudiantes hayan logrado que las naves enemigas se muevan 5 veces a la derecha utilizando un ciclo, les proponemos el siguiente desafío: “En el juego original, las naves enemigas se mueven cierta cantidad de veces a la derecha, y luego esa misma cantidad de veces hacia la izquierda. ¿Se animan a lograr que la nave, después de desplazarse 5 veces a la derecha, se mueva 5 veces a la izquierda?”. De esta manera, se está invitando a los estudiantes a resolver el punto 2.e de la ficha.



Solución de la consigna 2.e

Finalizado el desafío, proponemos a los estudiantes que verifiquen cómo funcionan las naves blancas y les preguntamos: “¿Las naves se mueven de la misma manera que las del juego original? En parte, sí. Logramos que se muevan de una manera muy similar, pero la secuencia de movimientos primero hacia la derecha y luego hacia la izquierda solo se ejecuta una vez. En el juego original, esa secuencia se repite en forma indefinida. Necesitamos un ciclo que ejecute las instrucciones indefinidamente”. Al usar el bloque `por siempre`, las instrucciones que componen el cuerpo del bloque no dejan de ejecutarse nunca. De esta forma se completa la consigna 2.f de la ficha.



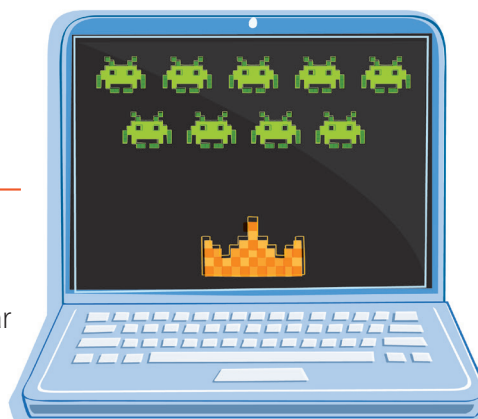
Solución de la consigna 2.f

CIERRE

A modo de cierre, planteamos a los estudiantes estas preguntas: “¿Por qué son importantes los ciclos cuando programamos?”. Se espera que lleguen a la conclusión de que nos permiten no repetir instrucciones y escribirlas una sola vez. “¿Cuál es la diferencia entre los bloques `repetir ()` y `por siempre`?”. Mientras que el primero se ejecuta una cantidad finita de veces, el segundo no termina.

PROGRAMAMOS LAS NAVES ENEMIGAS

En la actividad anterior lograste que la nave principal respondiese a dos eventos para poder moverse a la izquierda o a la derecha, dependiendo de qué tecla se presione. Ahora tenés que programar las naves enemigas para que se muevan todo el tiempo hacia la izquierda y la derecha, repitiendo un patrón determinado.



1. Diseñá o elegí a los personajes enemigos y ubicalos en el escenario. Asegurate de que, cada vez que comience una partida estén visibles y bien posicionados.



2. Programá a los personajes para que se muevan a la izquierda y a la derecha como en el Space Invaders. Si prestás atención al juego original, vas a ver que las naves se desplazan una cierta cantidad de veces a la derecha y luego hacia la izquierda. Hacerlo por partes, siguiendo las pautas de las siguientes consignas.
 - a. Buscá que las naves enemigas se muevan 10 pasos hacia la derecha.
 - b. Ahora tenés que conseguir que las naves se muevan 10 pasos, hagan una pequeña pausa, y vuelvan a moverse otros 10 pasos. Tené en cuenta que al crear una secuencia de bloques `mover () pasos`, el programa ejecuta todos los bloques de manera consecutiva, sin tiempo de espera. Para lograr que entre los movimientos haya una pausa antes de ejecutar el próximo `mover () pasos`, tenés que programarla. ¡Explorá la plataforma para encontrar un bloque que te permita detener el tiempo! ¿Qué bloque es?

NOMBRE Y APELLIDO:

CURSO:

FECHA:

c. Programá al personaje para que se desplace 5 veces a la derecha utilizando los bloques **mover () pasos** y **esperar () segundos**. ¿Cómo lo hiciste?

mover 10 pasos

esperar 1 segundos

d. Modificá el programa para que utilice un ciclo. Para esto, usá la estructura de control **repetir ()**, que se encuentra dentro de la categoría *Control*. ¿Cuántas veces desplazaste las naves enemigas?



e. Realizá los cambios necesarios para que las naves enemigas se muevan 5 veces a la izquierda después de moverse 5 veces a la derecha.

f. Por último, conseguí que las naves enemigas repitan indefinidamente los movimientos hacia la derecha y hacia la izquierda. ¿Qué bloque usaste para que las naves se desplacen de modo permanente?

CONSEJO

Como todas las naves enemigas se mueven de la misma manera, tenés que repetir las mismas instrucciones para cada una. Para evitar armar el programa para cada nave, tené en cuenta el siguiente consejo: podés arrastrar las instrucciones que ya programaste hacia los personajes que se encuentran dentro de la galería, como se observa en la imagen.



Actividad 3




Disparamos el proyectil

 DE A DOS

OBJETIVOS

- Ejercitar el uso de los ciclos.
- Programar usando sentencias condicionales compuestas.
- Realizar manejo de eventos.

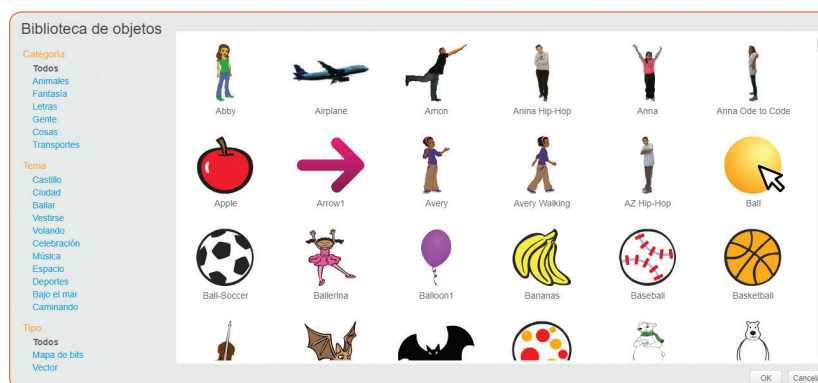
MATERIALES

-  Computadora
-  Scratch
-  Ficha para estudiantes

DESARROLLO

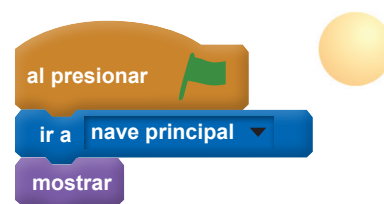
Para empezar, identificamos con los estudiantes las funcionalidades ya resueltas: el personaje principal se mueve hacia la izquierda y hacia la derecha cuando utilizamos las flechas del teclado; las naves contrarias se mueven automáticamente cada vez que se presiona la bandera verde. En esta actividad, intentaremos que la nave principal dispare proyectiles a las naves enemigas.

Empezamos proponiendo a los estudiantes que elijan el objeto *Ball*, que usaremos como proyectil para disparar a las naves enemigas. Sugerimos redimensionar el objeto para que quede proporcionado en relación con la escena del juego. Además, les pedimos que le pongan un nombre claro, como por ejemplo, *proyectil*.



Elección de la pelota (*Ball*) en la biblioteca de objetos

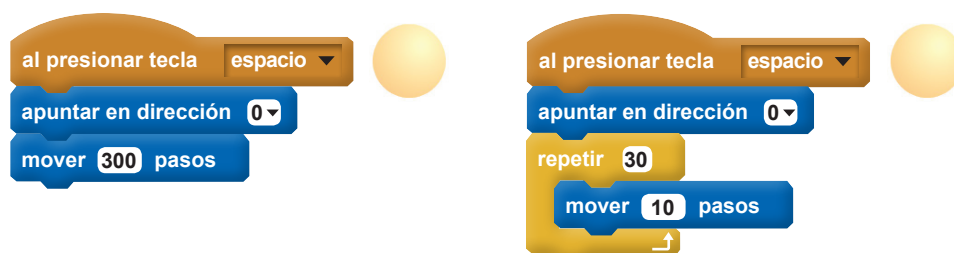
Al igual que hicimos con el resto de los objetos, empezamos identificando un estado inicial adecuado para el proyectil. En este caso, es oportuno que se ubique junto a la nave verde. El bloque `ir a []` de la categoría *Movimiento* sirve para este propósito. Permite que el objeto que está siendo programado se desplace hasta la posición de otro objeto de la escena. Nosotros haremos que el proyectil se posicione junto a la nave principal. Además, nos ocupamos de que esté visible.



Inicialización del proyectil

Para lograr que el proyectil se mueva, primero hay que identificar el evento que desencadena la acción. En este caso, *presionar la tecla espacio*. Proponemos a los estudiantes que programen el proyectil para que se mueva hacia arriba como el disparo del Space Invaders, resolviendo la primera consigna de la ficha. Luego,

pedimos que compartan las soluciones. Probablemente sean similares a alguna de las siguientes:



Posibles propuestas de los estudiantes para disparar el proyectil

Las dos propuestas resuelven el desafío. La instrucción `apuntar en dirección (0)` orienta el proyectil hacia arriba. Además, 300 es una cantidad de pasos que alcanza para que la pelota llegue hasta la parte superior de la escena. Sin embargo, al utilizar el bloque `mover [] pasos` con números grandes como argumento, el proyectil se mueve de modo muy brusco y se pierde la ilusión de un movimiento continuo. Por eso es mejor utilizar el ciclo `repetir ()`. En este caso, a pesar de que se recorra la misma distancia, la velocidad del movimiento será totalmente distinta. Al hacerlo de este modo, la pelota se desplaza 30 veces 10 pasos, mientras que con la primera opción se mueve 300 pasos de una vez.

Luego, indicamos a los estudiantes que jueguen al Space Invaders para comprender cómo funcionan los disparos; también les pedimos que resuelvan la segunda consigna de la ficha. El objetivo es que reconozcan que, una vez que el proyectil ha llegado al límite superior de la pantalla o ha colisionado con alguna nave enemiga, la nave verde está en condiciones de realizar un nuevo disparo.

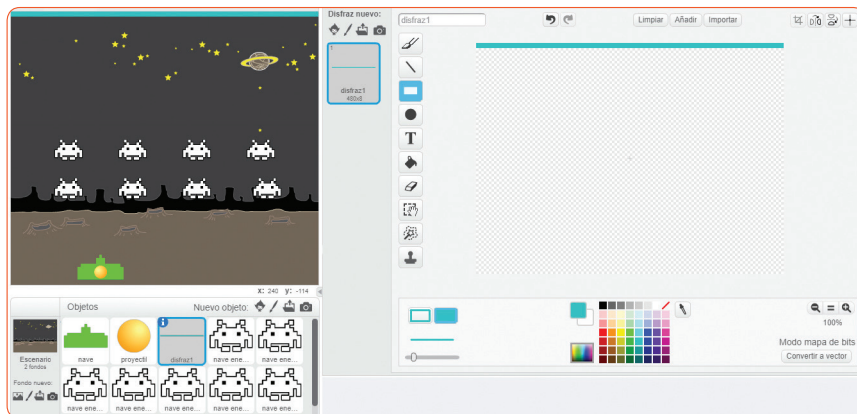
Proponemos el siguiente desafío a los estudiantes con el propósito de invitarlos a resolver la tercera consigna de la ficha: “¿Se animan a lograr que la pelota se mueva hasta llegar a la parte superior de la pantalla?”.

Para saber si la pelota ha llegado al extremo superior sin tener que calcular exactamente la cantidad de pasos necesarios, podemos utilizar el ciclo `repetir hasta que < >`. A diferencia del ciclo `por siempre`, que se ejecuta indefinidamente, y de `repetir ()`, que lo hace una cantidad fija de veces, el bloque `repetir hasta que < >` repite instrucciones hasta que una condición específica se cumpla.



Bloque para repetir instrucciones hasta que se cumpla una condición

Preguntamos a los estudiantes: “¿Qué podemos hacer para saber si llegamos a la parte superior de la pantalla?”. Una manera de lograrlo es agregar un objeto en el borde superior que ocupe todo el ancho de la escena y que sea de un color distinto al del fondo. Se recomienda utilizar la herramienta de dibujo que brinda Scratch y trazar un simple rectángulo o una simple raya. Para que resulte claro, es conveniente ponerle un nombre a este nuevo objeto para poder identificarlo. Por ejemplo, *límite superior*.



Creación de un objeto para cubrir el borde superior de la pantalla

Ahora hace falta encontrar alguna instrucción que nos permita saber si el proyectil está en contacto con este nuevo objeto. Dentro de la categoría *Sensores* se encuentran dos bloques que sirven para este propósito: `¿tocando []?` y `¿tocando el color []?`. Estos bloques nos permiten saber si un objeto está en contacto con algún otro objeto o color específico. Se puede utilizar cualquiera de los dos como condición para el ciclo `repetir hasta que < >`, para conseguir que la pelota no se siga moviendo una vez que toca el borde superior.



Posible solución para que el proyectil se mueva hasta alcanzar el límite superior de la pantalla

Planteamos el siguiente desafío: “Ya logramos identificar cuándo nuestro proyectil colisiona con el borde superior. ¿Se animan a extender el programa para que también detecte el choque con las naves enemigas?”. Proponemos a los estudiantes resolver la consigna 4 de la ficha.

A la derecha se observa un posible programa que podrían proponer los estudiantes, que no funciona como se espera. Hay que recordarles que en Scratch los bloques se ejecutan de manera secuencial, uno detrás del otro. Por lo tanto, hasta que el proyectil no choque con el límite superior, no se ejecutarán los demás ciclos. Se precisa una forma de verificar todas las condiciones en el mismo momento. En la categoría *Operadores* se encuentran dos bloques que permiten generar condiciones compuestas: `< > y < >` y `< > o < >`. En el caso del bloque `< > y < >`, la condición compuesta es verdadera solo si ambas



Programa incorrecto para resolver la consigna 4 de la ficha

subcondiciones lo son. Si una de ellas es falsa, la condición compuesta también. En cambio, para que una condición compuesta construida con el bloque `< > o < >` sea verdadera, alcanza con que solo una de las dos subcondiciones lo sea.



Bloques para condiciones compuestas

Preguntamos a los estudiantes: “¿Cuál de estas dos formas de combinar condiciones les parece que vamos a necesitar para resolver nuestro problema?”. La que nos provee el bloque `< > o < >`. Además, les explicamos que podemos poner bloques `< > o < >` dentro de otros bloques `< > o < >` para combinar así más de dos condiciones, como se necesita en este caso.



Posible solución para lograr que el proyectil avance, considerando colisiones contra el límite superior y dos naves enemigas

Algo que podemos observar es que, si en el juego hay muchas naves enemigas en lugar de dos, la condición será muy extensa. Una buena alternativa es utilizar el bloque `¿tocando el color []?`. De esta manera, usando el color de las naves enemigas, se puede reducir la condición para reconocer colisiones.¹



Solución de la consigna 4 usando los bloques `¿tocando []?` y `¿tocando el color []?`

CIERRE

Comentamos con los estudiantes que con los sensores se consiguió que el programa detecte el choque entre distintos objetos. Además, por primera vez se utilizó la estructura de control `repetir hasta que < >`. Por último, les hacemos notar que usamos condiciones compuestas para que el programa reconozca cuándo el proyectil debe detenerse.

¹ Esta solución será correcta solo si en la escena no hay otros objetos del mismo color que las naves enemigas.

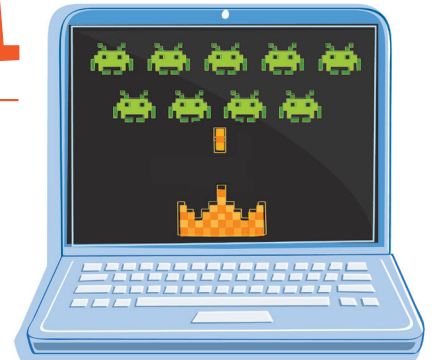
NOMBRE Y APELLIDO:

CURSO:

FECHA:

DISPARAMOS EL PROYECTIL

Ya programaste el movimiento de la nave principal y el de las naves enemigas. Ahora te falta crear y programar el proyectil. En esta actividad vas a lograr que el proyectil se mueva hasta chocar con el límite superior de la pantalla o con alguna de las naves enemigas.



1. Programá el objeto *Ball* para que se mueva como un proyectil. ¿Cómo lo hiciste?

¡ATENCIÓN!

Asegurate de que al comienzo el proyectil esté bien posicionado y visible.

2. Jugá al Space Invaders para comprender el funcionamiento de los proyectiles. ¿Cómo funcionan los disparos? ¿Es posible hacer un disparo tras otro?

3. Lográ que el proyectil se mueva hasta alcanzar el extremo superior de la pantalla y luego se detenga. ¿Qué bloques usaste?

4. Elegí una condición para que el programa también detecte la colisión del proyectil con las naves enemigas. ¿Cómo resolviste el desafío?



Secuencia Didáctica 2

PROCEDIMIENTOS Y PARÁMETROS

Un **procedimiento** es una porción de un programa más grande que encapsula una tarea específica. Los procedimientos suelen usarse para descomponer problemas complejos en piezas más simples. Además, son útiles para evitar la repetición de secuencias de instrucciones idénticas en los programas. Bien utilizados, mejoran notablemente la legibilidad de los programas.

Sin explicitarlo, ya los hemos usado: muchos de los bloques que utilizamos en la secuencia anterior son procedimientos. Bloques como `mover () pasos` o `apuntar en dirección ()` son ejemplos de algunos procedimientos que provee Scratch. Pero todavía no hemos creado ninguno. Para entender cómo hacerlo nos alejaremos de las computadoras y crearemos procedimientos para dibujar y escribir canciones. Además, se presentará la creación de procedimientos con **parámetros** para modificar sus comportamientos al utilizarlos con distintos argumentos.

.....

OBJETIVOS

- Introducir las nociones de procedimiento y parámetro.
- Poner en práctica la división de tareas como estrategia para resolver problemas.

.....

Actividad 1

Procedimientos para dibujar



INDIVIDUAL

OBJETIVOS

- Poner en práctica la división de tareas como técnica de programación.
- Crear procedimientos para resolver subproblemas.
- Usar procedimientos para no repetir secuencias de instrucciones iguales.

MATERIALES



Papel cuadriculado



Lápiz



Ficha para estudiantes

DESARROLLO

En esta actividad introducimos la noción de **procedimiento** y destacamos uno de sus usos más importantes: sirve para dividir un problema en varios subproblemas más sencillos; cada uno se resuelve con un procedimiento específico.

Comenzamos la clase recordando el lenguaje de programación con flechas utilizado en la actividad “¡A dibujar!” del capítulo 2, “Algoritmos y programas”. Pedimos a los estudiantes que resuelvan las consignas 1 y 2 de la ficha. Tienen que escribir un programa para dibujar la imagen que se muestra en la siguiente figura, en la que también se indica una posible solución. Una vez completada la tarea, invitamos a los estudiantes a que escriban sus programas en el pizarrón y los analizamos entre todos para ver si son correctos.



→ ↘ ↓ ↙ ← ↘ ↑ ↘ → → → → → ↘ ↓ ↙ ← ↘ ↑ ↘

Cuadrícula a programar y posible solución

Preguntamos a los estudiantes: “Observando con atención la figura, ¿notan partes que se repiten? ¿Cuántas veces se repiten?”. Se repite 2 veces el cuadrado negro de 2×2 . Proponemos a los estudiantes que resuelvan la consigna 3 de la ficha, donde tienen que definir una secuencia de instrucciones que dibuje únicamente un cuadrado negro de 2×2 . Una posible solución se muestra a continuación.

→ ↘ ↓ ↙ ← ↘ ↑ ↘

Sin importar cuál de los cuadrados negros se dibuje, las instrucciones para hacer ambos cuadrados son las mismas. En computación, para evitar repetir partes de un programa, podemos usar procedimientos. Los procedimientos se definen con un nombre y una secuencia de instrucciones. Es importante que el nombre de un procedimiento describa apropiadamente la función que cumple. En este caso, un buen nombre puede ser “Dibujar cuadrado”, como se muestra a continuación.

Dibujar cuadrado → ↘ ↓ ↙ ← ↘ ↑ ↘

Procedimiento para dibujar un cuadrado de 2×2

Una vez definidos, se puede recurrir a los procedimientos desde cualquier parte del programa. Para usarlos, se realiza una **llamada** (o invocación) usando el nombre del procedimiento. Cada vez que se llama a un procedimiento, se ejecutan las instrucciones que le asignamos.

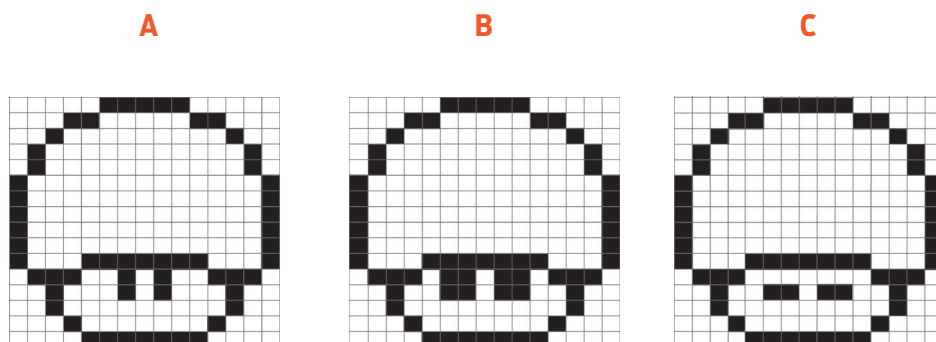
Indicamos a los estudiantes que escriban un programa para el mismo dibujo, pero esta vez usando el procedimiento “Dibujar cuadrado”, lo que corresponde a la consigna 4 de la ficha. A continuación presentamos una solución posible.

Dibujar cuadrado → → → → Dibujar cuadrado

Posible solución de la consigna 4 de la ficha

El procedimiento “Dibujar cuadrado” se llama 2 veces: una vez al empezar y otra antes de terminar. Entre estas dos llamadas se mueve el lápiz cuatro veces a la derecha. Como puede observarse, al usar procedimientos se consiguen programas más cortos y legibles. Además, en este caso nos permitió no escribir dos veces la secuencia de instrucciones para dibujar un cuadrado negro.

Luego de finalizada la actividad de los cuadrados, proponemos a los estudiantes que resuelvan la consigna 5 de la ficha. Tendrán que crear un programa que dibuje al personaje Toad (A). El objetivo inicial es que resuelvan la actividad sin usar procedimientos, para luego resaltar la importancia de utilizarlos para modularizar¹ programas.



Diferentes versiones del personaje Toad del juego Mario Bros

El desarrollo del programa les llevará un buen tiempo. Luego pedimos a los estudiantes que compartan sus programas. Reflexionamos con ellos: “¿Se imaginan si tuviéramos que crear un nuevo programa para cada uno de los Toads que se encuentran en la imagen de la ficha? ¿En qué se diferencian los dibujos?”. Son casi iguales: solo cambia la forma de los ojos. Les explicamos que esto quiere decir que, a la hora de crear los programas para cada uno de los Toads, tienen que repetir varias partes del programa que ya crearon. Les preguntamos: “¿Cómo podemos dividir en partes el problema principal?”. Les decimos que una posibilidad es dividirlo en tres: “Dibujar sombrero”, “Dibujar cara” y “Dibujar ojos”. Proponemos a los estudiantes que realicen la consigna 6 de la ficha, en la que se pide crear un procedimiento para cada una de las partes.

.....

¹ En informática, se usa este verbo para referirse a la operación de subdividir una aplicación en partes más pequeñas (módulos), cada una de las cuales es lo más independiente posible de la aplicación en sí y de las partes restantes.

Suponiendo que ya se hayan definido los procedimientos, les pedimos a los estudiantes que resuelvan la consigna 7 de la ficha, donde tienen que crear el programa que dibuje la imagen (A) utilizando los procedimientos creados para cada una de las partes. Un posible programa para dibujar al Toad clásico usando procedimientos es el siguiente:

Posicionar sombrero

Dibujar sombrero

Posicionar cara

Dibujar cara

Posicionar ojos

Dibujar ojos

Posible programa para dibujar a Toad usando procedimientos

Los procedimientos “Posicionar sombrero”, “Posicionar cara” y “Posicionar ojos” consisten en la secuencia de instrucciones que ubican el lápiz en el lugar adecuado para poder dibujar cada parte de Toad. Esta posición depende de cómo hayan definido los estudiantes cada uno de los procedimientos para dibujar. No es obligatorio que definan los procedimientos de posicionamiento; es aceptable que las instrucciones necesarias para posicionar el lápiz aparezcan “sueltas” en el programa que dibuja a Toad.

Como se señaló antes, si se compara el Toad de la figura (A) con los de las figuras (B) y (C), se puede notar que solo difieren en la forma de los ojos. Por lo tanto, los procedimientos “Posicionar sombrero”, “Dibujar sombrero”, “Posicionar cara” y “Dibujar cara” ya creados se pueden reutilizar para las otras variantes del dibujo, que es lo que se pide a los estudiantes en la consigna 8 de la ficha.

RÚBRICAS

Al final de la ficha de actividades para los estudiantes, incluimos una rúbrica. El objetivo es que los alumnos sepan qué estaremos evaluando y qué criterios utilizaremos.

CIERRE

Repasamos con los estudiantes que en esta actividad se crearon procedimientos para diferentes cosas. En primer lugar, se consiguió no escribir de forma repetida secuencias de instrucciones idénticas. Escribimos un único procedimiento para dibujar un cuadrado negro, al que luego llamamos dos veces desde el programa principal. Por otro lado, usamos los procedimientos para dividir un problema en partes más pequeñas. De este modo, conseguimos programas más modularizados y de fácil lectura. Por último, también se vio que un procedimiento definido en un cierto contexto luego puede ser usado en otro; en este caso, para realizar distintas versiones del personaje Toad.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

PROCEDIMIENTOS PARA DIBUJAR

Volvamos al pasado, pero esta vez para dibujar a un personaje de uno de los videojuegos más famosos de toda la historia: el Mario Bros. ¡Dibujemos a Toad usando el lenguaje de flechas!



1. ¿Recordás el lenguaje de flechas que usaste en la actividad “¡A dibujar!” del capítulo “Algoritmos y programas”? ¿Con cuántas instrucciones contabas en el lenguaje basado en flechas? ¿Cuáles eran esas instrucciones?

2. Escribí un programa en el lenguaje de flechas que dibuje la siguiente cuadrícula.



3. Escribí una secuencia de instrucciones con el lenguaje de flechas que solo dibuje el cuadrado negro de 2×2 de la imagen. Esta secuencia de instrucciones a la que tenés que ponerle un nombre es un **procedimiento**.

NOMBRE Y APELLIDO:

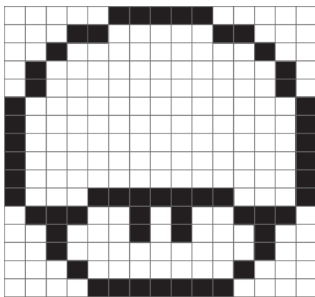
CURSO:

FECHA:

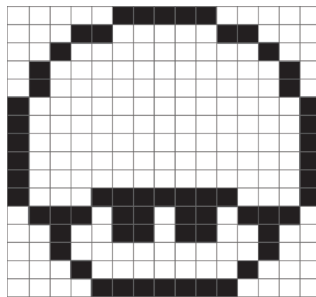
4. Escribí un nuevo programa que dibuje toda la cuadrícula, pero esta vez utilizando el procedimiento que acabás de crear.

5. Utilizando el lenguaje de flechas, creá un programa para dibujar la versión A del personaje Toad.

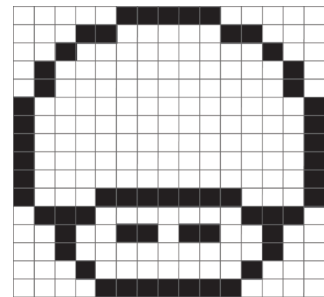
A



B



C



6. Teniendo en cuenta las tres partes que componen la imagen (ojos, sombrero y cara), creá un procedimiento para cada una.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

7. Escribí un nuevo programa que use los procedimientos creados para dibujar la versión (A) de Toad.

8. Pensá cómo podrías reutilizar estos procedimientos para dibujar las versiones (B) y (C) de Toad.
¿Qué hay que cambiar?

EVALUACIÓN

Estas son las tareas y la forma de calificar que tu docente considerará para evaluar cómo resolvés las actividades.

CALIFICACIÓN	ES NECESARIO TRABAJAR MÁS	BUENO-MUY BUENO	EXCELENTE
División del problema en partes	El estudiante no puede separar las partes de la figura.	El estudiante logra separar dos de las tres partes de la figura.	El estudiante logra identificar las tres partes de la figura.
Uso de procedimientos	El estudiante no comprende que es posible construir un procedimiento para dibujar cada parte y luego unirlos.	El estudiante identifica algunos procedimientos. Junta en un solo procedimiento partes que deberían estar separadas.	El estudiante comprende que es posible construir un procedimiento para dibujar cada parte y luego unirlos.

Actividad 2

Estribillos como procedimientos



TODO EL CURSO

OBJETIVOS

- Ejercitar la búsqueda de patrones.
- Reconocer el estribillo como una porción de una canción que se repite y asociarlo con la noción de procedimiento.

MATERIALES



Proyector (opcional)



Letra de la canción

DESARROLLO

En esta actividad se comparte una canción para trabajarla con los estudiantes. En el caso de contar con un proyector, facilitará la actividad proyectar la letra de la canción. Si no, es posible repartir fotocopias con la letra. Se recomienda utilizar canciones conocidas por los alumnos y que sean adecuadas en el ámbito escolar. A modo de ejemplo, se trabaja con *Sobre el puente de Avignon*, pero podría reemplazarse por cualquier canción con un estribillo que se repite.

SOBRE EL PUENTE DE AVIGNON

Sobre el puente de Avignon
todos bailan, todos bailan.
Sobre el puente de Avignon
todos bailan y yo también.

Hacen así...
así las lavanderas.
Hacen así...
así me gusta a mí.

Sobre el puente de Avignon
todos bailan, todos bailan.
Sobre el puente de Avignon
todos bailan y yo también.

Hacen así...
así las planchadoras.
Hacen así...
así me gusta a mí.

Sobre el puente de Avignon
todos bailan, todos bailan.
Sobre el puente de Avignon
todos bailan y yo también.

Hacen así...
así las costureras.
Hacen así...
así me gusta a mí.

Sobre el puente de Avignon
todos bailan, todos bailan.
Sobre el puente de Avignon
todos bailan y yo también.

Hacen así...
así los zapateros.
Hacen así...
así me gusta a mí.

Se puede cantar la canción en conjunto o simplemente darles unos minutos a los estudiantes para que la lean. Se les preguntará para que trabajen de manera grupal: “¿Hay alguna parte de la canción que se repita varias veces? ¿Cuántas veces se repite?”. Luego de unos minutos, los estudiantes comparten sus respuestas. En el caso de contar con un proyector, se puede sugerir que pasen e identifiquen qué partes se repiten. Se reflexiona con ellos: “Estas partes que se repiten, en el mundo de la música, se llaman **estribillos**. Seguramente se han encontrado alguna vez

con letras de canciones donde se escribe el estribillo una sola vez al principio de la canción y luego, en vez de repetirlo completamente, solo se escribe la palabra *estribillo* donde corresponde”.

SOBRE EL PUENTE DE AVIGNON

ESTRIBILLO:

Sobre el puente de Avignon
todos bailan, todos bailan.

Sobre el puente de Avignon
todos bailan y yo también.

Hacen así...
así las lavanderas.
Hacen así...
así me gusta a mí.

ESTRIBILLO

Hacen así...
así las planchadoras.
Hacen así...
así me gusta a mí.

ESTRIBILLO

Hacen así...
así las costureras.
Hacen así...
así me gusta a mí.

ESTRIBILLO

Hacen así...
así los zapateros.
Hacen así...
así me gusta a mí.

Se muestra a los estudiantes cómo quedaría la canción en ese caso, y se los invita a cantarla nuevamente utilizando la definición de estribillo. Una vez que haya finalizado la canción, se les pregunta: “¿Se modificó la canción que cantábamos o sigue siendo la misma?”. La canción sigue siendo la misma, pero ahora pueden diferenciarse varias partes: primero, la definición del estribillo y luego, la definición de la canción, donde el estribillo se menciona varias veces.

Preguntamos a los estudiantes: “Además de tener que escribir únicamente una vez la letra del estribillo, ¿qué otras ventajas notan en este procedimiento?”. Un beneficio adicional es que, al escribir el estribillo una sola vez, es más fácil corregir posibles errores. “Imaginemos que cometimos un error al escribir una parte de nuestro estribillo. Por ejemplo, escribimos *avignon* con minúscula. Tendríamos que corregir este error en todas las partes de la canción donde lo hubiésemos repetido. Utilizando un procedimiento, solo tenemos que modificarlo en un único lugar”.

CIERRE

Para concluir la actividad, reflexionamos con los estudiantes: “Decir que una parte de la canción es el estribillo nos permite usarlo la cantidad de veces que necesitemos escribiéndolo una sola vez. Con nuestros programas, podemos hacer algo parecido a lo que hicimos con la canción. En computación, al estribillo lo encapsulamos en un **procedimiento**. Hacer esto no solamente nos ahorra tener que escribir muchos programas con partes repetidas, sino que también nos permite tenerlos modularizados y bien organizados. Tengan siempre presente que los procedimientos se definen con un nombre, que preferentemente describe su propósito”.

Actividad 3

Programamos canciones en Scratch



DE A DOS

OBJETIVOS

- Introducir la noción de parámetro.
- Escribir un programa en el que se definen procedimientos con parámetros.

MATERIALES



Computadora



Scratch



Ficha para estudiantes

DESARROLLO

En esta actividad vamos a aprender a crear procedimientos con parámetros en Scratch. La actividad consiste en crear una animación en la que un personaje cante la canción *Un elefante se balanceaba*.

En las consignas 1 y 2 de la ficha se solicita que elijan el personaje que cantará y exploren la plataforma en búsqueda de un bloque para que el personaje “hable”. A modo de ejemplo, usaremos al gato Scratch. Además, utilizaremos el bloque `decir [] por () segundos` de la categoría *Apariencia* para que cante. Este bloque tiene dos parámetros: uno para la frase que dirá el personaje y otro para la cantidad de segundos que se mostrará en pantalla esta frase.

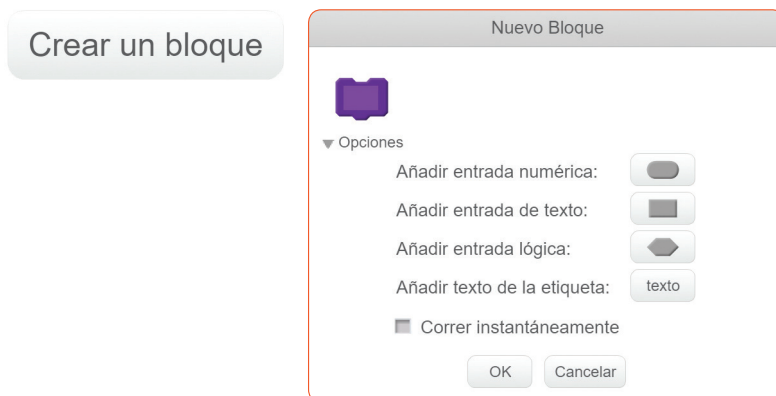
Invitamos a los estudiantes a resolver la consigna 3 de la ficha. Allí se les pide hacer que el personaje cante las primeras tres estrofas de la canción *Un elefante se balanceaba* usando el bloque `decir [] por () segundos`. Una vez que lo hayan hecho, se compartirán las soluciones propuestas entre toda la clase. Probablemente algún estudiante llegue a un programa similar al de la siguiente figura.



Programa que resuelve el desafío de la consigna 3 de la ficha

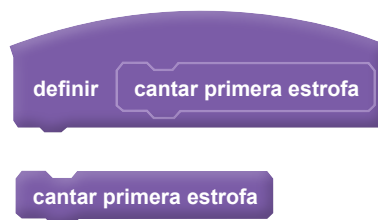
Reflexionamos con los estudiantes: “Para solo 3 estrofas usamos 13 bloques. Imaginemos que tenemos que escribir la canción hasta llegar a 100 elefantes. ¿Cómo lo harían? ¿Qué construcción, que estuviéramos trabajando en las últimas actividades, nos permite dividir un problema en partes?”. Se espera que hagan referencia a los procedimientos.

Scratch nos permite definir procedimientos, es decir, crear nuestros propios bloques. Hay que destacar que estos nuevos bloques solo pueden ser utilizados por el personaje seleccionado en la galería al momento de definirlos. En la categoría *Más bloques* se encuentra el botón *Crear un bloque*. Al hacer clic sobre él, se despliega el siguiente menú.



Menú para crear un procedimiento, una vez que se han desplegado las opciones

Al crear un procedimiento, primero, hay que ponerle un nombre que describa su propósito. Como se empieza con la primera estrofa, un nombre adecuado puede ser `cantar primera estrofa`. Para poner nombre al nuevo procedimiento, se hace clic en el cuadro de texto violeta. La primera estrofa se crea sin ningún parámetro, por lo que directamente se puede hacer clic en *OK*. Automáticamente se generará el bloque `definir [cantar primera estrofa]` en el sector de edición del programa. Además, dentro de la categoría *Más bloques*, se contará con un nuevo bloque `cantar primera estrofa` para ser utilizado.



Procedimiento `cantar primera estrofa`

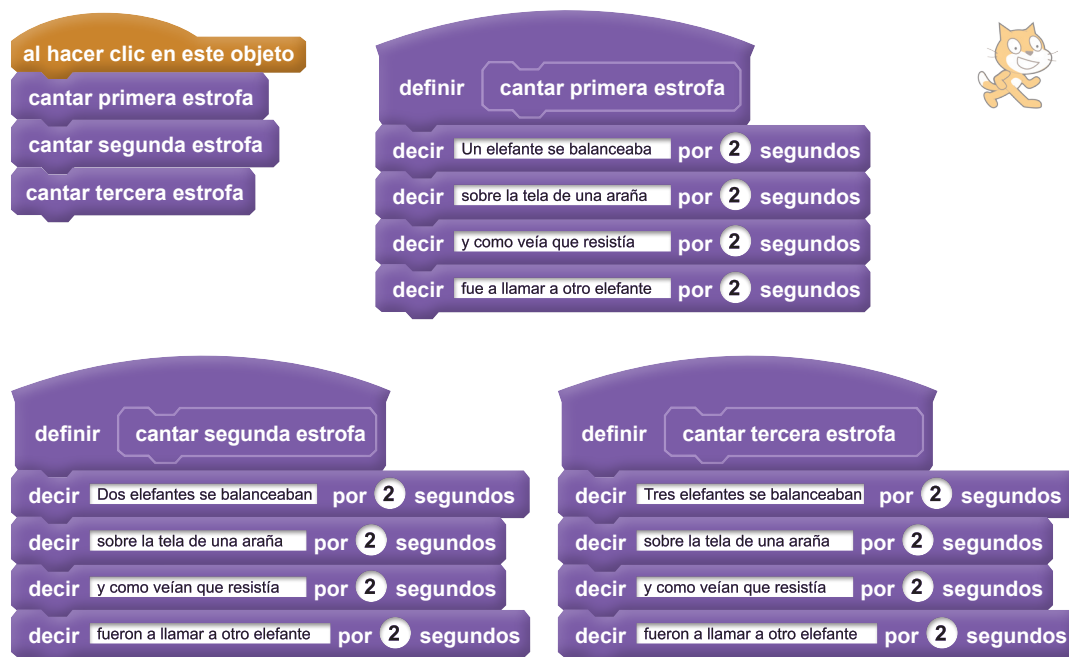
Preguntamos a los estudiantes: “¿Si creamos el siguiente programa, creen que el personaje reproducirá la primera estrofa?”. La respuesta es no. La definición del procedimiento `cantar primera estrofa` únicamente tiene un nombre, pero no las instrucciones que se ejecutarán cada vez que lo invoquemos. Tenemos que programarlo para que haga lo que nosotros queremos, como se propone en la consigna 4 de la ficha.

al hacer clic en este objeto
cantar primera estrofa



Invocación al procedimiento creado

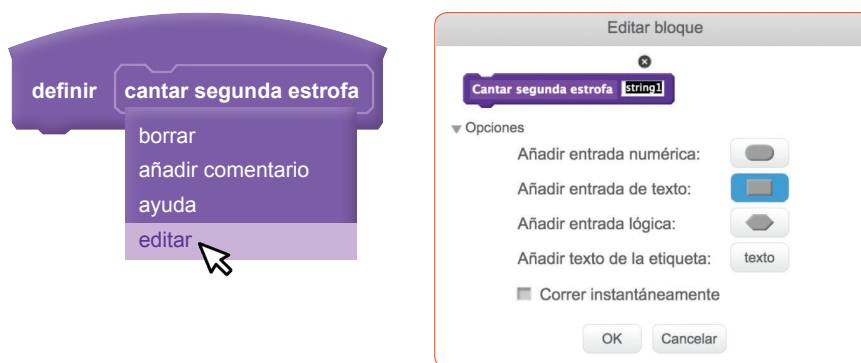
Les proponemos a los estudiantes que intenten que el personaje cante, además de la primera estrofa, la segunda y la tercera, siguiendo la consigna 5 de la ficha. En caso de contar con proyector, se recomienda que los estudiantes que resuelvan el desafío muestren sus programas. Es probable que se presente alguna solución similar a las siguientes.



Posibles soluciones de la consigna 5

Luego compartimos con los estudiantes la siguiente reflexión: “Las soluciones son válidas, ya que resuelven el desafío. Pero si nos ponemos a revisar los procedimientos `cantar segunda estrofa` y `cantar tercera estrofa`, notamos que son casi iguales: solo cambia la cantidad de elefantes. Pensemos que queremos modularizar nuestros programas y utilizar menor cantidad de bloques. En este caso, lo único que debemos modificar es el valor de cantidad de elefantes. ¿Cómo podemos hacer para que el comportamiento de un procedimiento sirva para ambas estrofas? La respuesta es: usando un **parámetro**”.

En Scratch se pueden modificar los procedimientos creados. Al hacer clic derecho sobre la definición del bloque `cantar segunda estrofa`, seleccionamos la opción *editar*. Al desplegar el menú *Opciones*, se desplegarán las posibilidades para agregar parámetros a los procedimientos. En Scratch los parámetros se denominan *entradas*. Existen tres tipos de entradas en Scratch: numérica, texto y lógica. En este caso, la opción correcta es *entrada de texto*. Al hacer clic sobre *Añadir entrada de texto*, automáticamente se actualiza nuestro bloque como `cantar segunda estrofa []`. Ahora en el bloque aparecerá una casilla con el texto *string1*¹, que es el nombre que asigna Scratch por defecto a los parámetros del tipo texto. Así como cuando usamos el bloque `decir [] por ()` segundos debemos incluir el texto que dirá el personaje y la cantidad de segundos que se mostrará, al usar el bloque que estamos definiendo se debe indicar la cantidad de elefantes de una estrofa particular.



Edición de un procedimiento para agregar un parámetro

Es recomendable nombrar los parámetros con un término relacionado con aquello que representan. Teniendo en cuenta que en este caso el parámetro representa una cantidad de elefantes, *cantidad* puede ser un buen nombre. Basta con reemplazar el texto *string1* por *cantidad* para hacerlo.

Es importante pensar en un nuevo nombre para el procedimiento, ya que, al incorporarle el parámetro *cantidad*, sirve no solo para resolver la segunda estrofa, sino también la tercera, la cuarta o la número cien si hiciera falta. Por lo tanto, un nombre más adecuado para el procedimiento es *cantar estrofa*. Para cambiar el nombre, hay que hacer nuevamente clic derecho sobre la definición del procedimiento y seleccionar la opción *Editar*. Una vez modificado el nombre, el bloque aparece como `cantar estrofa []` dentro de la categoría *Más bloques*.



Procedimiento con un parámetro

¹ *String* es el nombre de un tipo de dato para representar texto (o cadenas de caracteres).

Pedimos a los estudiantes que programen el cuerpo del procedimiento, como se pide en la consigna 6 de la ficha. Es probable que los estudiantes, al no saber cómo manipular bloques de texto, presenten un procedimiento similar al siguiente.



Posible solución por parte de los estudiantes

El procedimiento así definido no resuelve el desafío. Primero el personaje dice la cantidad de elefantes y dos segundos después, la frase “elefantes se balanceaban”. En Scratch hay un bloque que nos permite unir dos textos. En la categoría *Operadores* se encuentra `unir [] []`. Este bloque toma dos textos como argumentos y los enlaza uno detrás del otro. Por lo tanto, se puede unir el parámetro *cantidad* con el texto *elefantes se balanceaban* para formar la frase completa. En la siguiente figura se puede observar la solución final del desafío.



Solución de la actividad

CIERRE

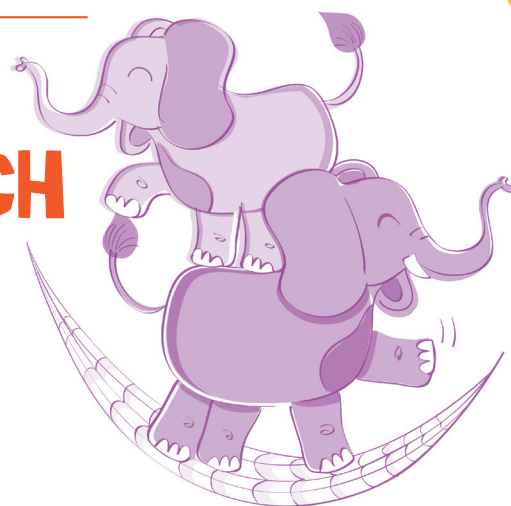
Crear procedimientos permite dividir un problema en varios problemas más pequeños. Por otro lado, se los puede llamar tantas veces como haga falta, sin necesidad de escribir varias veces fragmentos del programa que comprenden las mismas instrucciones. Por último, al usar parámetros, se consiguen soluciones más generales. En este caso, el comportamiento de los procedimientos se define una vez que se los llama y se les asigna valores concretos como argumentos. En el ejercicio, la *cantidad* de elefantes es el parámetro del procedimiento `cantar estrofa []` y *dos* y *tres* son los argumentos.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

PROGRAMAMOS CANCIONES EN SCRATCH



En esta actividad vas a programar un personaje que cantará la canción *Un elefante se balanceaba*. Para lograrlo, vas a tener que definir procedimientos en Scratch.

1. Elegí un personaje para que cante la canción. ¿Cuál elegiste?

2. Explorá el entorno de Scratch y buscá un bloque que le permita al personaje decir frases como en una historieta. ¿Qué bloque es?

3. Utilizando el bloque `decir [] por () segundos` hacé un programa para que el personaje elegido cante las primeras tres estrofas de la canción *Un elefante se balanceaba*. Tiene que cantar de a un verso por vez, cada vez que se haga clic sobre él. Escribí acá el programa que armaste.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

4. Programá el procedimiento `cantar primera estrofa` para que el personaje la reproduzca cada vez que hagas clic sobre él. ¿Cómo es el procedimiento que armaste?

5. Elaborá procedimientos similares para que el personaje cante la segunda y la tercera estrofa. ¿Cómo lo hiciste?

6. Programá el procedimiento `cantar estrofa [cantidad]` y al personaje principal para que cante las tres primeras estrofas. ¿Cómo quedó finalmente el programa?





Secuencia Didáctica 3

INVASORES DEL ESPACIO, LA CONTINUACIÓN

En esta secuencia didáctica se retoma el desarrollo del proyecto Invasores del espacio aplicando técnicas trabajadas en la secuencia anterior. Se recuperan las definiciones de **procedimientos** y **parámetros** para modularizar el videojuego. Además, se presentan situaciones de interacción entre el usuario y el programa, que los estudiantes deben tener en cuenta para que el videojuego funcione correctamente. Esto los conducirá a procesos de **testeo**. Es decir que deben realizar pruebas que permitan verificar si el programa se comporta de la manera esperada. Por último, se incorpora puntaje en el juego, para lo cual hace falta usar **variables**.

.....

OBJETIVOS

- Integrar las definiciones de procedimientos y parámetros a un proyecto de programación.
- Presentar el concepto de variable y distinguir su definición, su modificación y su lectura.

.....

Actividad 1




Procedimientos en Space Invaders

 DE A DOS

OBJETIVOS

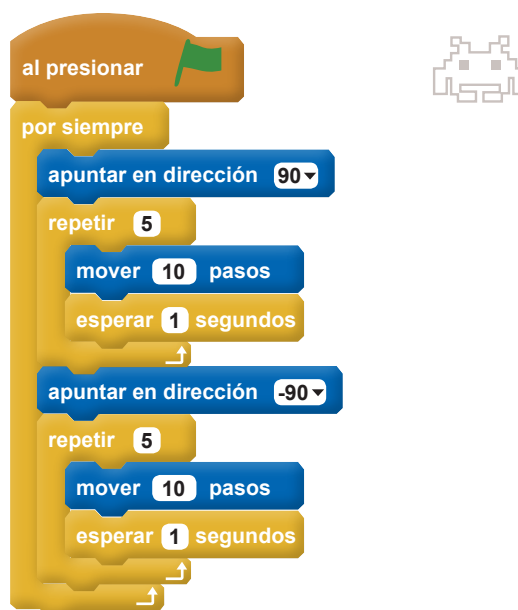
- Usar procedimientos para descomponer un problema en partes más sencillas.
- Definir procedimientos con parámetros.
- Refactorizar programas para obtener soluciones claras y compactas.

MATERIALES

-  Computadora
-  Scratch
-  Ficha para estudiantes

DESARROLLO

Comenzamos recuperando el programa que los estudiantes crearon para la actividad “Programamos las naves enemigas”, que es el que se muestra en la siguiente figura.

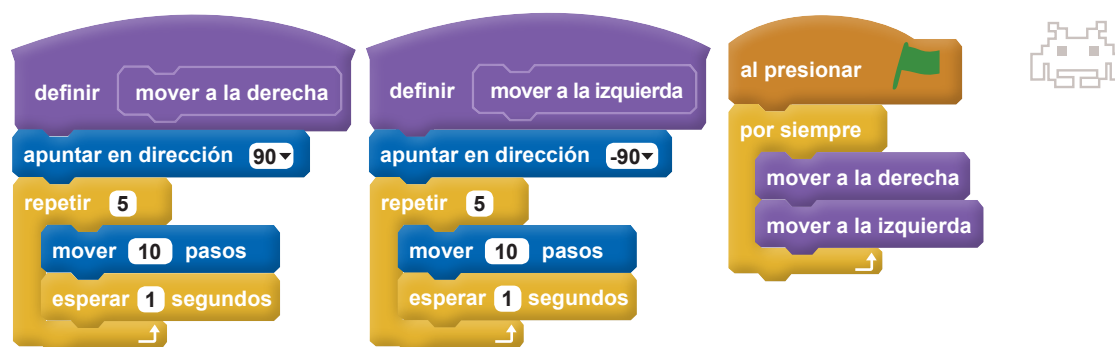


Programa creado para el movimiento de las naves enemigas

Preguntamos a los estudiantes: “¿Cuáles son las acciones principales que llevan a cabo las naves enemigas?”. Se espera que los estudiantes respondan “moverse a la derecha” y “moverse a la izquierda”.

Comentamos a continuación: “Nuestro programa comenzará a complejizarse. Para mantenerlo ordenado, es conveniente identificar las distintas partes que lo componen y resolver cada una en un procedimiento, es decir, descomponer nuestro problema en pequeñas unidades que resuelvan problemas chiquitos. Después, con cada parte resuelta, será sencillo combinarlas para completar nuestro programa. De esta forma, obtendrán un programa modularizado, ordenado y que se puede entender y modificar con facilidad”.

Proponemos a los estudiantes que dividan el problema en dos partes: desplazar naves a la derecha y desplazar naves a la izquierda. Los estudiantes deben resolver la consigna 1 de la ficha de la actividad. Para ello, tienen que crear dos procedimientos que permitan resolver los problemas de mover las naves enemigas a la derecha y a la izquierda. En la siguiente figura se observa una posible solución.



Programa modularizado

Solicitamos a los estudiantes que resuelvan la consigna 2 de la ficha, donde tienen que lograr que la nave lleve a cabo la siguiente secuencia: desplazarse 5 veces a la derecha, luego 5 veces a la izquierda, luego 13 veces a la derecha y finalmente 7 veces a la izquierda. El objetivo es recuperar el concepto de parametrización de procedimientos.

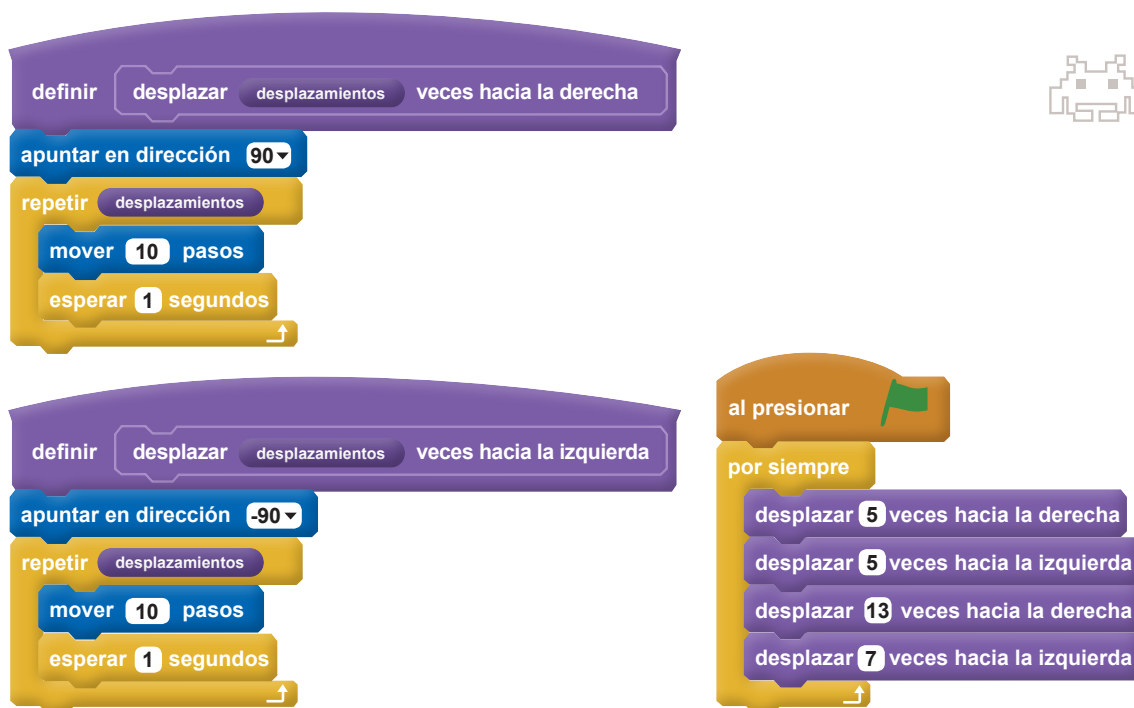
Es probable que algunos estudiantes propongan crear un procedimiento para cada cantidad de pasos, como se muestra a continuación en la columna de la izquierda. Otros pueden proponer que no se usen procedimientos, como se observa en la columna de la derecha.



Dos posibles propuestas de los estudiantes para resolver la consigna 2 de la ficha

Ambas aproximaciones son funcionalmente correctas, aunque desde el punto de vista de la programación están mal resueltas. En cada una se pueden encontrar secuencias de bloques muy parecidas a otras de la misma propuesta. Los procedimientos que muestra la columna de la izquierda solo difieren en la cantidad de repeticiones y la dirección del movimiento. En el programa de la columna de la derecha sucede algo parecido, aunque en este caso todas las instrucciones están en el cuerpo principal del programa.

Vale la pena destacar que cada vez que las naves enemigas se mueven en una cierta dirección, lo único que cambia es la cantidad de veces que se desplazan de a diez pasos. Esto da la pauta de que la cantidad de desplazamientos puede ser un parámetro numérico de los procedimientos que se ocupan de desplazar naves enemigas. Luego, desde el cuerpo principal del programa, se llama a los procedimientos usando los argumentos adecuados. Se les pide a los estudiantes que resuelvan la consigna 3 de la ficha. Una solución se muestra en la siguiente figura, donde el nombre elegido para el parámetro es *desplazamientos*.

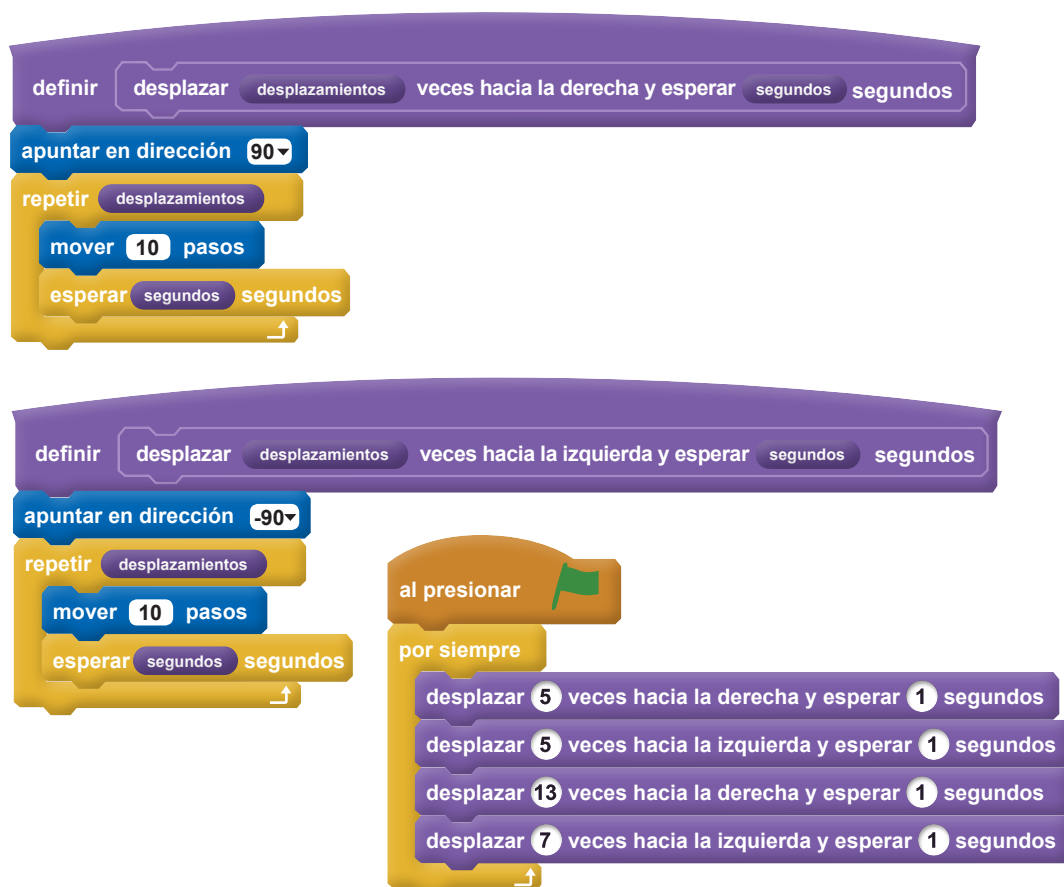


Programa con movimiento de naves contrarias parametrizado

Como puede observarse en la definición de `desplazar (desplazamientos) veces hacia la derecha`, se puede hacer que los parámetros de un procedimiento no aparezcan necesariamente al final. Para ubicarlos en otro lugar, al ir definiendo (o editando) el nombre de un procedimiento hay que ir intercalando el texto del nombre con el agregado de parámetros. En el ejemplo, primero habría que escribir "desplazar", luego hacer clic en *Añadir entrada numérica* (a la que llamamos *desplazamientos*) y por último en *Añadir texto a la etiqueta*, para completar el nombre con "veces hacia la derecha".

Es posible generalizar aún más los procedimientos. En Space Invaders, a medida que se avanza de nivel, la velocidad de las naves enemigas aumenta. Una forma de cambiar la velocidad de desplazamiento consiste en esperar menor cantidad de tiempo entre las sucesivas ejecuciones de los bloques `mover (10) pasos`. Teniendo esto en cuenta, proponemos a los estudiantes que resuelvan la consigna 4 de la ficha, para lo cual tienen que incorporar el tiempo de espera entre movimientos como un nuevo parámetro de los procedimientos. Nuevamente se deben modificar los procedimientos para agregar el nuevo parámetro. Un nombre apropiado es *segundos*, dado que representa el tiempo de espera que habrá entre los desplazamientos de las naves enemigas.

Una vez que hayamos modificado los procedimientos, en la categoría *Más bloques* aparecerán los bloques `desplazar (1) veces hacia la derecha y esperar (1) segundos` y `desplazar (1) veces hacia la izquierda y esperar (1) segundos`¹. En la siguiente figura se observa cómo quedan definidos el programa y los procedimientos utilizando los dos parámetros requeridos.



Procedimientos de movimiento con los parámetros *desplazamientos* y *segundos*

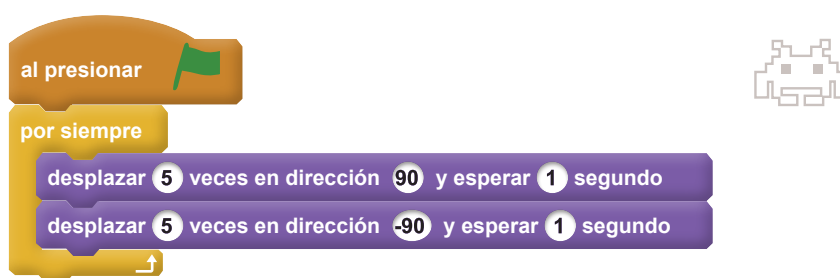
¹ En Scratch, el valor por defecto para los parámetros numéricos es 1.

Como se puede observar, ambos procedimientos son muy parecidos: difieren únicamente en la dirección del movimiento. Por lo tanto, si incorporamos un nuevo parámetro que represente la dirección, se obtiene una solución con un único procedimiento, que se muestra en la siguiente figura. De este modo, resolvemos la consigna 5 de la ficha.



Posible solución de la consigna 5

Finalmente, les pedimos que resuelvan la consigna 6 de la ficha. Allí se pide que las naves se muevan igual que en el Space Invaders: intercaladamente, irán 5 veces a la derecha y 5 veces a la izquierda.



Las naves se desplazan como en Space Invaders

CIERRE

Reflexionamos con los estudiantes: “Muchos de los bloques que hemos utilizado son procedimientos. Un **procedimiento** es un conjunto de instrucciones que resuelve un problema específico y podemos utilizarlo todas las veces que lo necesitemos. Los procedimientos tienen un nombre que describe su propósito, y pueden tener parámetros que modifiquen su comportamiento. Si prestamos atención, podemos observar que muchos de los bloques que provee Scratch son procedimientos. Por ejemplo, `mover () pasos`, `apuntar en dirección ()` y `decir [] por () segundos`. Cada uno de ellos tiene un propósito específico y un nombre descriptivo. En este caso, algún programador ya definió el conjunto de instrucciones que contienen, y nosotros podemos usarlos directamente”.

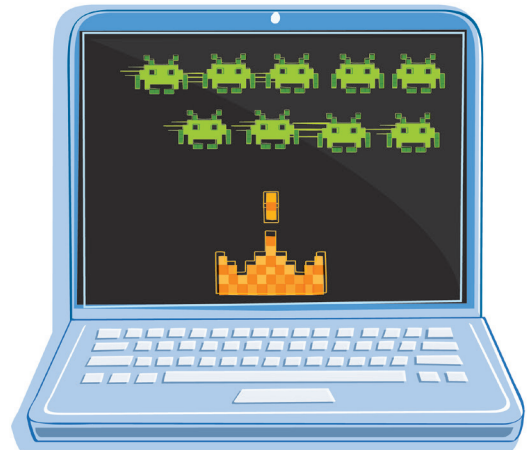
NOMBRE Y APELLIDO:

CURSO:

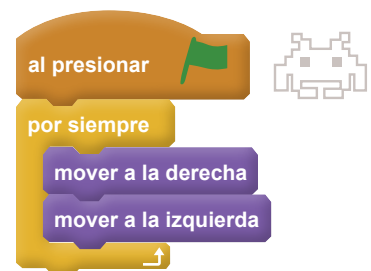
FECHA:

PROCEDIMIENTOS EN SPACE INVADERS

En las últimas actividades aprendiste a definir tus propios procedimientos. Ahora vamos a crear algunos para lograr que las naves enemigas se desplacen por la pantalla.



1. Creá dos procedimientos en Scratch para que las naves enemigas se muevan a la derecha y a la izquierda. Debajo de **al presionar bandera verde**, dentro del bloque **por siempre** tiene que haber solo dos bloques que se llamen **mover a la derecha** y **mover a la izquierda**. Una vez completada la tarea, observá y describí las similitudes y diferencias entre ambos procedimientos.



2. Ahora tenés que conseguir que las naves enemigas se muevan 5 veces a la derecha, luego 5 veces a la izquierda, luego 13 veces a la derecha y finalmente 7 veces a la izquierda. ¿Cómo lo hiciste?

NOMBRE Y APELLIDO:

CURSO:

FECHA:

3. Modificá los procedimientos `mover a la izquierda` y `mover a la derecha` para que tomen un parámetro que indique la cantidad de veces que se deben desplazar las naves cada vez que se usen estos procedimientos. ¿Cuál puede ser un buen nombre para identificar el propósito del parámetro?

4. Modificá los procedimientos `mover a la derecha` y `mover a la izquierda` para que ahora también se pueda cambiar el tiempo que la nave se encuentra detenida cada vez que los usemos. ¿Cuál puede ser un buen nombre para identificar el nuevo parámetro que tenés que agregar?

5. ¿Cómo podrías hacer para tener un solo procedimiento en lugar de tener dos, uno para la izquierda y otro para la derecha? Hacelo en la computadora y después copialo acá abajo.

6. Ahora vamos a dejar todo funcionando como en el Space Invaders. Usando el procedimiento del punto 5, tenés que lograr que las naves enemigas se muevan ininterrumpidamente 5 veces a la derecha y 5 veces a la izquierda. ¿Cómo quedó el programa?

Actividad 2

Corregimos y testeamos el movimiento del proyectil



DE A DOS

OBJETIVOS

- Presentar el envío de mensajes para la comunicación entre objetos.
- Introducir nociones de testeo de programas.

MATERIALES



Computadora



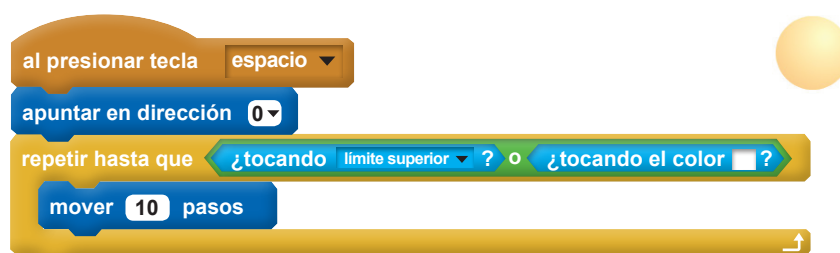
Scratch



Ficha para estudiantes

DESARROLLO

Comenzamos recordando cómo se programó el movimiento del proyectil en la actividad “Disparamos el proyectil”. Allí se consiguió que el proyectil se moviera hasta chocar con una nave enemiga o con el borde superior de la escena. El programa se muestra a continuación.



Solución de la actividad “Disparamos el proyectil”

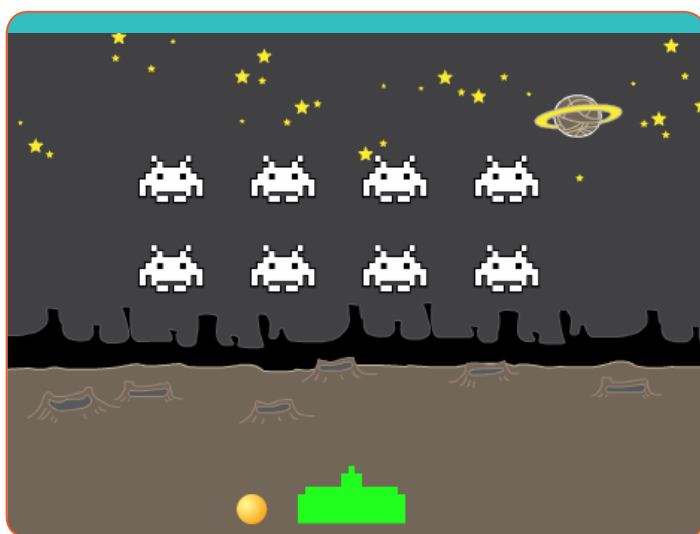
Como ya vimos, al contar con solo un proyectil, este debe volver a la nave verde para poder ser disparado nuevamente. Sin embargo, con el programa desarrollado hasta el momento, cuando el proyectil impacta contra una nave enemiga o con el borde de la pantalla, se detiene por completo. Invitamos a los estudiantes a que resuelvan la consigna 1 de la ficha de la actividad. Allí se les pide que el proyectil regrese a la nave verde luego de una colisión.

Es probable que los estudiantes encuentren el bloque `ir a []` de la categoría *Movimiento*. La instrucción se tiene que usar con el argumento `nave principal`. Preguntamos: “¿Cuándo tiene que ejecutarse?”. Luego de una colisión. “Entonces, ¿dónde deberíamos colocar este bloque?”. A continuación del bloque `repetir hasta que < >`, como muestra la siguiente figura.



Resolución de la consigna 1

A continuación les pedimos que resuelvan la consigna 2, en la que se solicita que testeen el videojuego en busca de errores de funcionamiento. Es muy probable que los estudiantes observen que, luego de que el proyectil se posiciona junto a la nave, no acompaña sus desplazamientos. La siguiente figura ilustra el problema.



Error en nuestro videojuego

Con lo hecho hasta el momento, el proyectil se reubica junto a la nave principal solo después de un choque. Preguntamos a los estudiantes: “¿En qué momento el proyectil debería dejar de seguir a la nave?”. Solo cuando se lo dispara. Es decir, tendría que desplazarse junto a ella hasta que el usuario presione la tecla *espacio*. En la categoría *Sensores* se encuentra el bloque `¿tecla [espacio] presionada?` que se usará para identificar el momento en el que el proyectil debe dejar de seguir a la nave. Es el momento de preguntarles: “¿Se animan a extender el programa para que el proyectil se mueva junto a la nave?” e invitarlos a resolver la consigna 3 de la ficha.

tecla espacio presionada?

Sensor para detectar que se presiona una tecla

En Scratch, los objetos pueden enviar **mensajes** que podrían ser recibidos por diferentes objetos. Por ejemplo, el proyectil podría enviar un mensaje que dé cuenta de que ha alcanzado una nave enemiga cada vez que impacta contra alguna. Luego, diferentes objetos podrían, cada uno a su modo, hacer algo al recibir este mensaje. Por ejemplo, la nave impactada podría desaparecer y la nave principal, volver al centro del borde inferior de la pantalla. En esta parte de la actividad vamos a usar un truco que deberíamos compartir con los alumnos, pues es muy improbable que se les ocurra. El truco consiste en que el mensaje del proyectil sea manejado por el mismo proyectil.

En la categoría *Eventos* se encuentra el bloque `enviar []`. Al desplegar la flecha se puede crear un nuevo mensaje. En el ejemplo, usaremos el nombre *seguir nave*. En la siguiente figura se puede observar la solución del desafío.



El proyectil sigue a la nave luego de alcanzar una nave enemiga o el límite superior de la pantalla

Invitamos a los estudiantes a testear el comportamiento del proyectil. Inmediatamente notarán que, cuando comienza el juego, el proyectil no acompaña los movimientos de la nave. Con el programa tal como está, los dos objetos se desplazan juntos solo después de que el proyectil ha sido disparado al menos una vez. Solicitamos a los estudiantes que resuelvan la consigna 4 de la ficha, que pide que el proyectil siga a la nave desde que el juego comienza. Para conseguirlo alcanza con enviar el mensaje *seguir nave* cuando se presiona la bandera verde.



El proyectil sigue a la nave al comenzar una partida

CIERRE

Reflexionamos con los estudiantes: “¿Alguna vez les ha pasado que, al usar un programa en la computadora o una aplicación del celular, se han encontrado con algún comportamiento raro, que no esperaban? Es común que se cuelen errores al programar; es algo que les sucede incluso a programadores muy experimentados. Por eso es muy importante testear nuestros programas, para poder verificar que funcionen como nosotros esperamos”.

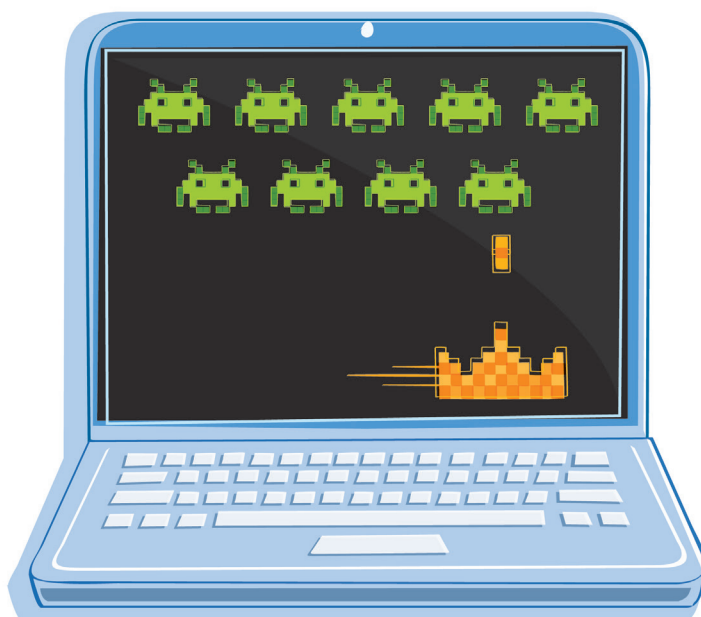
NOMBRE Y APELLIDO:

CURSO:

FECHA:

CORREGIMOS Y TESTEAMOS EL MOVIMIENTO DEL PROYECTIL

En esta actividad vas a lograr que el proyectil, luego de colisionar, regrese automáticamente a donde se encuentra la nave.



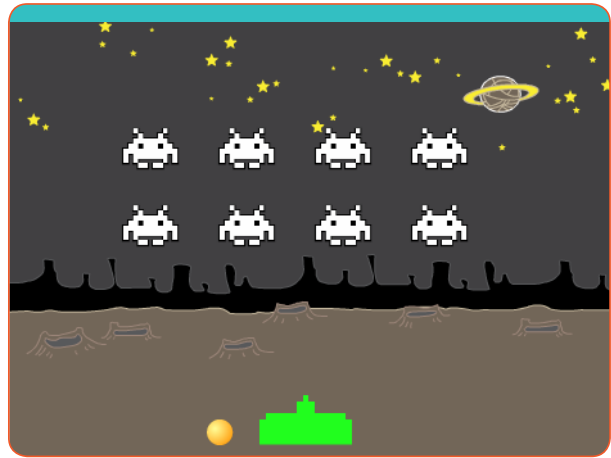
1. Hací que el proyectil, luego de impactar contra una nave enemiga o contra la parte superior de la pantalla, regrese a donde está la nave principal. ¿Cómo lo lograste?

NOMBRE Y APELLIDO:

CURSO:

FECHA:

2. Probá tu juego. Ejecutá el programa y detectá errores. ¿Qué problemas encontraste?



3. Intentá que el proyectil siga a nuestra nave hasta que se presione la barra espaciadora. ¿Enviaste mensajes para conseguirlo? ¿Cómo?

4. Ahora tenés que conseguir que el proyectil siga a nuestra nave apenas comienza el juego. ¿Cómo lo hiciste? ¿Cuándo comienza el juego?

Actividad 3

Eliminamos naves enemigas



DE A DOS

OBJETIVOS

- Programar respuestas de un programa frente a estímulos externos.
- Presentar la noción de secuencialidad durante la ejecución de un programa.

MATERIALES



Computadora



Scratch



Ficha para estudiantes

DESARROLLO

En la actividad “Corregimos y testeamos el movimiento del proyectil”, conseguimos que el proyectil se comportara de la manera buscada. En esta actividad nos vamos a encargar de que las naves enemigas desaparezcan cuando son alcanzadas por un proyectil.

Comenzamos preguntando a los estudiantes “¿Qué sucede en el Space Invaders cuando un proyectil impacta contra una nave enemiga?”. Cuando esto ocurre, la nave enemiga se desintegra. Proponemos a los estudiantes que programen las naves enemigas de forma tal que desaparezcan cuando las alcanza un proyectil. Luego de un tiempo, les preguntamos si pudieron resolver el desafío. Es probable que no lo hayan logrado del todo, pero puede ser que hayan encontrado algunos bloques que son útiles para alcanzar el objetivo planteado. Por ejemplo, el bloque `¿tocando []?` de la categoría *Sensores* y el bloque `esconder` de la categoría *Apariencia*. Con nuestra ayuda, los estudiantes reconocerán que hay que usar estos bloques dentro de una sentencia condicional, como muestra la siguiente figura.



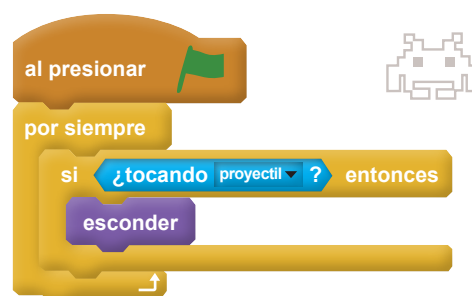
Bloques para ocultar una nave enemiga al colisionar con un proyectil

Continuamos preguntando a los estudiantes: “¿A partir de qué momento se tienen que ejecutar estas instrucciones?”. En el instante en el que comienza la ejecución del videojuego, es decir, al presionar la bandera verde. Proponemos a los estudiantes que agreguen el bloque `al presionar bandera verde` y testeen el programa, que es probable que sea similar al de la siguiente figura.

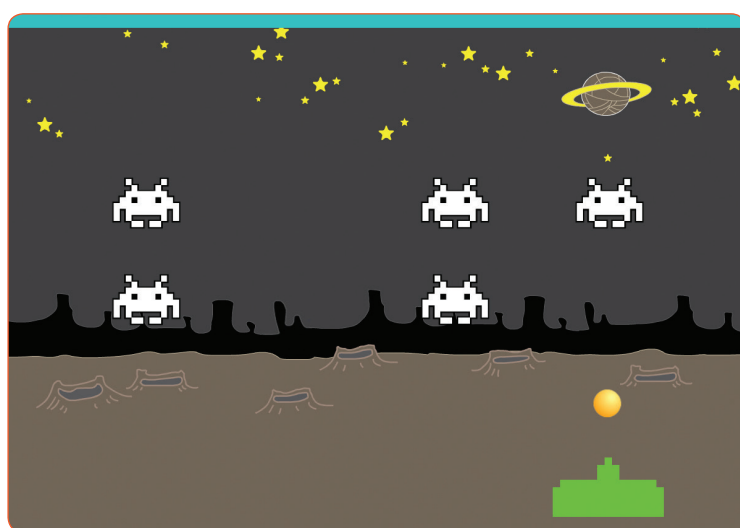


Posible propuesta (incorrecta) de solución de la consigna 1

Luego de unos minutos, les preguntamos a los estudiantes si sus videojuegos funcionan bien. En el programa de la figura anterior, las naves enemigas no desaparecen cuando las alcanza un proyectil. Esto se debe a que la condición `¿tocando [proyectil]?` solo se chequea una única vez al comenzar la ejecución del programa. El chequeo debe hacerse todo el tiempo, pues *a priori* no sabemos cuándo un proyectil impactará contra una nave enemiga. A la derecha se muestra cómo hacerlo.



Solución de la consigna 1



Eliminación de las naves enemigas

Transcurridos unos minutos en los que los estudiantes prueban sus programas, les preguntamos si notan algo extraño. Se espera que algunos adviertan que, en algunas ocasiones, cuando el proyectil choca con alguna nave enemiga, la nave desaparece pero el proyectil sigue su marcha. Esto sucede a pesar de que en la actividad “Corregimos y testeamos el movimiento del proyectil” nos encargamos de que, al chocar contra una nave enemiga, el proyectil se posicionase junto a la nave principal. Les preguntamos a los estudiantes: “¿Por qué creen que sucede esto?”. Si bien Scratch nos permite disparar distintas ejecuciones de bloques en apariencia en un mismo momento (por ejemplo, al usar varias instancias del bloque `al presionar bandera verde`), lo cierto es que en la computadora se ejecutan primero unas y después otras. En este caso lo que sucede es que (i) en primer lugar, la nave enemiga chequea si hay colisión; (ii) como sí hay colisión, la nave se esconde; y (iii) el proyectil chequea si hay colisión. Al momento de hacer este último chequeo ya no hay colisión, porque la nave enemiga desapareció previamente de la pantalla.

Hay una artimaña que podemos compartir con los estudiantes para que completen sus programas. Si inmediatamente después de chequear la condición `¿tocando [proyectil]?` pausamos la ejecución del programa de la nave, le damos tiempo al proyectil para que pueda detectar la colisión. En Scratch podemos pausar una ejecución con el bloque `esperar () segundos` de la categoría *Control*. La siguiente figura muestra una solución completa del desafío planteado.



Solución completa del desafío

CIERRE

A modo de cierre, reflexionamos con los estudiantes acerca de los ciclos infinitos que se agregaron al juego en esta actividad y en algunas de las anteriores. “¿Los ciclos `por siempre` existen a nuestro alrededor? Si nos ponemos a pensar, estamos rodeados de objetos programados para repetir indefinidamente ciertas instrucciones. Por ejemplo, los semáforos están programados para alternar las luces en forma indefinida. Hay situaciones en que el ciclo infinito se ve interrumpido por algún factor externo, como por ejemplo un corte de luz. ¿Se les ocurre un ejemplo distinto al del semáforo?”.

NOMBRE Y APELLIDO:

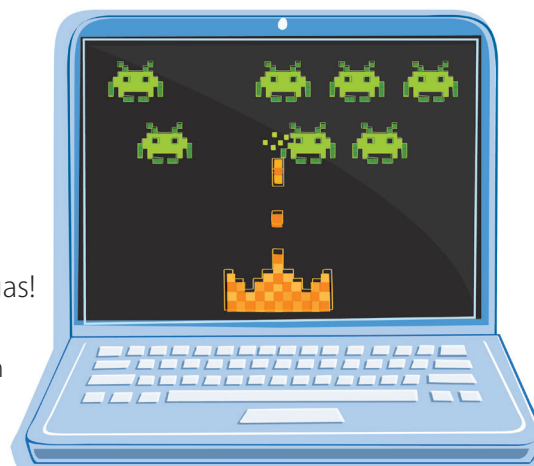
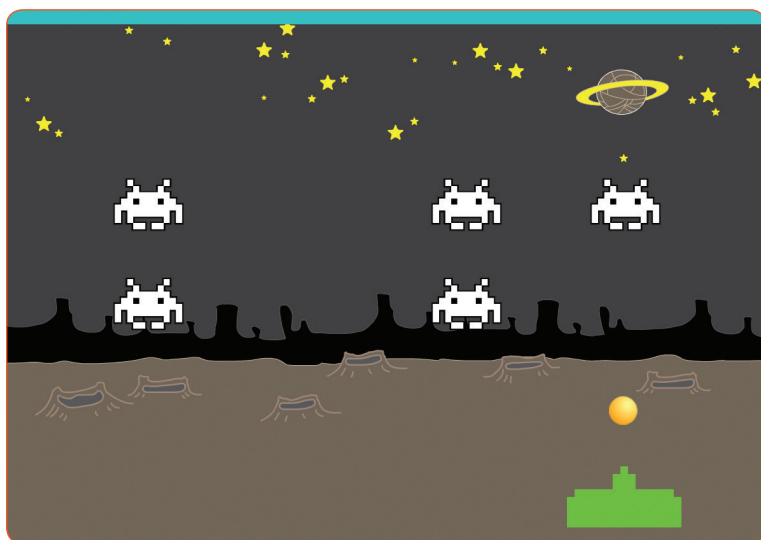
CURSO:

FECHA:

ELIMINAMOS NAVES ENEMIGAS

¡Ahora nos vamos a encargar de pulverizar a las naves enemigas!

1. Tenés que programar las naves enemigas para que desaparezcan de la pantalla cuando las alcance un proyectil.



PISTA

Investigá si pueden servir los bloques

¿tocando proyectil ?

esconder

2. ¿Cumpliste el objetivo? Contá cómo lograste que las naves enemigas desaparezcan al ser alcanzadas por un proyectil.

Actividad 4

Sumamos puntos



DE A DOS

OBJETIVOS

- Usar variables en un proyecto de programación.
- Diferenciar las distintas instancias de uso de variables: inicialización, lectura y modificación.

MATERIALES



Computadora



Scratch



Ficha para estudiantes

DESARROLLO

En la actividad “Eliminamos naves enemigas” conseguimos que las naves enemigas desaparezcan cuando las alcanza un proyectil. En el juego original, además, a medida que las eliminamos, vamos acumulando puntos. En esta actividad vamos a incorporar el puntaje en nuestra versión del Space Invaders.

Comenzamos preguntando a los estudiantes “¿Qué juegos conocen en los cuales se suman puntos?”. Una vez que hayan respondido, les contamos que en esta actividad agregaremos puntaje a nuestro videojuego. Preguntamos: “¿Cuándo sumamos puntos en el Space Invaders? ¿Cuántos puntos tenemos al comenzar el juego?”. Los puntos se consiguen a medida que eliminamos naves enemigas, y al comenzar el juego el puntaje es 0. Les pedimos que completen la consigna 1 de la ficha de la actividad, en la que tienen que contestar estas preguntas.

A continuación, les proponemos que sigan con la consigna 2. Allí se les pide que agreguen puntaje al videojuego. Deben darse cuenta de que, para completar el desafío, necesitan una variable. Luego de unos minutos, consultamos si pudieron resolver el desafío. Es probable que no. En tal caso, los guiamos con preguntas: “¿Por qué no pudimos llevar la cuenta del puntaje? ¿Qué nos está haciendo falta?”. Conducimos la discusión para que los estudiantes comprendan que se necesita un lugar donde guardar el puntaje. Además, es necesario que el puntaje se modifique cada vez que un proyectil impacta en una nave enemiga.

Para crear variables en Scratch, dentro de la categoría *Datos* está el botón *Crear una variable*. Al presionarlo hay que (i) ponerle un nombre a la variable y (ii) definir quiénes podrán leer y modificar su contenido. Es muy importante utilizar nombres que identifiquen el propósito de la variable. En este caso, *puntaje* sería un nombre adecuado. Además, se debe elegir si la variable será visible para todos los objetos de nuestro programa o solo para el que se encuentra seleccionado en la galería de *Objetos*. En este caso es conveniente seleccionar *Para todos los objetos*, porque cada una de las naves enemigas, al ser impactada, tendrá que cambiar su valor.

Crear una variable

Variable nueva

Nombre de la variable:

☒ Para todos los objetos
 ☐ Sólo para éste objeto

Creación de una variable

Una vez que creamos la variable *puntaje*, en la categoría *Datos* aparecerán nuevos bloques, como los que se muestran en la figura de la derecha. El bloque `fijar [puntaje] a []` permite guardar un valor arbitrario en la variable *puntaje*. Por su parte, `cambiar [puntaje] por ()` suma el segundo argumento¹ al valor guardado en la variable hasta ese momento.



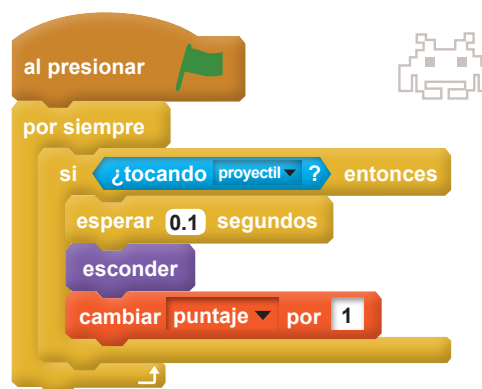
Bloques que se pueden utilizar a partir de la creación de una variable

Preguntamos a los estudiantes para que reflexionen: “Al comenzar el juego el puntaje tiene que ser cero. ¿Cuándo comienza nuestro juego?”. Cuando presionamos la bandera verde. Como la variable *puntaje* es visible para todos los objetos, a cualquiera de ellos se le puede agregar el fragmento de programa que establece su valor inicial. Sin embargo, como la variable *puntaje* no está explícitamente asociada a un objeto de la escena, una alternativa prolija es inicializarla como programa del escenario. Al igual que con el resto de los objetos, para lograrlo hace falta que el escenario se encuentre seleccionado. Por otro lado, para que el puntaje se pueda ver dentro de la escena, hay que usar el bloque `mostrar variable [puntaje]`.



Inicialización de la variable *puntaje*

Preguntamos a continuación: “¿Cuándo cambia el puntaje?”. Cada vez que impactamos una nave enemiga. Explicamos: “En la actividad ‘Eliminamos naves enemigas’ conseguimos que las naves desaparecieran de la escena cuando las alcanzaba un proyectil. En el mismo fragmento del programa deberíamos también incrementar el puntaje”. Es importante notar que la parte del programa a la que nos referimos está asociada a las naves enemigas. Por lo tanto, el cambio hay que realizarlo en cada una de ellas.

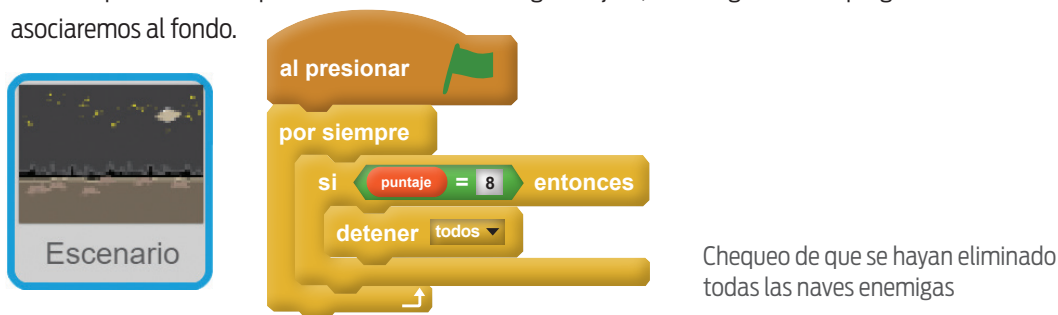


Sumamos 1 punto al impactar una nave

¹ Fijar [] a [] y cambiar [] por [] usan 0 y 1 como valores por defecto para sus segundos parámetros.

Para completar el juego todavía falta establecer cuándo termina. Explicamos a los estudiantes: “En el Space Invaders pasamos de nivel cuando eliminamos todas las naves enemigas, que en nuestra versión del juego son 8. ¿Cómo podemos saber si se eliminaron todas?”. Como el puntaje coincide con la cantidad de naves que se van eliminando, alcanza con ir chequeando si se alcanzó el valor 8. Proponemos a los estudiantes que resuelvan la consigna 3 de la ficha, que pide modificar el programa de forma tal que, cuando se hayan eliminado todas las naves, el juego concluya.

Una posible solución es ir consultando permanentemente si el puntaje es igual a 8, para lo cual necesitamos usar el bloque `[] = []` de la categoría *Operadores*. Además, en caso de que así sea, detendremos el juego con el bloque `detener [todos]`, que interrumpe la ejecución de todos los objetos. Como este chequeo no está explícitamente asociado a algún objeto, este fragmento de programa también lo asociaremos al fondo.



Preguntamos a los estudiantes: “¿Qué sucede si varios jugadores quieren jugar al Space Invaders? ¿Hay alguna manera de que obtengan diferentes puntajes o todos terminarán cuando alcancen los 8 puntos?”. Con lo programado hasta ahora, el juego termina únicamente cuando ya no quedan naves enemigas. Por lo tanto, el puntaje de todos los jugadores será siempre el mismo: 8. Proponemos a los estudiantes que resuelvan la consigna 4 de la ficha, que pide limitar la cantidad de disparos posibles a 10. Es decir, ahora el juego terminará, o bien cuando no queden naves enemigas, o bien cuando se agoten los proyectiles.

Para resolver el desafío planteado se necesita crear una nueva variable para ir contando la cantidad de disparos durante el juego. Un nombre apropiado para la variable es *proyectiles disparados*. Inicialmente su valor será 0, y se incrementará en 1 cada vez que se dispare un proyectil. El valor inicial lo establecemos luego de inicializar el valor de la variable *puntaje*, que habíamos asociado al fondo. Por otro lado, el incremento de la variable se incluirá en lo programado sobre el proyectil. Es decir, cada vez que se presione la barra espaciadora, actualizaremos el valor de la variable *proyectiles disparados*.

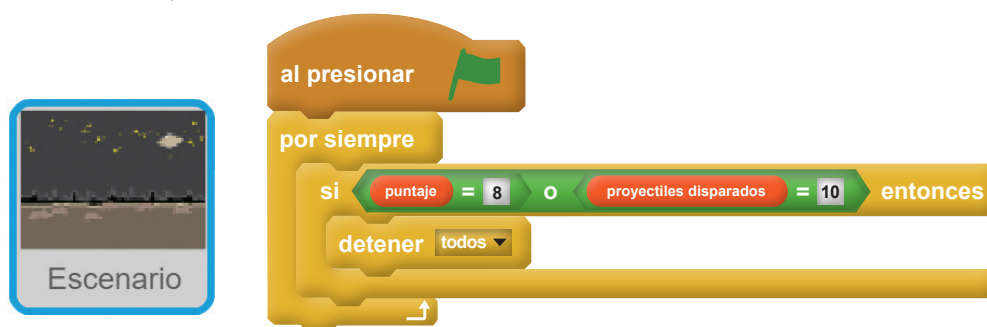


Inicialización de las variables del programa



Contamos los proyectiles que disparamos

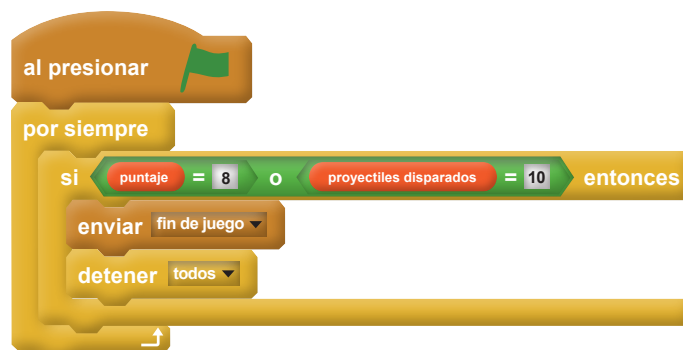
Como el juego tiene que terminar cuando no queden naves enemigas o cuando se agoten los proyectiles, hay que modificar la condición que se chequea para saber si el juego debe detenerse o no. Por lo tanto, hay que construir una nueva condición que use el bloque `< > o < >`. A continuación puede observarse una posible solución.



Fragmento de programa para finalizar una partida

Finalmente, la última consigna de la ficha les pide a los estudiantes que incorporen un título final a las partidas. Específicamente, que oculten los objetos y luego se muestre un cartel con la leyenda “JUEGO FINALIZADO”.

Para resolver el desafío, en primer lugar haremos que, cuando se produzca alguna de las condiciones de finalización de una partida, se envíe el mensaje “fin del juego” a todos los objetos de la escena. Notemos que el pedido de detención a todos los objetos es lo último que se hará en el juego.



Se notifica a todos los objetos que el juego está concluyendo

Cada objeto se ocultará al recibir el mensaje que marca que el programa está en proceso de detención. Todos los objetos desaparecerán usando los mismos dos bloques: `al recibir [fin de juego]` y `esconder`.



Los objetos del juego desaparecen de la pantalla

Solo nos resta ocuparnos del cartel. Lo primero que debemos hacer es crear el objeto que lo representará. Para esto seleccionamos la herramienta de dibujo de Scratch y, una vez dentro, hacemos clic sobre el ícono *Texto*. A continuación, directamente escribimos “JUEGO FINALIZADO” sobre el panel a la derecha.

Nuevo objeto:



Siguiendo una lógica inversa a la del resto de los objetos, apenas se notifica que el juego está finalizando, el cartel se muestra. Y cada vez que una partida arranca, se oculta.



JUEGO FINALIZADO

Programa del cartel de fin de juego

RÚBRICAS

Al final de la ficha de los estudiantes, incluimos una rúbrica para que los alumnos sepan qué estaremos evaluando y qué criterios utilizaremos.

CIERRE

A modo de cierre, se puede usar la siguiente historia para profundizar sobre el concepto de variable: “Me gustaría contarles de qué manera organizo mis pertenencias. Como algunos de ustedes, en mi dormitorio tengo un armario con varios cajones. Los cajones son pequeños, por lo que en cada uno de ellos puedo guardar solo un objeto. Con esto quiero decir que en un cajón guardo solo una remera, en otro solo tengo un pantalón y así con todos los cajones. Además, como mi memoria es muy mala, a cada cajón le pongo una etiqueta que describe lo que está guardado. Al programar, disponemos de algo similar a un armario con cajones, y en cada uno de ellos podemos guardar diferentes datos o valores. Los cajones vendrían a ser nuestras **variables**. Imagínense, si el profe necesita etiquetar sus cajones para saber qué guarda en cada uno, lo importante que puede ser para nosotros ponerle a cada variable un nombre que describa su propósito. En el caso del armario nos facilitará saber dónde guardamos la ropa; un buen nombre para nuestra variable nos facilitará leer nuestros programas y comprender qué estamos guardando o modificando”.

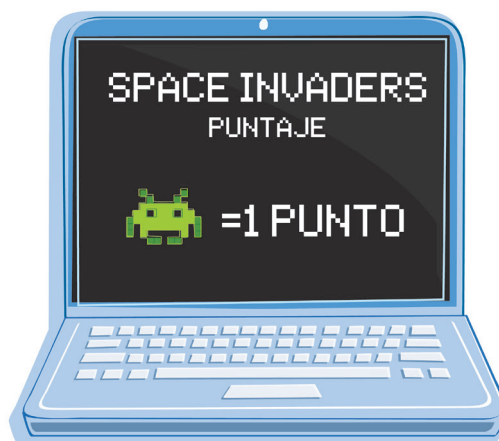
NOMBRE Y APELLIDO:

CURSO:

FECHA:

SUMAMOS PUNTOS

¿Cómo podemos agregar una pizca de adrenalina a nuestro juego? ¡Ahora vamos a incorporar puntaje al Space Invaders!



1. ¿Cuándo sumamos puntos en el Space Invaders? ¿Cuál debería ser el puntaje al empezar el juego?

2. Ahora agregá puntaje a tu videojuego. Para hacerlo, vas a tener que usar una variable. ¿Cómo lo hiciste?

3. Programá el videojuego utilizando el bloque `[] = []` y la variable en la que guardás el puntaje, para que cuando se hayan eliminado todas las naves se detenga la ejecución de todos los objetos. ¿Cómo quedó el programa?



NOMBRE Y APELLIDO:

CURSO:

FECHA:

4. Hagamos el juego más entretenido. Cambiá el programa para que solo se puedan hacer diez disparos. Así, el juego terminará tanto si no quedan naves enemigas como si no nos quedan proyectiles para disparar.
5. Vamos a cerrar el juego con un estilo que esté a su altura. Una vez que los personajes se detienen, en primer lugar tenés que hacer que se escondan. A continuación, debe aparecer un cartel que diga “JUEGO FINALIZADO”.

PISTA

Para resolver la consigna 4, vas a tener que crear una nueva variable y usar el bloque



JUEGO FINALIZADO

EVALUACIÓN

Estas son las tareas y la forma de calificar que tu docente considerará para evaluar cómo resolvés las actividades.

CALIFICACIÓN	ES NECESARIO TRABAJAR MÁS	BUENO-MUY BUENO	EXCELENTE
Movimientos de las naves	El estudiante no logra que ninguna nave se mueva.	El estudiante logra que se muevan las naves, pero no puede definir procedimientos para cada uno de los movimientos, o bien no puede dar los parámetros de movimiento de forma correcta.	El estudiante logra que se muevan las naves creando procedimientos que luego reutiliza. Además, logra definir parámetros correctamente.
Acciones de los proyectiles	El estudiante no logra que un proyectil siga a un objeto.	El estudiante logra desarrollar un proyectil que sigue a un objeto, pero no logra identificar el evento que indica que tiene que dejar de seguirlo. No hay tiempo de espera entre una acción y otra.	El estudiante logra desarrollar un proyectil que siga al objeto y que se detenga al presionar una tecla. Consigue asignar tiempo de espera entre una acción y otra realizando así la tarea de gestión de tiempo.
Asignación de puntajes (definición, uso y modificación de variables)	El estudiante no logra que el juego tenga puntaje.	El estudiante logra que las acciones del jugador generen puntos usando una variable, pero el puntaje no se calcula correctamente.	El estudiante logra que las acciones del jugador sumen puntos que se guardan en una variable y se actualizan correctamente.

REPRESENTACIÓN DE DATOS

SECUENCIA DIDÁCTICA 1

REPRESENTACIÓN DE NÚMEROS

Nos aproximamos al sistema binario
Contamos con ceros y unos

SECUENCIA DIDÁCTICA 2

REPRESENTACIÓN DE TEXTO

Representamos letras
Mensajes secretos

SECUENCIA DIDÁCTICA 3

TRIVIA DE PREGUNTAS

¡Que gire la ruleta!
Gira la ruleta y... ¡no va más!
Agregamos un poco de azar
Identificamos el color elegido
Y finalmente preguntamos

Las computadoras hacen y nos posibilitan hacer cosas con datos. Nos permiten ver, escuchar, crear y editar datos: sacar una foto, ver un video, escuchar una canción o escribir un mensaje de chat. Son capaces de hacer cálculos y nos dejan enviar y recibir contenidos a través de las redes. Al usar una computadora, jugamos en pequeños mundos inventados que solo existen como datos. Estos datos están almacenados en la memoria de la computadora y una parte de ellos se muestra en la pantalla.

Al finalizar este capítulo, los estudiantes podrán representar datos de forma similar a como se hace en una computadora. Representarán números, palabras y colores. Para ello, proponemos diferentes actividades para el aula y un proyecto de programación que les permitirá acercarse a la representación de información de diversa naturaleza.



Secuencia Didáctica 1

REPRESENTACIÓN DE NÚMEROS

Todos los datos que utiliza una computadora se almacenan usando solo dos valores. Estos valores se suelen representar con 0 y 1, pero también podrían ser sí y no, blanco y negro, puño y palma, etc. Solo es necesario elegir dos signos bien distintos para que podamos distinguirlos sin ninguna duda.

El uso de los dígitos 0 y 1 es tan común que a partir de ellos se inventó una palabra: **bit**, formada por las dos primeras letras y la última letra de *dígito binario* en inglés, *binary digit*. Un bit puede tener solo dos valores y por tal motivo se los llama *dígitos binarios*.

Esta secuencia didáctica comprende dos actividades sin computadora para que los estudiantes se familiaricen con el sistema de numeración binario.

OBJETIVOS

- Mostrar cómo las computadoras digitales representan números.
- Representar números usando solo los símbolos 0 y 1.

Actividad 1

Nos aproximamos al sistema binario¹



GRUPAL (5)

OBJETIVOS

- Presentar el sistema de numeración binario.
- Construir números binarios a partir de números decimales.

MATERIALES



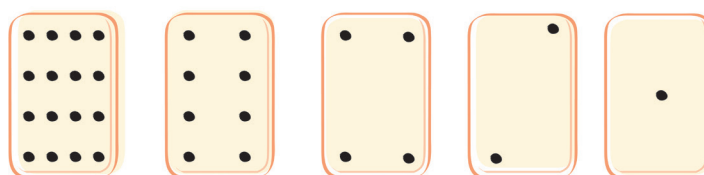
5 cartas tamaño A4 o mayor



Ficha para estudiantes

DESARROLLO

Para esta actividad vamos a necesitar un juego de cinco cartas de tamaño A4 o mayor para que toda la clase pueda observarlas con facilidad. Cada naipes tiene uno o más puntos de un lado y nada del reverso, como muestra la siguiente figura.



Juego de cartas de la actividad

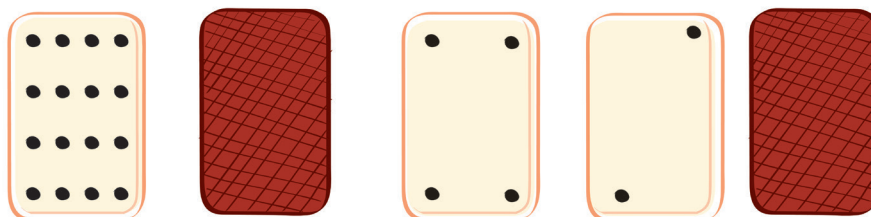
Presentamos los naipes a la clase y elegimos cinco estudiantes para que sostengan las cartas frente al resto. Las cartas deben estar en el orden que se indica en la ilustración, pero la clase debe ver solo el reverso de las cartas. Es decir, los puntos tienen que quedar ocultos para el resto de la clase.



Pedimos al estudiante que sostiene la carta con un solo punto que la dé vuelta y luego solicitamos lo mismo al que está a su lado. Antes de pedir al tercer estudiante que gire su carta, preguntamos a la clase: “¿Cuántos puntos creen que va a tener la tercera carta?”. Probablemente contesten: “Tres”. Pedimos al tercer estudiante que dé vuelta su carta, que tiene cuatro puntos. Cada carta tiene el doble de puntos que la carta que está a su derecha. Repetimos la pregunta, pero ahora sobre la cuarta carta a descubrir y luego sobre la quinta. A esta altura, se espera que los estudiantes puedan deducir que estas tienen ocho y dieciséis puntos respectivamente.

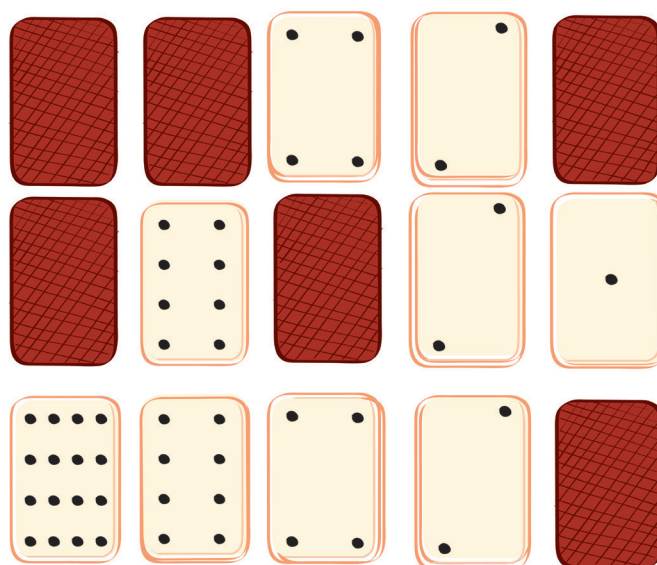
¹ Adaptación de la actividad “Count the Dots” de CS Unplugged, disponible en <https://goo.gl/9w3mvW>.

Tras esta presentación, repartimos a los alumnos la ficha de la actividad y les pedimos que recorten las cartas. Luego les preguntamos: “¿Cómo pueden conseguir que solo 22 puntos estén visibles?”. Esperamos que propongan la siguiente solución.



Representación del número 22 con el juego de cartas

Cuando resuelvan esta pregunta, les proponemos que representen otras cantidades como, por ejemplo, 6, 11 y 30. Para responder correctamente deberían disponer las cartas de las siguientes maneras.



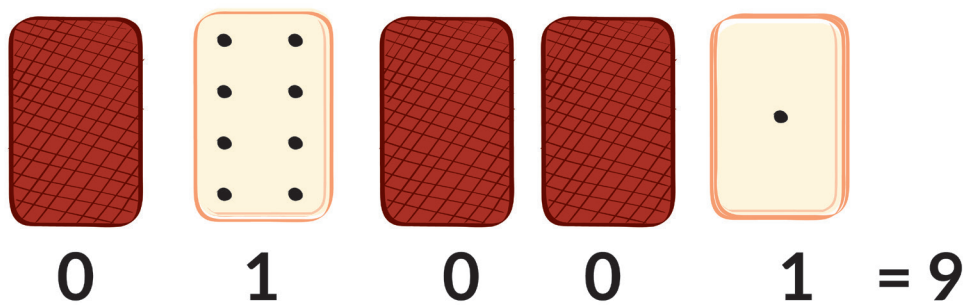
Representaciones de los números 6, 11 y 30 con el juego de cartas

Preguntamos: “¿Se les ocurre algún número que no pueda ser representado?”. Cualquier número mayor o igual a 32 no puede representarse, dado que el total de puntos con las 5 cartas es 31. “¿Cómo pueden representar el cero?”. Con todas las cartas boca abajo. “¿Y el 31?”. Con todas las cartas boca arriba.

A continuación, pedimos a los estudiantes que formen grupos de 5 integrantes. El desafío que se les propone es que, usando las cartas, representen todos los números entre 0 y 31 en orden, uno a continuación del otro. Uno de ellos debe manipular las cartas mientras el resto mira atentamente cómo los naipes cambian de lado, observando que algunos son más usados que otros.

Con nuestra ayuda, los estudiantes deberían ir descubriendo algunos patrones. Por ejemplo, que cada carta gira la mitad de las veces que la carta a su derecha. Además, pueden notar que una de cada dos veces necesitan usar la carta con un solo punto. Aquí podemos relacionar el concepto matemático de paridad: cada vez que la carta con un punto está boca arriba, estaremos representando un número impar y cada vez que esté boca abajo, un número par.

Finalmente, mostramos que usar únicamente los símbolos 0 y 1 nos alcanza para expresar todos los números que pueden representarse con las cartas. Para eso, copiamos en el pizarrón el siguiente dibujo.



El estado de las cartas representado por una secuencia de ceros y unos

Les hacemos notar a los estudiantes que debajo de cada carta boca arriba hay un 1 y debajo de cada carta boca abajo, un 0. Usamos estos valores para indicar los dos estados posibles en los que puede estar una carta: boca arriba y boca abajo. Aun si borramos las cartas, la información sobre si cada una estaba boca arriba o boca abajo queda codificada en la secuencia de bits.

CIERRE

Les contamos a los estudiantes que las computadoras usan solo dos valores para representar toda la información que manipulan. En general, nos vamos a referir a esos valores usando los símbolos 0 y 1. Internamente, los circuitos electrónicos de la computadora utilizan dos niveles de voltaje distintos.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

NOS APROXIMAMOS AL SISTEMA BINARIO



¿Sabías que existen distintas formas de contar? Todo lo que ves o escuchás en la computadora, palabras, imágenes, números e incluso películas ¡se almacena usando solo ceros y unos!

1. Cortá las cartas que están al final de la ficha. Ordenalas de mayor a menor, como lo hace el chico en la figura de la derecha. Ahora da vuelta las cartas de forma tal que, manteniéndolas en el mismo orden, solo se vean 22 puntos. ¿Qué naipes deben quedar con los puntos visibles y cuáles al revés?

2. Averiguá cómo obtener los números 6, 11 y 30. ¿Hay más de una forma de hacerlo? ¿Cuál es el número más grande que podés conseguir con las cartas? ¿Cuál es el más pequeño? ¿Hay algún número que no puedas armar entre el más pequeño y el más grande?

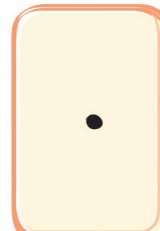
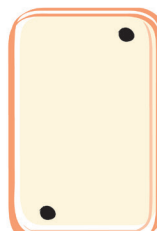
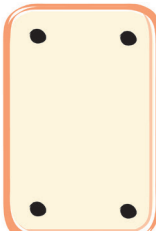
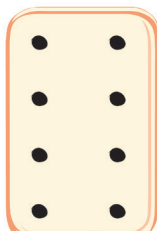
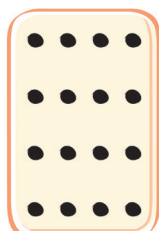
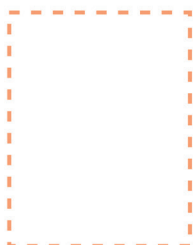
3. Ahora tratá de obtener los números 1, 2, 3 y 4 en orden. ¿Podés pensar un método lógico y confiable para dar vuelta las cartas de forma tal de aumentar de a uno cualquier número?

NOMBRE Y APELLIDO:

CURSO:

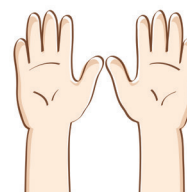
FECHA:

4. Mirá las cartas de nuevo. Si tuvieras que colocar la siguiente carta de la secuencia, ¿cuántos puntos debería tener? ¿Qué pasa con la carta que le seguiría a esa? ¿Cuál es la regla que estás siguiendo para hacer las nuevas cartas? ¡La cantidad de puntos por carta crece a gran velocidad!



5. Si observás cuidadosamente la secuencia, podés encontrar una relación interesante. Sumá: $1 + 2 + 4 + 8$ y escribí el resultado. Ahora probá sumar $1 + 2 + 4 + 8 + 16$. ¿Qué pasa cuando sumás todos los números desde el principio?

6. Con tus dedos podés llegar mucho más allá de 10. Si usás los dedos de una sola mano de modo que cada uno de ellos represente el valor de una de las cartas con puntos, podés contar de 0 a 31. Son 32 números en total. ¡No te olvides de que el 0 es un número también! Ahora, tratá de contar en orden con los dedos. Si un dedo está hacia arriba es un 1 y si está hacia abajo es un 0. ¿Qué pasa si contás con los 10 dedos de tus manos? ¡Podés ir desde 0 a 1023! Es decir, 1024 números.



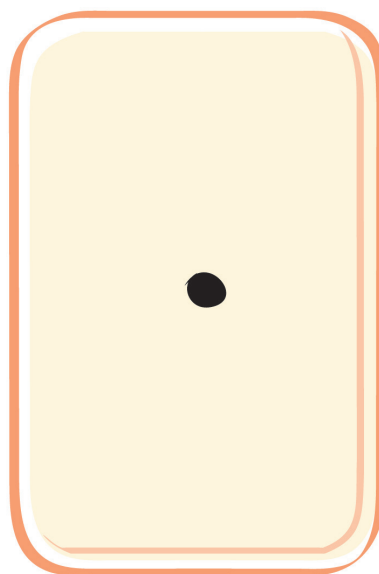
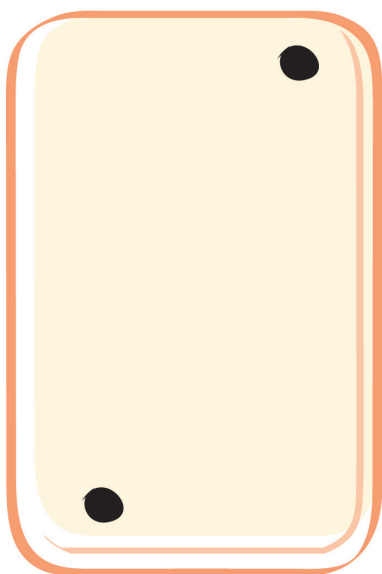
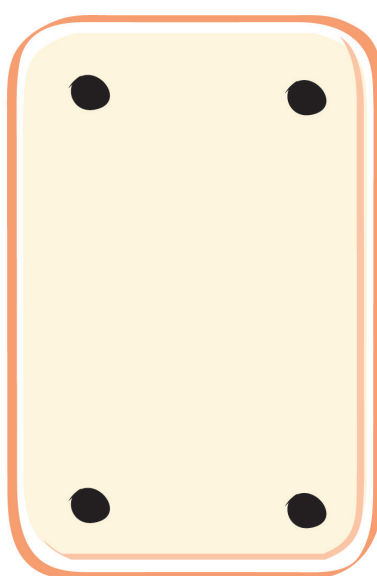
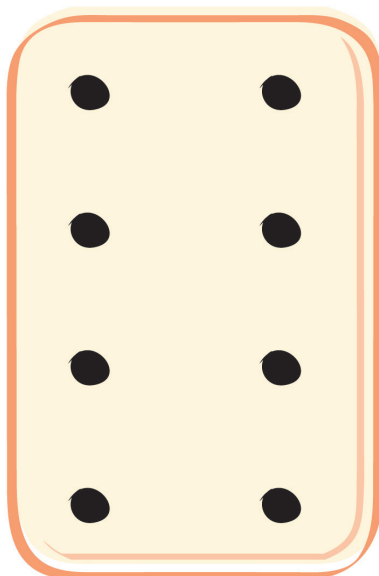
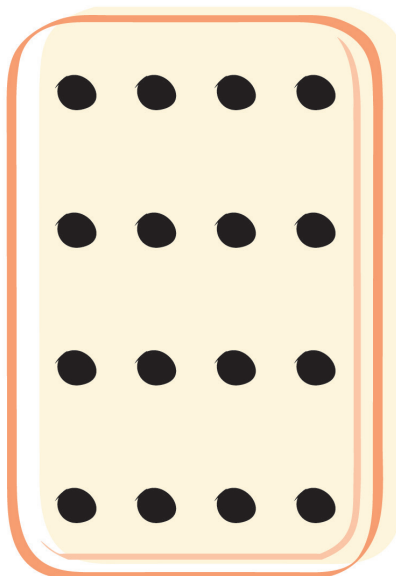
Si con los dedos de dos manos podés contar hasta 1023, ¿cuál es el número más grande que puede alcanzar un extraterrestre de cuatro manos si tiene cinco dedos en cada una?

NOMBRE Y APELLIDO:

CURSO:

FECHA:

Cartas para recortar



Actividad 2

Contamos con ceros y unos



INDIVIDUAL

OBJETIVOS

- Realizar una práctica con el sistema binario.
- Mostrar que cualquier número entero puede representarse con dos símbolos.
- Evidenciar que el uso de los símbolos 0 y 1 es arbitrario.

MATERIALES



Papel



Lápiz



Ficha para estudiantes

DESARROLLO

Comenzamos formulando a los estudiantes la siguiente pregunta: “¿Cómo contamos cuando usamos el sistema decimal?”. Les pedimos que cuenten de 0 a 10 y, a medida que lo hacen, vamos anotando en el pizarrón cada uno de los números. Una vez que estén todos escritos, les preguntamos si notan alguna diferencia entre el 10 y el resto de los números. Es esperable que algún estudiante conteste que, mientras que para los que están entre el 0 y el 9 se usa solo un dígito, en el caso del 10 se usan dos.

A continuación preguntamos: “¿Cómo harían para contar si únicamente estuvieran disponibles el 0 y el 1? ¿Cómo representarían el número 2?”. Algún estudiante podría darse cuenta de que una forma de hacerlo es escribiendo 10.

Copiamos en el pizarrón la siguiente tabla y pedimos a la clase que vaya indicando cómo se completa la última columna, de a una fila por vez.

CANTIDAD REPRESENTADA	SISTEMA DECIMAL	SISTEMA BINARIO
	0	
•	1	
• •	2	
• • •	3	
• • • •	4	
• • • • •	5	

Así queda la tabla completada correctamente.

CANTIDAD REPRESENTADA	SISTEMA DECIMAL	SISTEMA BINARIO
	0	0
•	1	1
• •	2	10
• • •	3	11
• • • •	4	100
• • • • •	5	101

Una vez que hayan completado las 6 filas de la tabla, les entregamos la ficha de la actividad y les pedimos que completen la tabla que allí se encuentra. Al hacerlo, obtendrán la representación binaria de los números 0 al 15, como se muestra a continuación.

CANTIDAD REPRESENTADA	SISTEMA DECIMAL	SISTEMA BINARIO
● ● ● ● ● ●	6	110
● ● ● ● ● ● ●	7	111
● ● ● ● ● ● ● ●	8	1000
● ● ● ● ● ● ● ● ●	9	1001
● ● ● ● ● ● ● ● ● ●	10	1010
● ● ● ● ● ● ● ● ● ● ●	11	1011
● ● ● ● ● ● ● ● ● ● ● ●	12	1100
● ● ● ● ● ● ● ● ● ● ● ● ●	13	1101
● ● ● ● ● ● ● ● ● ● ● ● ● ●	14	1110
● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	15	1111

CIERRE

A modo de cierre, les explicamos a los estudiantes que, en lugar de usar los símbolos 0 y 1, el sistema binario podría usar otros, como por ejemplo ○ y ● para el cero y el uno respectivamente. Siguiendo esta pauta, ● ○ ● ○ codifica el número 10. Esta es la base de la representación de datos: todo lo que pueda estar en dos estados claramente distinguibles puede representar lo que sea. Cuando escribimos en un papel, normalmente usamos 0 para uno de los estados y 1 para el otro. Internamente, la computadora usa dos niveles de voltaje diferentes.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

CONTAMOS CON CEROS Y UNOS

¡Ahora vamos a escribir números tal como lo hacen las computadoras!



1. Completá la tabla usando solo ceros y unos.

CANTIDAD REPRESENTADA	SISTEMA DECIMAL	SISTEMA BINARIO
	0	
•	1	
• •	2	
• • •	3	
• • • •	4	
• • • • •	5	
• • • • • •	6	
• • • • • • •	7	
• • • • • • • •	8	
• • • • • • • • •	9	
• • • • • • • • • •	10	
• • • • • • • • • • •	11	
• • • • • • • • • • • •	12	
• • • • • • • • • • • • •	13	
• • • • • • • • • • • • • •	14	
• • • • • • • • • • • • • • •	15	

NOMBRE Y APELLIDO:

CURSO:

FECHA:

2. Hasta ahora nos las hemos arreglado para representar números usando los símbolos 0 y 1. Pero ¿qué diferencia habría si en su lugar usáramos \bigcirc y \bullet ? ¡La elección de los símbolos es arbitraria! Descubrí cuáles son los números detrás de estas secuencias de símbolos.¹

$$\begin{array}{ccccccccc} \checkmark & \checkmark & \times & \times & \checkmark & = & \square \\ \checkmark = 1 & \times = 0 \end{array}$$

$$\begin{array}{ccccccccc} \text{thumbs up} & \text{thumbs down} & \text{thumbs up} & \text{thumbs down} & = & \square \\ \text{thumbs up} = 1 & \text{thumbs down} = 0 \end{array}$$

$$\begin{array}{ccccccccc} \uparrow & \downarrow & \uparrow & = & \square \\ \uparrow = 1 & \downarrow = 0 \end{array}$$

$$\begin{array}{ccccccccc} + & + & \times & + & = & \square \\ + = 1 & \times = 0 \end{array}$$

$$\begin{array}{ccccccccc} \odot & \bigcirc & \bigcirc & \bigcirc & \bigcirc & = & \square \\ \odot = 1 & \bigcirc = 0 \end{array}$$

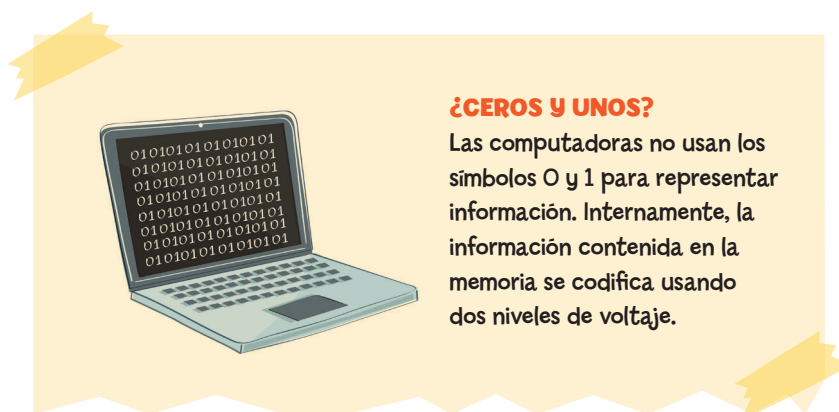
$$\begin{array}{ccccccccc} \curvearrowright & \curvearrowleft & \curvearrowleft & \curvearrowleft & \curvearrowright & = & \square \\ \curvearrowright = 1 & \curvearrowleft = 0 \end{array}$$

$$\begin{array}{ccccccccc} \text{cloud} & \text{sun} & = & \square \\ \text{cloud} = 1 & \text{sun} = 0 \end{array}$$

$$\begin{array}{ccccccccc} \spadesuit & \spadesuit & \spadesuit & \spadesuit & \spadesuit & = & \square \\ \spadesuit = 1 & \diamond = 0 \end{array}$$

$$\begin{array}{ccccccccc} \triangle & \nabla & \triangle & \nabla & \nabla & = & \square \\ \triangle = 1 & \nabla = 0 \end{array}$$

$$\begin{array}{ccccccccc} \text{happy face} & \text{sad face} & = & \square \\ \text{happy face} = 1 & \text{sad face} = 0 \end{array}$$



¿CEROS Y UNOS?

Las computadoras no usan los símbolos 0 y 1 para representar información. Internamente, la información contenida en la memoria se codifica usando dos niveles de voltaje.

¹ Adaptación de la actividad "Working with binary" de CS Unplugged, disponible en <https://goo.gl/D8Q136>.



Secuencia Didáctica 2

REPRESENTACIÓN DE TEXTO

Todos los días usamos textos en nuestros dispositivos digitales: utilizamos servicios de mensajería instantánea, hacemos comentarios en las redes sociales, leemos páginas web y muchas cosas más. ¿Cómo hacen las computadoras para guardar, transmitir y mostrar estos textos si solo saben representar números? Al finalizar esta secuencia didáctica los estudiantes comprenderán las ideas subyacentes a las técnicas de representación de texto usadas en los sistemas digitales.

.....

OBJETIVOS

- Mostrar cómo las computadoras digitales representan texto usando números.
- Realizar prácticas de codificación y decodificación de texto.

.....

Actividad 1

Representamos letras¹



DE A DOS

OBJETIVOS

- Realizar una práctica de codificación de letras con números.
- Mostrar la relación entre una cantidad fija de bits y el número de cosas distintas que pueden representar.

MATERIALES

- Papel
- Lápiz
- Ficha para estudiantes

DESARROLLO

Comenzamos relatando a la clase la siguiente situación. Tomás está atrapado en el último piso de un centro comercial. Es justo antes de Navidad y quiere llegar a su casa con los regalos. Ha intentado llamar a alguien, incluso gritando, pero nadie responde. Se está haciendo de noche y al otro lado de la calle divisa a una chica que está usando una computadora. ¿Cómo puede atraer su atención?

Tomás mira a su alrededor para ver qué podría usar. Entonces se le ocurre una brillante idea: ¡utilizar las luces del árbol de Navidad para enviarle un mensaje! Son cinco luces que puede prender y apagar en forma independiente. ¿Le alcanzan para enviar el mensaje?



Proponemos a los estudiantes que cuenten las letras del alfabeto. Luego, discutimos cómo se pueden codificar las letras a través del uso de números. Les preguntamos: “¿Con qué número podemos representar la A, que es la primera letra del abecedario? ¿Y la B? ¿Y la C?”. Las respuestas serán correctas siempre que, por un lado, los números propuestos sean diferentes para las distintas letras y, por otro, que sean menores a treinta y dos, que es el número más grande que se puede representar con cinco bits (o cinco luces, en este caso).

Les recordamos que Tomás tiene solo cinco luces y que cada luz puede representar un solo bit. El 0 se puede representar con la luz apagada y el 1 con la luz prendida. Además, sugerimos asociar números y letras siguiendo caminos ascendentes, como el orden habitual de los números y el orden alfabético.

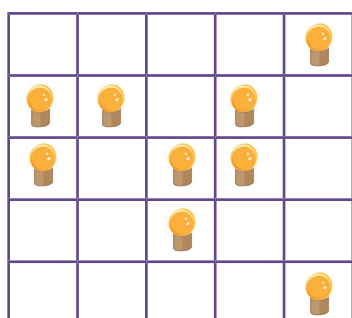
Solicitamos a los estudiantes que se pongan en parejas y les repartimos la ficha de la actividad. Les pedimos que armen una tabla de correspondencia entre las letras y los números entre 0 y 27. Se muestra a continuación una posible codificación del alfabeto usando números. Hacemos notar que, además del alfabeto, se está codificando el símbolo *espacio en blanco*; en este caso, con el número 0.

¹ Adaptación de la actividad “Sending Secret Messages” de CS Unplugged, disponible en <https://goo.gl/D8Q136>.

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	A	B	C	D	E	F	G	H	I	J	K	L	M
14	15	16	17	18	19	20	21	22	23	24	25	26	27
N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z

Codificación de letras del alfabeto como números decimales

Pedimos que uno de los integrantes de la pareja piense una frase para que Tomás pida ayuda. Luego, solicitamos que la codifique usando un sistema binario. La tarea del otro miembro de la pareja es decodificar la frase. Luego intercambiarán roles. A continuación se da la codificación de la palabra *ayuda* usando las luces de Tomás.



Codificación de la palabra *ayuda* usando luces navideñas

Por último, proponemos a los estudiantes que observen el teclado ilustrado en la ficha y que cuenten cuántos símbolos nos permite escribir. Les preguntamos: “¿Cuántos caracteres tendría que poder representar una computadora que use este teclado?”. Les recordamos que los números, el espacio en blanco y los restantes símbolos también son caracteres, y que las letras pueden aparecer en minúscula y en mayúscula. “Entonces, ¿cuántos bits necesita una computadora para almacenar todos los caracteres de este teclado?”.



Son 81 símbolos en total. Por lo tanto, harán falta al menos 7 bits, pues con 6 solo se podrán representar $2^6 = 64$ símbolos diferentes, mientras que con 7 se pueden codificar hasta $2^7 = 128$.

CIERRE

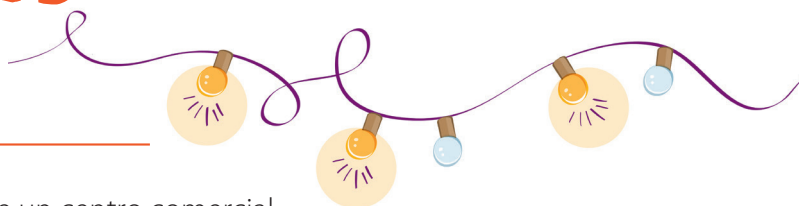
Para finalizar la actividad, contamos a los estudiantes que todos los símbolos del alfabeto se pueden representar a través de una codificación llamada ASCII. ASCII es un acrónimo de la expresión en inglés *American Standard Code for Information Interchange* (Código Estándar Estadounidense para el Intercambio de Información). Se basa en el uso de 7 bits para representar 128 caracteres, pero para algunos idiomas se usan códigos más largos. Actualmente, ASCII ha sido reemplazada por otras codificaciones que incluyen más bits y, por lo tanto, representan un mayor número de símbolos.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

REPRESENTAMOS LETRAS



Tomás está atrapado en el último piso de un centro comercial. Es justo antes de Navidad y quiere llegar a casa con los regalos. Ha intentado llamar a alguien, incluso gritando, pero no hay nadie alrededor. Se está haciendo de noche y al otro lado de la calle ve a una chica trabajando con su computadora. ¿Cómo puede atraer su atención?



Mira a su alrededor y se le ocurre una brillante idea: utilizar las luces del árbol de Navidad para enviarle un mensaje! Son 5 luces que puede prender y apagar de forma independiente. ¿Le alcanzarán para enviar un mensaje a la chica?

1. Indicale a Tomás una forma de codificar las letras del alfabeto español con números. ¡No te olvides de incluir el espacio en blanco para que pueda mandar mensajes con más de una palabra!

0	1	2	3	4	5	6	7	8	9	10	11	12	13
14	15	16	17	18	19	20	21	22	23	24	25	26	27

2. Tomás ya tiene una forma de representar letras usando números. Pensá una frase y escribí la secuencia de números que la componen.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

3. Tomás, además, ya sabe cómo escribir un número usando un sistema binario. Usando solo las luces navideñas, tenés que escribir la frase que pensaste. Tené presente que cada fila representa solo un número.

4. Al igual que Tomás, las computadoras usan números binarios para representar caracteres. ¿Cuántos caracteres tiene que representar una computadora con este teclado? Tené en cuenta que los números, el espacio en blanco y los restantes símbolos también son caracteres, y que las letras pueden aparecer en minúscula y en mayúscula.



5. ¿Cuántos bits necesita una computadora para almacenar todos los caracteres del teclado? ¿Le alcanzan 5 como a Tomás?

Actividad 2

Mensajes secretos

INDIVIDUAL

OBJETIVOS

- Traducir números expresados en sistema binario a sistema decimal.
- Decodificar un texto codificado con números.

MATERIALES

 Papel

 Lápiz

 Ficha para estudiantes

DESARROLLO

Presentamos la actividad a la clase con la siguiente consigna: “Imaginen que trabajan como guardaparques del Parque Nacional Quebrada del Condorito. Un día les avisan que se perdió un montañista que llevaba una máquina para pedir auxilio en caso de extraviarse. Se sabe que esta máquina envía mensajes en clave a través de dos tipos de energía: alta y baja. Alta es 1 y baja es 0. Les llega su mensaje. No hay una computadora en todo el parque. ¡Tienen que descifrar el mensaje como si ustedes fueran la compu!

Repartimos las fichas a los estudiantes y continuamos con el relato: “Para decodificar el mensaje recibido, miren la tabla donde está codificado el alfabeto”.

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	A	B	C	D	E	F	G	H	I	J	K	L	M
14	15	16	17	18	19	20	21	22	23	24	25	26	27
N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z

Codificación de las letras del alfabeto usando números decimales

Pedimos a los estudiantes que decodifiquen el mensaje recibido. Al hacerlo descubrirán que el montañista se encuentra perdido en el río.

1	0	0	0	1	P
0	0	1	0	1	E
1	0	0	1	1	R
0	0	1	0	0	D
0	1	0	0	1	I
0	0	1	0	0	D
1	0	0	0	0	O
0	0	0	0	0	
0	0	1	0	1	E
0	1	1	1	0	N
0	0	0	0	0	
0	0	1	0	1	E
0	1	1	0	0	L
0	0	0	0	0	
1	0	0	1	1	R
0	1	0	0	1	Í
1	0	0	0	0	O

Codificación de la frase “perdido en el río” con números binarios

RÚBRICAS

Al final de la ficha de los estudiantes, incluimos una rúbrica para que los alumnos sepan qué estaremos evaluando y qué criterios utilizaremos.

CIERRE

A modo de cierre, comentamos a los alumnos que el código morse fue uno de los primeros usados para la comunicación a distancia. Es el que se usa en el telégrafo. Un telégrafo transmite escritura a distancia (del griego: *tele*, 'distancia' y *grafo*, 'escritura'), de modo semejante a como el teléfono transmite la voz a distancia (del griego: *tele* y *foné*, 'voz').

NOMBRE Y APELLIDO:

CURSO:

FECHA:

MENSAJES SECRETOS

Imaginá que sos guardaparque del Parque Nacional Quebrada del Condorito. Un día te avisan que un montañista que salió a la mañana todavía no ha regresado. Por suerte, llevaba una máquina para enviar mensajes de auxilio en clave, que emite señales mediante dos niveles de energía: alto y bajo. Recibís su mensaje codificado, ¡pero el problema es que no hay computadoras en el parque para interpretarlo!



1. Como esta cuestión es de vida o muerte, tenés que descifrar el mensaje. Mirá la tabla con el código que usa la máquina para codificar caracteres.

0	1	2	3	4	5	6	7	8	9	10	11	12	13
	A	B	C	D	E	F	G	H	I	J	K	L	M
14	15	16	17	18	19	20	21	22	23	24	25	26	27
N	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z

Aquí tradujimos el mensaje recibido: el nivel de energía alto está representado con el 1 y el nivel bajo, con el 0. Primero tenés que convertir esas secuencias en un número decimal y, luego, usar este número para descubrir las letras del mensaje codificado. ¿Qué dice el mensaje? ¿Dónde se perdió el montañista?

REPRESENTACIÓN BINARIA					REPRESENTACIÓN DECIMAL	MENSAJE
1	0	0	0	1		
0	0	1	0	1		
1	0	0	1	1		
0	0	1	0	0		
0	1	0	0	1		
0	0	1	0	0		
1	0	0	0	0		
0	0	0	0	0		
0	0	1	0	1		
0	1	1	1	0		
0	0	0	0	0		
0	0	1	0	1		
0	1	1	0	0		
0	0	0	0	0		
1	0	0	1	1		
0	1	0	0	1		
1	0	0	0	0		

NOMBRE Y APELLIDO:

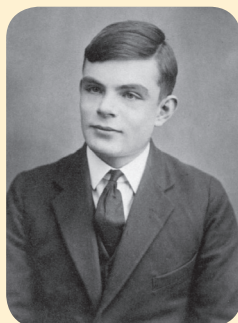
CURSO:

FECHA:

EVALUACIÓN

Estas son las tareas y la forma de calificar que tu docente considerará para evaluar cómo resolvés la actividad.

CALIFICACIÓN	ES NECESARIO TRABAJAR MÁS	BUENO-MUY BUENO	EXCELENTE
Planteo del problema	El estudiante no comprende cómo se resuelve el problema.	El estudiante describe el sistema para descifrar, pero con algunos errores.	El estudiante genera un sistema correcto para descifrar y escribir el código. Puede ser usando tablas o relacionando letras con números en una hoja.
Representación numérica	El estudiante no puede traducir números binarios en decimales ni decodificar letras.	El estudiante puede decodificar algunos números binarios, pero no todos.	El estudiante puede decodificar todos los números binarios del mensaje correctamente, incluso los espacios en blanco.
Apropiación del sistema binario	El estudiante no puede explicar en qué consiste un sistema binario.	El estudiante puede explicar en qué consiste un sistema binario.	El estudiante puede imaginar otras situaciones en que se usan codificaciones de dos valores como en el sistema binario.



DE PELÍCULA

¿Sabías que los nazis usaron mensajes secretos para comunicarse durante la Segunda Guerra Mundial? El matemático inglés Alan Turing creó una máquina capaz de tomar el código elaborado por los alemanes y escribirlo como una tira de ceros y unos. Fue un acontecimiento clave para el triunfo de los aliados en esa guerra.



Secuencia Didáctica 3

TRIVIA DE PREGUNTAS

Como cierre del capítulo, proponemos desarrollar un proyecto de programación. Los estudiantes comenzarán revisando conceptos vistos en los capítulos anteriores, tales como eventos, parámetros y condicionales. Luego, resolverán desafíos relacionados con la representación de datos aprendida en este capítulo. La propuesta es que, a lo largo de cinco actividades, puedan crear un juego de preguntas y respuestas.

.....

OBJETIVOS

- Presentar una forma de codificar colores.
- Aplicar los conceptos trabajados en un proyecto de programación.

.....

Actividad 1





¡Que gire la ruleta!

DE A DOS

OBJETIVOS

- Poner en práctica el manejo de eventos.
- Comunicar objetos mediante envíos de mensajes.

MATERIALES

-  Computadora
-  Scratch
-  Proyector (opcional)
-  Ficha para estudiantes

DESARROLLO

Con el fin de repasar los temas de programación y representación de datos que vimos hasta ahora, construiremos un juego de preguntas y respuestas: una versión simplificada del muy difundido Preguntados.



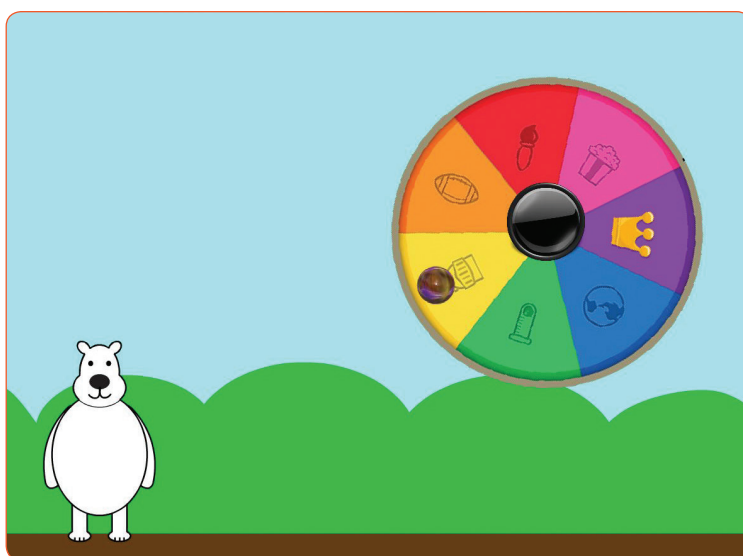
Portada del juego en línea Preguntados

Comenzamos la actividad preguntando a los estudiantes: “¿Conocen algún juego de preguntas y respuestas para el celular o la computadora?”. Dado el éxito de Preguntados, es muy probable que algunos sepan de él. Les contamos que vamos a trabajar en un proyecto en el que ellos mismos harán su propia versión del juego. Consultamos si saben cómo jugarlo. Para explicarles su funcionamiento, una posibilidad es jugarlo en clase o mostrar algún video con un proyector.

Si ninguna de las opciones es posible, les contamos en qué consiste. Es un juego de preguntas y respuestas en el que se utiliza una ruleta con colores que se hace girar. Cada color representa un área temática y hay muchas preguntas asociadas a cada una. Se trata de preguntas de opción múltiple, es decir, cada pregunta incluye una serie de posibles respuestas, de las cuales solo una es correcta. A modo de ejemplo, pensemos que el rojo podría representar la categoría *Deportes*, el verde *Literatura*, el amarillo, *Conceptos de programación*, etc. Para jugar, se hace girar la ruleta. Cuando frena, hay que responder una pregunta de la categoría correspondiente al color seleccionado. Una de las preguntas asociadas al color rojo podría ser: “¿Que jugador argentino hizo dos goles a Inglaterra en el partido que disputaron en el Mundial de Fútbol de México en 1986?”. Las opciones de respuesta pueden ser: A. Valdano, B. Borghi, C. Maradona y D. Francescoli. Si se selecciona la alternativa correcta, se suma un punto.

Antes de empezar a trabajar con la computadora, es necesario definir siete categorías de preguntas con los estudiantes. Les pedimos que las copien bajo la primera consigna de la ficha de la actividad.

A continuación, armamos la escena. Les contamos que necesitamos cuatro objetos fundamentales: la ruleta, un botón para hacerla girar, una bolilla y el personaje que formula las preguntas. Pedimos a los estudiantes que los elijan y también que seleccionen un escenario para usar de fondo.¹ Es importante que la bolilla quede ubicada sobre uno de los colores de la ruleta.



Propuesta de armado de escena con cuatro objetos: ruleta, botón, bolilla y personaje

Preguntamos a los estudiantes: “¿Cuándo tiene que girar la ruleta? ¿Cómo la hacen girar?”. Solicitamos que empiecen a programar sus ruletas para que giren cuando se presiona el botón. Damos un tiempo para realizar esta tarea y luego les consultamos si pudieron realizarla con éxito. Lo más probable es que hayan encontrado la instrucción `girar () grados`, que se encuentra en la categoría *Movimiento*. Como lo que queremos es que la ruleta empiece a girar cuando se presiona el botón, debemos recordarles en este momento la noción de evento. Un evento es un suceso que genera una reacción del programa. En este caso, queremos que la ruleta responda al evento *presionar el botón*. Motivamos la exploración del entorno de Scratch en busca de algún bloque que les permita resolver el desafío.

Luego de unos minutos, preguntamos a los estudiantes si pudieron encontrar algún bloque que les permitiera identificar un clic sobre el botón. Les pedimos que nos digan cómo se llama el bloque y en qué categoría se encuentra. Deberían reconocer que en la categoría *Eventos* se encuentra el bloque `al hacer clic en este objeto`. Es muy importante que, al programar el manejo del evento, se encuentre seleccionado el botón en la galería de *Objetos*.

.....

¹En caso de querer distribuir entre los estudiantes las imágenes que ilustran el manual, pueden descargarse del proyecto disponible en <https://goo.gl/ftbLpT>.

Les pedimos que terminen el programa para que la ruleta gire cuando se aprieta el botón. Es muy probable que, al tratar de resolver esta tarea, los estudiantes provoquen que gire el botón en lugar de la ruleta. Si eso sucediese, podemos aprovechar para reiterar que las instrucciones hay que dárselas al objeto que queremos que las lleve a cabo. Entonces preguntamos: “Si el manejo del evento lo tienen programado sobre el botón, ¿cómo pueden hacer para que sea la ruleta la que se mueva?”.

Para que el objeto ruleta gire, tenemos que enviar un mensaje dando aviso de que se ha presionado el botón. Una vez que lo reciba, la ruleta actuará en consecuencia. En la categoría *Eventos* se encuentra el bloque `enviar []`. Allí se puede escribir como argumento el mensaje *se presionó el botón de giro*. Con esto, cada vez que lo apretamos, el botón anunciará el evento que se produjo.

Solo resta indicar qué hará la ruleta al recibir el mensaje. Por ahora nos alcanza con indicarle que gire lo necesario para avanzar un color. Como son siete los colores, avanzamos un séptimo de una vuelta de 360°.

CIERRE

Para cerrar la actividad, hacemos hincapié en que, en este caso, el manejo de un evento disparó el envío de un mensaje y que, a su vez, el mensaje provocó una reacción en otro objeto.

al hacer clic en este objeto

girar 15 grados

Programa que hace girar al botón en lugar de la ruleta

al hacer clic en este objeto

enviar se presionó el botón de giro ▼

El botón envía un mensaje anunciando que se lo ha presionado

al recibir se presionó el botón de giro ▼

girar 360 / 7 grados

La ruleta gira al recibir el mensaje del botón

NOMBRE Y APELLIDO:

CURSO:

FECHA:

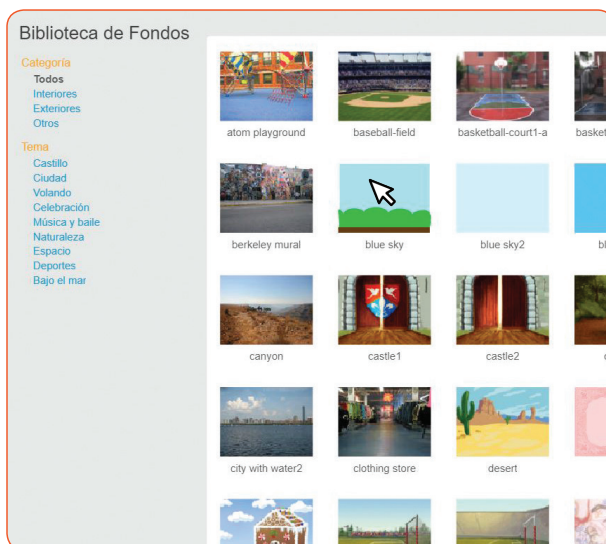
¡QUE GIRE LA RULETA!

¿Alguna vez jugaste al Preguntados? Se trata de un juego de preguntas y respuestas que usa una ruleta de colores. Se hace girar la ruleta y, cuando frena, hay que responder la pregunta que toca. ¡Empezá a armar tu propio juego de preguntas y respuestas en Scratch!



1. Tenemos que decidir las siete categorías donde se incluirán las preguntas y las respuestas. Escribilas acá abajo.

2. Elegí los objetos principales: la ruleta, el botón, una bolilla y el personaje que hará las preguntas. Agregá, además, un escenario para el fondo. ¿Qué fondo usaste? ¿Y qué personajes?



3. Hacé que la ruleta gire cuando se presione el botón. Por ahora, solo tenés que conseguir que pase de un color al siguiente.

Actividad 2

Gira la ruleta y... ¡no va más!



DE A DOS

OBJETIVO

- Ejercitar el uso de ciclos.

MATERIALES



Computadora



Scratch



Ficha para estudiantes

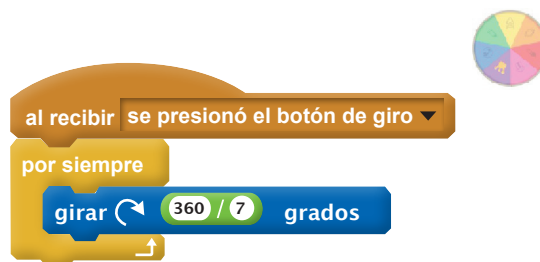
DESARROLLO

Comenzamos haciendo un repaso de la actividad “¡Que gire la ruleta!”. Recapitulamos el trabajo realizado diciendo: “Empezamos a desarrollar nuestro juego. Elegimos los personajes, el escenario y logramos que nuestra ruleta gire cada vez que presionamos el botón. ¿Cómo lo hicimos?”. Lo conseguimos manejando el evento que se produce al hacer clic sobre el botón.

Continuamos señalando que, de la misma manera que necesitamos un evento que identifique que hemos presionado un botón, necesitamos un mensaje que avise a la ruleta que empiece a girar. “Logramos que la ruleta gire, pero me parece que todavía falta algo. ¿Cómo hacemos para que la ruleta gire más tiempo y de manera continua?”. Pedimos a los estudiantes que exploren la herramienta. Luego de unos minutos, preguntamos si pudieron lograrlo.

En la categoría *Control* se encuentra el bloque **por siempre**, que permite repetir indefinidamente las acciones que nosotros queramos. Podemos arrastrar hasta allí todos los bloques que deseemos. Pedimos a los estudiantes que prueben y vean qué sucede al colocar allí un bloque de giro.

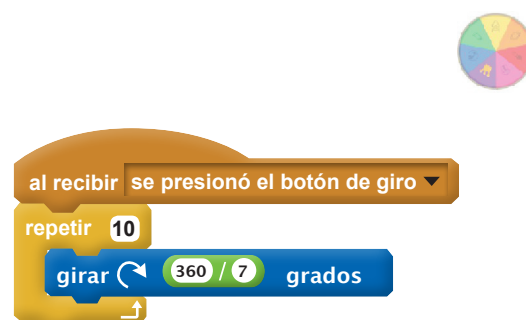
Preguntamos: “¿Qué sucede ahora?”. Muy probablemente algún estudiante responda que la ruleta gira continuamente y nunca se detiene. Esto es así porque ese es el comportamiento que programamos. Señalamos que, en general, el fragmento de programa que se encuentra dentro de un bloque **por siempre** se ejecuta indefinidamente.



La ruleta gira indefinidamente

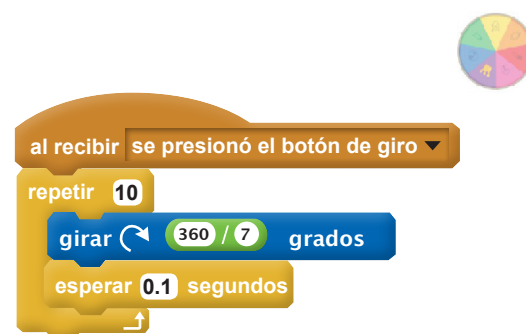
Precisamos que la ruleta se detenga luego de algún tiempo. “¿Se animan a buscar alguna instrucción que provoque que la ruleta gire pasando por los colores una cierta cantidad de veces y luego pare?”.

Luego de unos minutos, preguntamos a los estudiantes si pudieron lograr la tarea planteada. Si alguno lo logró y se anima a explicarlo a sus compañeros, le pedimos que lo haga. Alternativamente, lo podemos contar de la siguiente manera: “Necesitamos que nuestra ruleta gire cambiando de color varias veces. Precisamos un ciclo, pero no uno que funcione para siempre. Queremos que se repita una cantidad de veces y luego frene”. Dentro de la categoría *Control* encontramos el bloque `repetir ()`. A diferencia de `por siempre`, se ejecuta una cantidad finita de veces. Por defecto viene con el número diez como argumento, pero podemos cambiarlo a gusto. Combinando este bloque con `girar () grados` conseguimos que la ruleta gire varias veces antes de detenerse.



La ruleta gira 10 veces

Preguntamos a los estudiantes: “¿Qué pasa ahora cuando se presiona el botón de la ruleta?”. Guiamos a los estudiantes a que indiquen que la ruleta gira demasiado rápido. Los invitamos a explorar la herramienta para encontrar alguna forma de que la ruleta gire más despacio. Les damos un tiempo y luego les indicamos que, para que los giros sean más pausados, podemos usar el bloque `esperar () segundos` cada vez que la ruleta cambia de color. El valor usado como argumento define la duración de la pausa. En el ejemplo, una décima de segundo.



La ruleta gira haciendo una pausa entre color y color

CIERRE

Reflexionamos con los estudiantes sobre los distintos ciclos usados en la actividad. Mientras uno provoca la repetición indefinida de una serie de instrucciones, el otro solo lo hace un número finito de veces.

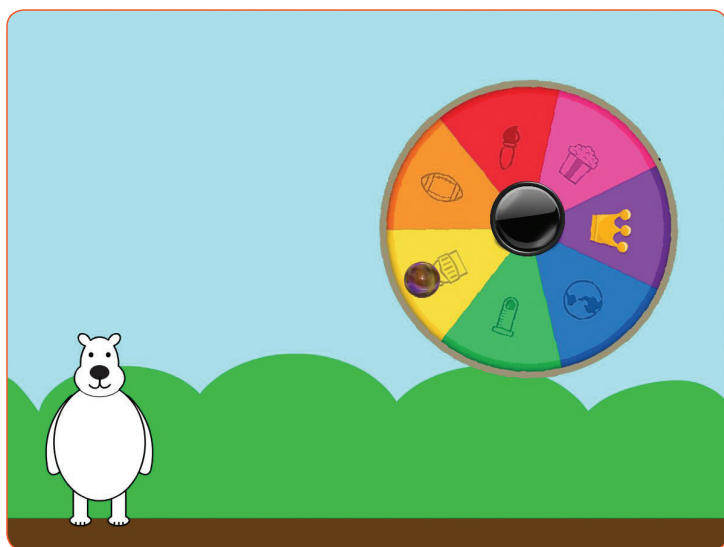
NOMBRE Y APELLIDO:

CURSO:

FECHA:

GIRA LA RULETA Y... ¡NO VA MÁS!

¡Seguimos desarrollando nuestro juego de preguntas y respuestas! Ya tenemos nuestro personaje, manejamos el evento del botón y mandamos un mensaje a la ruleta. Abrió Scratch y seguí las consignas.



1. Usá un bloque para que la ruleta gire sin parar. ¿Cómo lo hiciste?

2. ¿Cómo podemos hacer para que la ruleta gire algunas veces y luego pare?
Investigalo en Scratch.

3. ¡La ruleta gira tan rápido que marea! Buscá una manera de que la ruleta vaya más lento para disfrutar de su movimiento. ¿Qué bloque usaste?

Actividad 3

Agregamos un poco de azar



DE A DOS

OBJETIVOS

- Introducir la noción de aleatoriedad.
- Enviar mensajes para dar a conocer el estado de un programa.

MATERIALES



Computadora



Scratch



Proyector (opcional)



Ficha para estudiantes

DESARROLLO

En general, cuando giramos una ruleta no sabemos de antemano cuál será el resultado. Este depende de factores tales como la fuerza a la que se la hace girar, el instante en el que se lanza la bola, etc. A diferencia de una ruleta normal, la ruleta que los estudiantes programaron hasta el momento siempre rota sobre su eje la misma cantidad de grados. Es decir que, antes de presionar el botón, podemos calcular cuál será el siguiente color seleccionado. En la primera parte de esta actividad usaremos **números aleatorios** para producir un comportamiento más parecido al de las ruletas reales. Luego, nos ocuparemos de notificar cuándo se detiene la ruleta, lo que significa que un nuevo color ha sido seleccionado.

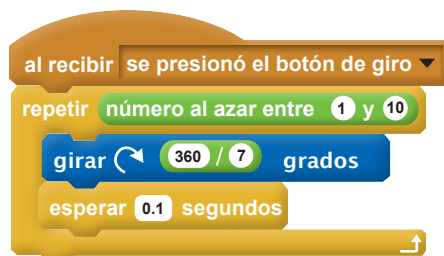
Podemos comenzar la actividad proyectando un video que muestre cómo gira una ruleta. Si no se cuenta con un proyector, se puede pedir a los estudiantes que lo observen ejecutando la aplicación Preguntados en sus dispositivos móviles. “¿En qué se diferencia esta ruleta de la que estamos desarrollando?”. Guiamos la discusión de forma tal de concluir que, a diferencia de lo que sucede en el juego original, en el nuestro la ruleta gira siempre la misma cantidad de grados.

Preguntamos: “¿Cómo podemos modificar el programa para que nuestra ruleta no gire siempre la misma cantidad de veces cada vez que presionamos el botón?”. Damos unos minutos para que los estudiantes exploren Scratch y luego preguntamos si lograron resolver el desafío. Puede ser que alguien proponga cambiar el valor del argumento del bloque `repetir ()`. Compartimos con los demás estudiantes esta propuesta y preguntamos: “¿Conseguimos de este modo que cada vez que presionemos el botón la ruleta varíe la cantidad de veces que se produce un cambio de color?”. No, lo único que logramos fue que la cantidad de veces sea otra, pero sigue siendo fija. Siguiendo esta lógica, deberíamos cambiar el argumento cada vez que apretamos el botón. Y aun así, en cada oportunidad sabríamos de antemano cuál sería el próximo color seleccionado.

Preguntamos a la clase “¿Conocen algún juego de mesa en el que intervenga el azar?”. Luego, proponemos un ejemplo: “Piensen qué pasa cuando jugamos con un dado. Cuando lo tiramos, ¿siempre se repite el mismo número o va cambiando? ¿Qué números les pueden tocar al dar vuelta el cubilete? ¿Cuál es el menor número que puede salir? ¿Y el mayor?”.

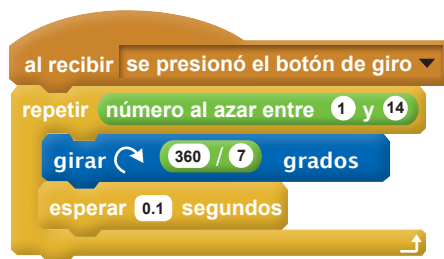
Cuando tiramos un dado, nunca sabemos con exactitud qué saldrá. Pero sí sabemos que los números posibles están entre 1 y 6. Les pedimos a los estudiantes que exploren el entorno de Scratch y busquen un bloque que les permita reproducir lo que sucede al tirar un dado. Es esperable que descubran la instrucción `número al azar entre () y ()` de la categoría *Operadores*, cuyos parámetros permiten indicar los límites inferior y superior del rango del cual se seleccionará un número entero al azar.

Ahora, en lugar de elegir un número específico de repeticiones, podemos generar un número aleatorio entre un mínimo y un máximo. Proponemos a los estudiantes que utilicen este bloque en sus producciones. Es esperable que arriben a una solución que use números al azar entre 1 y 10, que son los valores por defecto para los argumentos.



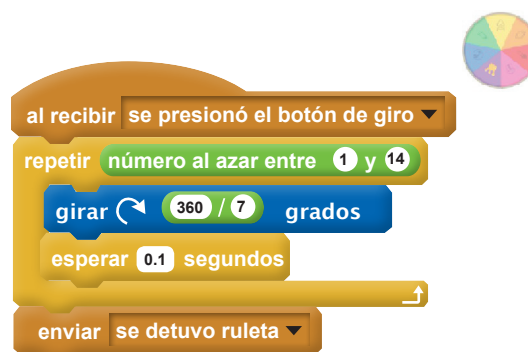
La ruleta gira una cantidad de veces al azar

Si bien la solución es correcta, al hacer girar la ruleta no todos los colores tendrán la misma probabilidad de ser seleccionados. Esto se debe a que tenemos 7 colores diferentes y estamos eligiendo números entre 1 y 10. Por ejemplo, si se selecciona el número 1, el color resultante será el mismo que si se selecciona el 8. Lo mismo sucede con el 2 y el 9 y con el 3 y el 10. Sin embargo, hay solo un valor que hará que se seleccione cada uno del resto de los colores. Para que la elección sea equiprobable hay que usar como segundo argumento un valor que sea múltiplo de 7; por ejemplo, 14.



Todos los colores tienen la misma probabilidad de ser seleccionados

Preguntamos a la clase: “¿Cuándo queda efectivamente seleccionado un nuevo color?”. Una vez que la ruleta frena. “¿Se les ocurre cómo indicar que la ruleta se ha detenido?”. Es un buen momento para recuperar la noción de comunicación entre objetos mediante el envío de mensajes. En este caso, enviaremos el mensaje *se detuvo la ruleta*, que será interceptado por la bolilla para reconocer el nuevo color. De esto último nos ocuparemos en la próxima actividad.



La ruleta avisa que se ha detenido

CIERRE

Como cierre, comentamos que el uso de números al azar es muy común en el universo de la computación. Muchas veces se tiene que simular un proceso con un final incierto, para lo cual este tipo de números es muy útil. Por último, podemos comentar que muchos teóricos de la computación se interesan y estudian los fenómenos aleatorios. Este tema es, sin dudas, un nexo con otras disciplinas como, por ejemplo, la Física.

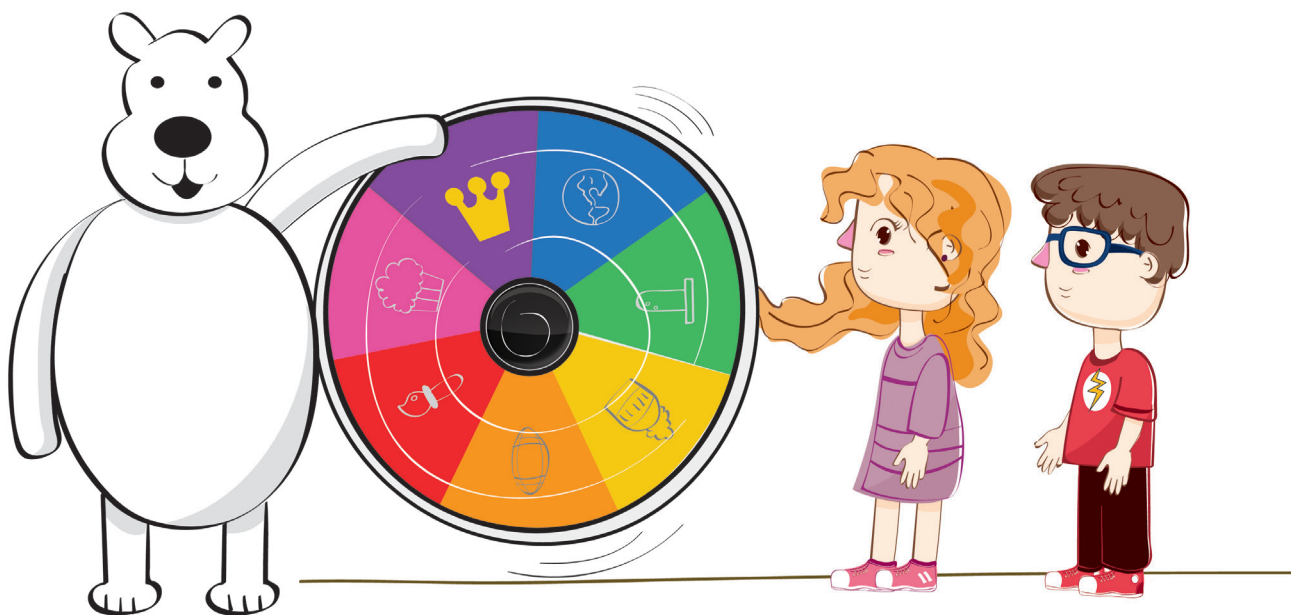
NOMBRE Y APELLIDO:

CURSO:

FECHA:

AGREGAMOS UN POCO DE AZAR

Las ruletas se usan en juegos de azar, y nuestra trivia no es una excepción. ¡Vamos a transformar nuestro juego en un auténtico juego de azar!



1. ¿Cómo hacemos para que la ruleta no gire siempre la misma cantidad de veces? Explorá Scracht y buscá cómo hacer para que la ruleta gire una cantidad de veces diferente cada vez que presionamos el botón. ¿Qué bloque usaste?

2. Ahora, programá la ruleta para que envíe un mensaje cada vez que se detiene. Este mensaje, más adelante, será interceptado por la bolilla para avisar qué color salió. ¿Qué mensaje enviaste?

Actividad 4




Identificamos el color elegido

 DE A DOS

OBJETIVOS

- Presentar la codificación de colores RGB.
- Ejercitar el uso de mensajes para comunicar objetos.

MATERIALES

-  Computadora
-  Scratch
-  Ficha para estudiantes

DESARROLLO

Iniciamos esta actividad preguntando: “¿Alguna vez mezclaron témperas para crear nuevos colores? El rojo, el amarillo y el azul son los colores primarios; si los mezclamos, podemos formar otros colores. ¿Qué pasa si mezclamos rojo y azul? ¿Y rojo con amarillo?”.

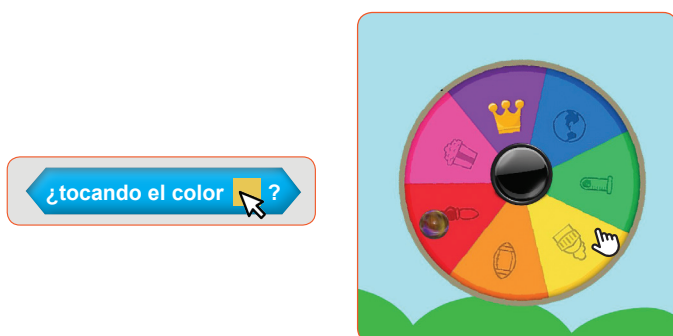
Explicamos que los colores que vemos en las pantallas también se basan en la mezcla de tres colores. El punto de partida es una pantalla negra que va adquiriendo color a través de luces. Cada píxel (menor unidad homogénea en color que forma parte de una imagen digital) se compone de tres luces minúsculas: una roja, una verde y una azul. A este conjunto de colores se los llama RGB por sus siglas en inglés: *Red* (rojo), *Green* (verde) y *Blue* (azul). Al aumentar y disminuir la cantidad de luz que emite cada una de estas luces, se puede formar un colosal número de colores. Estas mezclas se denominan *aditivas*.

¿Qué tiene que ver todo esto con nuestro juego de preguntas? Recapitulamos la actividad anterior y señalamos que, una vez que la ruleta se detiene, la bolilla queda apoyada sobre un color específico. Ahora tenemos que incorporar una nueva característica a nuestro programa. La bolilla tendrá que indicar el color seleccionado mandando un mensaje, que posteriormente usará el personaje animador para hacer la siguiente pregunta. En el camino, aprenderemos la codificación de colores RGB.



Algunos bloques de la categoría *Sensores* son útiles para explorar la relación de los objetos con su entorno. Hay uno que, en este caso, será de gran ayuda: `¿tocando el color []?`. Al usarlo debemos indicar un color como argumento.

Este bloque representa la expresión *el objeto está en contacto con alguna superficie del color*. La condición será cierta en algunas oportunidades y falsa en otras. El camino más sencillo para seleccionar un color como argumento consiste en hacer clic sobre el cuadrado de del bloque `¿tocando el color []?`, lo que transforma el puntero del ratón en una pequeña mano con el dedo índice extendido. Entonces, al seleccionar un punto de la pantalla, el color del punto será incorporado como argumento del bloque.



Selección de un color para `¿tocando el color []?`

Sin embargo, nosotros vamos a usar esta actividad como excusa para presentar la codificación RGB, con el propósito de conocer una forma de representar colores. RGB usa números en el rango de 0 a 255 para indicar la intensidad de cada una de las tres luces que componen un píxel, con lo que usaremos 8 bits para cada color (con 8 dígitos binarios pueden representarse 256 valores distintos). A los grupos de 8 bits se los conoce como **bytes**. Si el byte correspondiente a la luz roja es 0, significa que está completamente apagada. Si es 255, completamente encendida. En el caso de nuestro juego, el verde de la ruleta está representado en RGB por los números (36, 201, 104); el amarillo por (250, 221, 65); el naranja por (250, 148, 65); el rojo por (243, 43, 54); el rosado por (243, 77, 175); el lila por (142, 69, 184) y el azul por (44, 120, 199). Estos valores son adecuados si se están usando las imágenes provistas en este manual. Si no fuera así, deben adaptarse a los colores de la imagen usada para la ruleta.

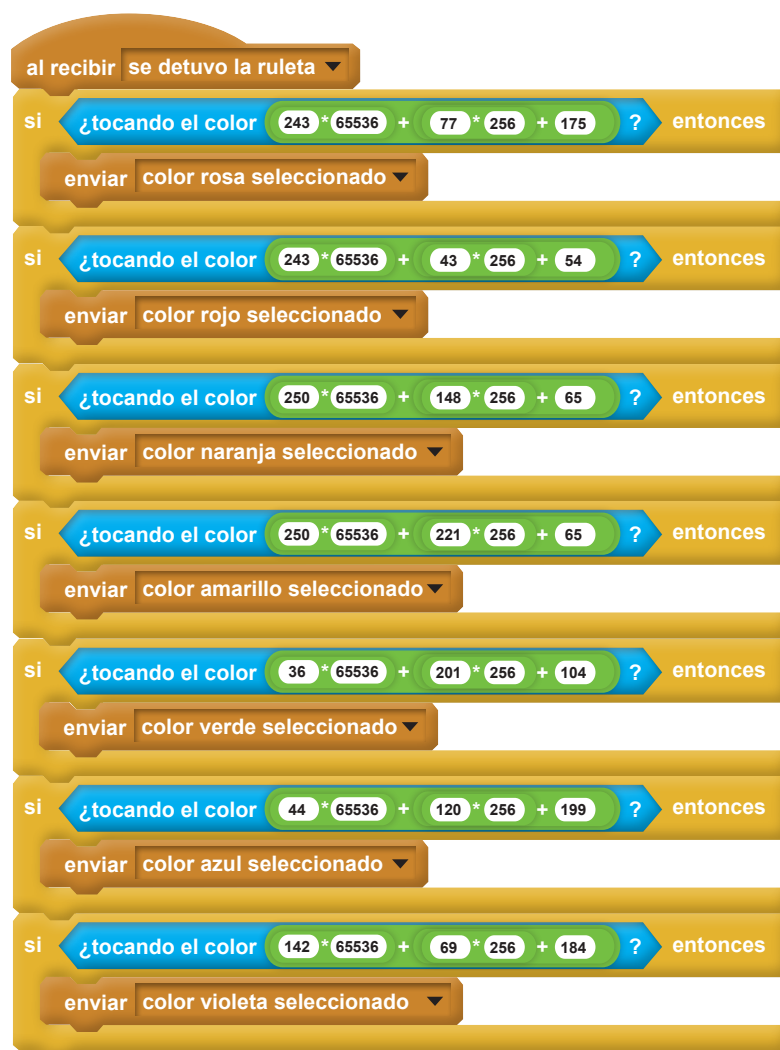
El bloque `¿tocando el color []?` admite como argumento un número, cuyo valor se usa para representar un color en particular. Los 3 bytes utilizados para codificar un color en RGB pueden escribirse como un solo número de la siguiente manera: para indicar a la computadora que el rojo se corresponde con el byte a la izquierda de los tres, debemos multiplicar el valor de rojo por 2^{16} . De esta manera, indicamos que hay otros 16 bits a su derecha que corresponden a los bytes del verde y del azul. 2^{16} es 65.536. De forma similar indicaremos que el verde es el byte del medio y diremos que tiene 8 bits a su derecha, reservados para la componente azul. Para esto debemos multiplicar el valor del verde por 2^8 , que da 256. Finalmente, el byte del azul no lo tenemos que multiplicar por ningún factor, pues es el que está a la derecha. Para terminar sumaremos el rojo, más el verde, más el azul. Por ejemplo, para representar el naranja cuya codificación RGB es (250, 148, 65), usamos el número que resulta de la cuenta $250 \times 65536 + 148 \times 256 + 65$. No hace falta que lo calculemos, pues la computadora lo hará por nosotros.¹

`¿tocando el color` `250 * 65536 + 148 * 256 + 65` `?`

.....

¹ Al igual que en la mayoría de los lenguajes de programación, el símbolo utilizado para identificar la operación de multiplicación es * y no ×.

Una vez que construimos la expresión numérica de cada color, podemos incorporar a nuestro programa el mensaje que envía la bolilla cuando la ruleta se detiene. Por el modo en que diseñamos el programa, la bolilla siempre termina tocando un único color. Una posible solución se exhibe a continuación.



Porción del programa que produce que la bolilla avise el color seleccionado una vez que se detiene la ruleta

CIERRE

Comentamos con los estudiantes que, esencialmente, una codificación consiste en utilizar una cosa para denotar otra. En esta actividad, usamos tríos de números para representar colores.

NOMBRE Y APELLIDO:

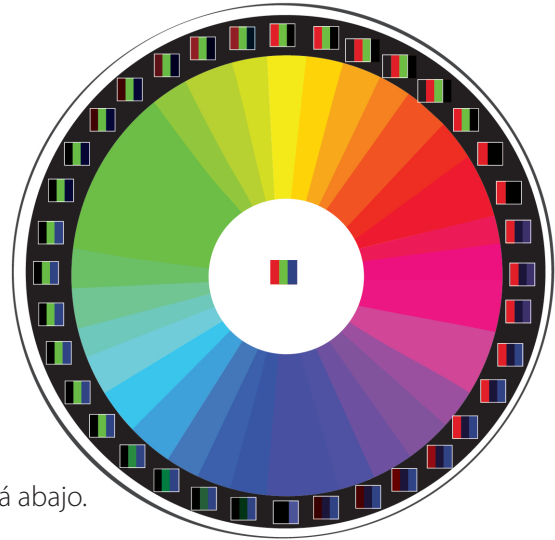
CURSO:

FECHA:

IDENTIFICAMOS EL COLOR ELEGIDO

¡Hoy vamos a jugar con la representación de colores! Mirá la rueda de colores con los píxeles RGB.

Montones de colores se consiguen regulando la intensidad de las luces roja, azul y verde combinadas. Por ejemplo, un color naranja fuerte se representará como (222, 102, 36). Es decir, se requiere mucho rojo, bastante verde y algo de azul.



1. Usando cualquier programa de edición de imágenes¹ conseguí las representaciones en RGB de los 7 colores de la ruleta de nuestra trivia de preguntas y copialos acá abajo.

2. Formulá, con ayuda de tu docente, cada código RGB de los colores de la ruleta con un único número. ¿Qué cuenta tuviste que hacer?

3. Buscá en Scratch los bloques que hacen falta para comparar un número que refleja un código RGB y el color que está tocando la bolilla. ¿Qué bloques usaste? ¿En qué categoría están?

4. Según el color detectado, enviá un mensaje que indique el color seleccionado. ¿Qué dice tu mensaje?

CÓMO SE FORMAN LOS COLORES EN RGB

Quando la luz roja está completamente prendida y las demás están apagadas se representa como (255, 0, 0). De forma similar, el verde se representa con (0, 255, 0) y el azul con (0, 0, 255). El amarillo se consigue prendiendo completamente las luces roja y verde (255, 255, 0), el fucsia con las luces roja y azul (255, 0, 255) y el cian con las luces verde y azul (0, 255, 255). Si apagamos todas las luces (0, 0, 0), tenemos oscuridad total, que es el negro. Y si prendemos todas las luces tenemos saturación de luz, que da blanco (255, 255, 255).

¹ Se puede usar, por ejemplo, la que se encuentra disponible en <https://goo.gl/JKqJNb>.

Actividad 5

Y finalmente preguntamos



DE A DOS

OBJETIVO

- Construir un programa que reciba entradas del usuario.

MATERIALES



Computadora



Scratch



Ficha para estudiantes

DESARROLLO

Iniciamos la clase preguntando a los estudiantes: “¿Cómo podemos usar los mensajes que envía la bolilla para que el personaje haga las preguntas que queremos?”. Enfatizamos que el personaje conductor tiene que hacer las preguntas de la categoría asociada al color en el cual cae la bolilla. Repartimos la ficha y pedimos a los estudiantes que resuelvan la primera consigna.

Luego de unos minutos, consultamos si lograron resolver el desafío. Es muy probable que no hayan podido, pues implica incorporar el uso de un nuevo bloque.

Para resolver el desafío hace falta usar una instrucción que permita que el personaje principal haga las preguntas. En la categoría *Apariencia* podemos encontrar dos bloques similares. Ambos se utilizan para que los objetos “hablen” como en las historietas. El bloque `decir [] por () segundos` tiene dos parámetros. El primero indica el texto que dirá el personaje y el segundo durante cuánto tiempo hablará. Por su lado, `decir []` recibe solo el texto. ¿Por cuánto tiempo se mostrará el diálogo en este caso? Les pedimos a los estudiantes que prueben los bloques para descubrir que, en este caso, el globito con el texto no desaparece más de la pantalla.

decir Hola por 2 segundos

decir Hola

Bloques para que los personajes "hablen"

Como queremos que el personaje sea quien haga las preguntas, debemos programarlo para que realice esa tarea. Recordemos la actividad del principio de este proyecto sobre el armado de las categorías con sus respectivas preguntas y respuestas. Les pedimos a los estudiantes que las recuperen y las usen adecuadamente para que, cuando el personaje principal reciba un mensaje que indica que un nuevo color ha sido seleccionado, haga una pregunta.

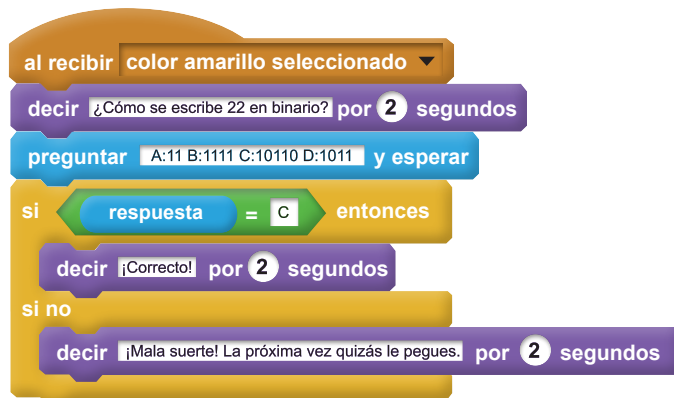
Una vez presentada la pregunta, el personaje principal debe dar opciones y aguardar la respuesta del jugador. El bloque `preguntar [] y esperar` sirve para este propósito. Este bloque muestra el texto provisto como argumento y bloquea la ejecución del programa hasta que el usuario ingrese un texto y presione `return`. Una vez que esto sucede, el valor ingresado queda almacenado en el bloque `respuesta`.

preguntar ¿Cuál es tu nombre? y esperar

respuesta

Bloques para hacer una pregunta y recuperar la respuesta

A continuación se muestra un ejemplo en el que asociamos el color amarillo a una categoría hipotética de preguntas que llamamos *Representación de datos*.



El personaje pregunta y aguarda una respuesta

Los estudiantes deben entregar su juego terminado. Se sugiere hacer un intercambio de juegos entre los grupos. Se plantean dos tipos de evaluaciones: evaluación entre pares y evaluación del docente junto con el grupo.

RÚBRICAS

Al final de la ficha de los estudiantes, incluimos una rúbrica para que los alumnos sepan qué estaremos evaluando y qué criterios utilizaremos.

CIERRE

Una vez terminada la actividad, hemos completado una versión del juego Preguntados. Remarcar que en esta última actividad escribieron por primera vez un programa que espera el ingreso de información por parte del usuario.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

Y FINALMENTE PREGUNTAMOS



Estamos muy cerca de terminar el juego.
Los desafíos que quedan por resolver son los siguientes.

1. ¿Cómo podemos usar los mensajes que envía la bolilla para que el personaje principal pregunte lo que queremos? ¡A programar!

2. El último desafío es mostrar el juego a un grupo de compañeros o a tu docente para probarlo. Tu docente o tus compañeros tienen que completar las frases que vas a encontrar a continuación. Cada grupo debe escribir una breve valoración del juego de otro grupo completando las siguientes oraciones:

a. Lo que más me gustó del juego fue:

b. Lo que más me sorprendió del juego fue:

c. Le recomiendo al grupo que mejore su juego haciendo las siguientes tareas:

NOMBRE Y APELLIDO:

CURSO:

FECHA:

EVALUACIÓN

Estas son las tareas y la forma de calificar que tu docente considerará para evaluar cómo resolvés las actividades..

CALIFICACIÓN	ES NECESARIO TRABAJAR MÁS	BUENO-MUY BUENO	EXCELENTE
Conceptos	Los estudiantes no han logrado usar las nociones de ciclo, parámetro, evento, etc. para desarrollar el juego.	Los estudiantes han logrado usar, con algunos errores, las nociones de ciclo, parámetro, evento, etc. para desarrollar el juego.	Los estudiantes han logrado usar las nociones de ciclo, parámetro, evento, etc. para desarrollar el juego.
Tareas realizadas	Los estudiantes no revisaron su juego.	Los estudiantes han revisado y mejorado su juego pocas veces o no han podido resolver los problemas observados.	Los estudiantes han revisado y mejorado su juego varias veces y han logrado proponer soluciones que pueden servir para otras situaciones.
Actitud hacia el aprendizaje	Los estudiantes no trabajaron en grupo ni pudieron crear algo propio.	Los estudiantes trabajaron en grupo, pero ante alguna dificultad no pudieron resolverla.	Los estudiantes trabajaron en grupo, se hicieron preguntas, compartieron sus ideas y pudieron crear algo propio.

06

LA COMPUTADORA

SECUENCIA DIDÁCTICA 1

LES QUITAMOS EL VELO A LAS
COMPUTADORAS

¿Qué sabemos sobre las computadoras?

Computadoras por todos lados

Las partes de toda computadora

Una nueva máquina

¡A conocer la computadora por dentro!

Los capítulos anteriores trataban acerca del *software*, es decir, los programas que una computadora ejecuta al realizar tareas. Este capítulo se centra en el **hardware** o, lo que es lo mismo, los componentes físicos que forman las computadoras. Sin el *hardware*, el *software* no existiría.

En este capítulo se proponen actividades que acercarán a los estudiantes a una nueva idea de computadora y les permitirán descubrir sus elementos tangibles.



Secuencia Didáctica 1

LES QUITAMOS EL VELO A LAS COMPUTADORAS

Las computadoras están muy presentes en nuestra vida cotidiana: controlan una gran variedad de artefactos, tanto en la industria como en el hogar. La mayoría de nosotros tiene una idea sobre qué es una computadora.

Aunque este objeto suele asociarse con dispositivos de escritorio y portátiles, en la actualidad las computadoras son una parte fundamental de todo tipo de aparatos: desde teléfonos celulares hasta televisores, pasando por automóviles, semáforos, cámaras fotográficas, de video, etc. Casi cualquier máquina automática que realice una tarea compleja está montada sobre una computadora.

Entonces, ¿qué es una computadora? Para responder esta pregunta, se propone una serie de actividades con un doble objetivo: por un lado, aproximarnos a una noción de computadora y, por el otro, pensar las computadoras como el corazón de la tecnología moderna.

.....

OBJETIVOS

- Arribar a una nueva idea de computadora.
- Conocer las partes fundamentales de una computadora.
- Reconocer la importancia de las computadoras en la vida cotidiana.

.....

Actividad 1




¿Qué sabemos sobre las computadoras?

GRUPAL (3)

OBJETIVOS

- Analizar diferentes tipos de computadoras.
- Conceptualizar la idea de computadora.

MATERIALES

-  Computadora
-  Conexión a Internet
-  Ficha para estudiantes

DESARROLLO

En la vida cotidiana, usamos la palabra *computadora* para referirnos a los dispositivos de escritorio o portátiles. Esta idea se originó en la década del ochenta, debido en gran medida a la masificación de las computadoras de uso personal. Sin embargo, deja afuera muchísimos dispositivos computacionales con los que interactuamos diariamente.

En esta actividad analizaremos distintas dimensiones de las computadoras: sus momentos de vigencia, la tecnología involucrada, los tamaños y su modo de interacción con los humanos. Veremos que, al considerar cada una de estas características, siempre podemos encontrar ejemplos que difieren de nuestra representación habitual de lo que es una computadora.



Comenzamos la actividad con una discusión grupal para averiguar qué saben los estudiantes sobre computadoras. “¿Qué computadoras conocen?”. Se espera que mencionen las computadoras de escritorio y las portátiles. En caso de que no surjan otros ejemplos, les explicamos que también un celular, una *tablet* y un televisor moderno tienen computadoras en su interior.

Para comenzar a desarmar el preconceito de computadora, podemos preguntarles a los estudiantes: “¿Qué diferencia hay entre un teléfono celular y una *tablet*?”. Guiamos la discusión de forma tal de llegar a estas conclusiones: por un lado, tienen distinta forma (las *tablets* suelen ser más grandes); por otro, tienen funciones distintas (por ejemplo, con uno podemos hacer llamadas y con la otra, no).

Continuamos preguntando: “¿Qué tienen en común los dispositivos mencionados?”. Todos ellos reciben información, la procesan y generan un resultado con el

que luego hacen algo. Por ejemplo, un televisor recibe una señal digital de cable, la decodifica y reproduce imágenes que muestra por la pantalla y sonidos que emite por los parlantes. Un teléfono celular recibe una señal con datos que viaja por el aire, la decodifica, identifica de qué se trata y, por ejemplo, nos avisa que llegó un nuevo mensaje de alguna red de mensajería instantánea.

A partir de los ejemplos mencionados, proponemos a los estudiantes que piensen algunas diferencias con relación a cómo interactuamos con estos dispositivos. Podemos formular preguntas tales como: “¿Todos tienen pantalla táctil? ¿Con cuáles usamos un ratón?”. Es esperable que durante la discusión se indique, por ejemplo, que algunos tienen parlantes y otros no, o que solo algunos tienen teclado. Si no se mencionó, se puede comentar que un *router* hogareño wifi es una computadora que no tiene ni teclado ni pantalla.

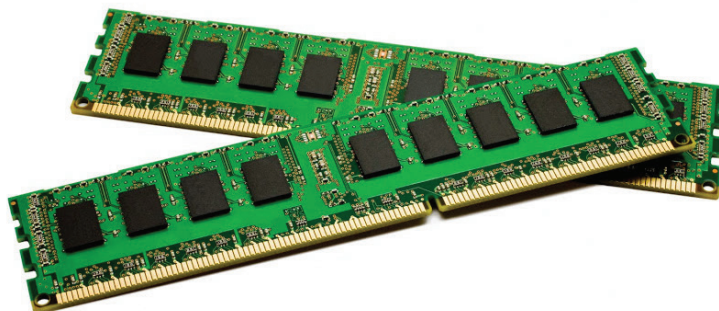
Explicamos a los estudiantes que existen componentes que no siempre están presentes en una computadora. Cada uno cumple una función específica. Por ejemplo, los parlantes emiten sonidos; el teclado, cada vez que se presiona una tecla, envía una señal que le permite a la computadora reconocer la tecla oprimida. A los componentes de este tipo se los denomina genéricamente *periféricos*.

En este punto contamos con todo lo necesario para elaborar con los alumnos una primera noción de computadora. Una posible definición surge pensando en lo que una computadora hace. Discutimos con la clase a partir de la siguiente pregunta: “¿Qué hace una computadora?”. Como se dijo antes, una computadora recibe información, la procesa y produce nueva información.

¿QUÉ HACE UNA COMPUTADORA?

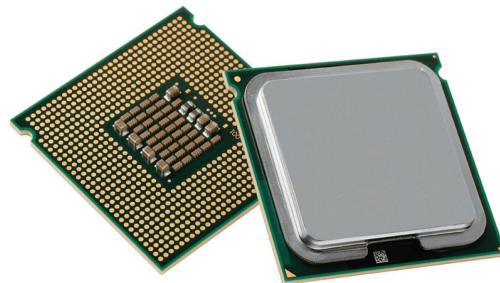
Una computadora recibe información, la procesa y produce nueva información.

Continuamos preguntando: “¿Qué tiene, en líneas generales, toda computadora?”. En principio, tiene al menos una **entrada** por la que obtiene la información y una **salida** por la que expresa un resultado. Además, las computadoras tienen una **memoria**, que es el componente físico donde se almacena la información.



Módulos de memoria de una computadora

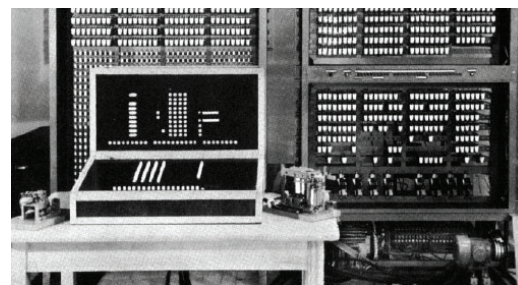
Finalmente, una computadora tiene un núcleo principal, que habitualmente se conoce como **unidad central de procesamiento** o CPU por sus siglas en inglés (*Central Processing Unit*). Se trata de un grupo de circuitos electrónicos integrados que constituyen el soporte físico para que todo el *software* pueda realmente ejecutarse¹, una instrucción tras otra.



Unidad central de procesamiento

Repartimos las fichas a los estudiantes y los alentamos a que busquen información en Internet que les permita resolver las consignas. La actividad propone buscar computadoras de acuerdo a distintas características, como su tamaño o la época en que existieron. A continuación, a modo de referencia, se reproduce la historia de un grupo de computadoras que permiten completar las tablas de la actividad.

La primera consigna propone la búsqueda de información sobre tres computadoras anteriores a 1960. La computadora Z3 se creó en Alemania en 1941 y fue la primera máquina programable completamente automática.



Computadora Z3, 1941

La Mark 1, de 1944, fue la primera computadora electromecánica. Construida por IBM en Estados Unidos, en 1947 se instaló en Harvard. En el mismo país, en 1946, se construyó la famosa computadora ENIAC, acrónimo de *Electronic Numerical Integrator And Computer* (Computador e Integrador Numérico Electrónico, en español). Fue una de las primeras computadoras que se denominan *de propósito general*. Inicialmente ENIAC se diseñó para calcular la trayectoria de proyectiles y realizar operaciones matemáticas para el ejército de Estados Unidos.



Computadora Mark 1, 1944

¹ Las características de la memoria y la unidad central de procesamiento se abordan en el próximo capítulo.

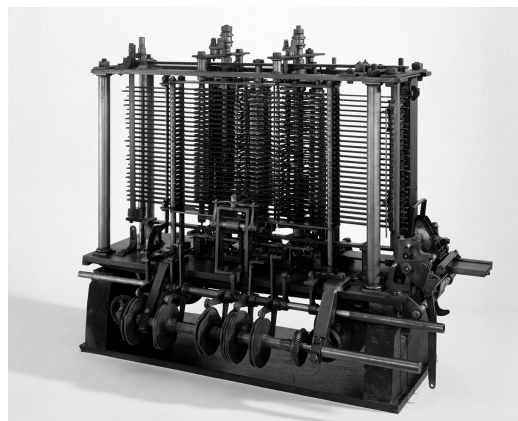
LAS PROGRAMADORAS DE ENIAC

La máquina fue diseñada por dos ingenieros, John Presper Eckert y John William Mauchly, pero seis mujeres se ocuparon de programarla: Betty Snyder Holberton, Jean Jennings Bartik, Ruth Lichterman Teitelbaum, Kathleen McNulty Mauchly Antonelli, Frances Bilas Spence y Marlyn Wescoff Meltzer.



El gobierno de los Estados Unidos contrató a este grupo de matemáticas para escribir los programas en sistema binario y realizar las diferentes operaciones que, en ese entonces, implicaban conectar y desconectar cables. Ellas no solo lograron el cometido, sino que modificaron la evolución de la programación entre los años cuarenta y cincuenta. Recién en 1997 se reconocieron sus méritos y se las incluyó en el Salón de la Fama de *Women in Technology International*. Además, en 2014 se realizó el documental *The Computers*, basado en su historia.

La segunda consigna plantea encontrar dos computadoras de dos siglos distintos. Por ejemplo, la máquina analítica diseñada por el matemático británico Charles Babbage entre 1833 y 1842. Babbage intentó construir la máquina, pero no logró completarla. Se trataba de una computadora mecánica programable. La entrada se imprimía sobre tarjetas perforables, la salida se producía usando un mecanismo que combinaba un equipo de dibujo y una campana, y el funcionamiento se montaba sobre una estructura similar a un telar. Años más tarde, la matemática y escritora Augusta Ada King escribió varios programas para la máquina diseñada por Babbage. Augusta era británica e hija del poeta Lord Byron y de la matemática y poeta Anne Isabella Noel Byron. Se la considera la primera programadora de la historia y se la conoce como Ada Lovelace.



Máquina analítica de Babbage

Otro posible ejemplar para completar la consigna es la computadora electromecánica Z1. Fue diseñada entre 1935 y 1938, en Alemania. Se proponía como solución al problema de tener que hacer muchas veces cálculos complejos muy similares.



Computadora electromecánica Z1, 1935-1938

Basaba su funcionamiento en el uso de relés eléctricos. Sin embargo, nunca llegó a funcionar de forma adecuada. Se cree que se destruyó durante un bombardeo aliado sobre Berlín, en 1943, dos años antes de la finalización de la Segunda Guerra Mundial.

La tercera consigna plantea encontrar tres computadoras cuyos mecanismos de entrada y salida no estén dados por un teclado alfanumérico y una pantalla. Se les puede sugerir a los estudiantes que piensen en artefactos que les resulten familiares y averigüen si son computadoras o las contienen. Un ejemplo son los *routers* wifi, dispositivos que se usan para brindar acceso a una red informática en forma inalámbrica. No suelen incluir teclado ni pantalla. Por otro lado, un sintetizador musical es una computadora cuya entrada son las teclas de un piano y su salida es un parlante. Por último, los autos modernos vienen equipados con una computadora que oficia de “cerebro” del automóvil; valiéndose de información proveniente de sensores, controla aspectos tales como la inyección de nafta en el motor y la climatización del interior del coche. En algunos casos, además, almacena las preferencias de cada conductor que usa el vehículo, lo que permite acomodar en forma automática la posición de los espejos retrovisores, la altura del asiento, etc.



La cuarta y última consigna pide encontrar computadoras actuales de distintos tamaños. Para resolverla, se puede recurrir a dispositivos tales como reproductores de música, teléfonos celulares, computadoras portátiles y de escritorio.

CIERRE

Como cierre de la actividad pasamos en limpio lo trabajado: una computadora es un dispositivo que recibe información, la procesa y genera nueva información; usa una memoria para almacenar los datos y una unidad central de procesamiento para realizar cálculos. Muchos artefactos que forman parte de nuestra vida cotidiana son computadoras, a pesar de que no solemos pensar en ellos como tales.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

¿QUÉ SABEMOS SOBRE LAS COMPUTADORAS?

Sabemos que existen las portátiles y las de escritorio, pero el mundo de las computadoras es mucho más amplio ¡Conocélo!

Buscá información en la web para completar las tablas que se muestran a continuación. Además, podés imprimir y pegar imágenes, y luego compartirlas con tus compañeros.



1. Vamos a viajar por el tiempo. Buscá tres computadoras anteriores a 1960.

AÑO	NOMBRE DE LA COMPUTADORA	IMAGEN

CLEMENTINA

¿Sabías que en 1961 llegó a nuestro país la primera computadora con propósito científico? Se trataba de una Ferranti Mercury a la que se apodó Clementina. Esta computadora funcionaba a válvulas y medía 18 m de largo. La entrada de datos y programas se hacía a través de cintas de papel perforadas. ¡La memoria era de solo 1K!

Hoy, estas características nos parecen obsoletas, pero en esa época Clementina fue el sostén de un proyecto de vanguardia dedicado a la investigación, la docencia y la oferta de servicios que fue pionero en nuestro país y la región.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

- 2.** Encontrá y compará dos computadoras de distintos siglos. ¡Sí, siglos! No debería costarte demasiado encontrar alguna del siglo XIX.

SIGLO	COMPUTADORA	CARACTERÍSTICAS	IMAGEN

- 3.** Encontrá tres computadoras que no tengan un teclado alfanumérico como entrada ni una pantalla como salida. Sugerencia: pensá si algunos aparatos de uso habitual son o no computadoras.

COMPUTADORA	ENTRADA	SALIDA	IMAGEN

NOMBRE Y APELLIDO:

CURSO:

FECHA:

4. Otro punto importante es el tamaño. Buscá computadoras de hoy en día que tengan diferentes tamaños. Completá los nombres en la siguiente tabla:

LA COMPUTADORA CABE EN	NOMBRE DE LA COMPUTADORA	IMAGEN
Billetera		
Bolsillo		
Bolso		
Mesa		

Actividad 2

Computadoras por todos lados



DE A DOS

OBJETIVOS

- Explicitar que muchos artefactos de uso cotidiano se montan sobre una computadora.
- Analizar ejemplos de máquinas que no son computadoras.

MATERIALES



Computadora



Conexión a Internet



Ficha para estudiantes

DESARROLLO

En esta actividad pondremos en evidencia que las computadoras están muy presentes a nuestro alrededor. El recorrido nos ayudará a descubrir que, por ejemplo, los semáforos, las máquinas expendedoras de boletos y las impresoras se montan sobre una computadora. A la vez, también mostraremos máquinas en las que no intervienen dispositivos digitales. Con esto buscamos arrojar luz sobre la frontera entre aquello a lo que llamamos computadora y aquello que no lo es.

Vamos a usar una serie de preguntas para despejar dudas sobre si un artefacto puede o no ser una computadora o contenerla. Comenzamos pidiéndoles a los estudiantes que comparen una radio analógica con un reproductor de audio digital. Para el análisis, utilizamos una pregunta que copiamos en el pizarrón: **El artefacto ¿recibe entradas?**



En una radio se indica una frecuencia usando una perilla. El reproductor mp3 tiene una serie de botones que permiten operarlo. Por lo tanto, se verifica que ambos reciben entradas. Preguntamos a continuación: “**¿Genera salidas?**”. Los dos producen señales sonoras. Uno las envía a un cable y el otro las emite usando un parlante. Nuevamente, para ambos dispositivos la respuesta es afirmativa.






Indagamos luego: “**¿Toma decisiones? ¿Realiza cálculos? ¿Almacena información?**”. En el caso de la radio, la respuesta es negativa: recibe una señal que captura con la antena y usa circuitos electrónicos y el parlante para transformarla en ondas sonoras; sin embargo, realiza toda esa transformación sin que en ningún momento se digitalice la información, es decir, sin que se la traduzca a un sistema binario. Tampoco realiza cálculos ni almacena información. Esto da la pauta de que en este caso no se trata de una computadora.

Por el contrario, el reproductor de música guarda archivos en formato mp3, que luego son procesados por un *software* de reproducción de audio, que permite acciones tales como navegar entre canciones y armar y almacenar listas de reproducción. En este caso, sí estamos frente a un artefacto que es controlado por una computadora.

Invitamos a los estudiantes a ubicarse en parejas para pensar las respuestas a las preguntas planteadas considerando una aspiradora antigua y una inteligente. Deberían llegar a la conclusión de que la primera no es una computadora y sospechar que la segunda sí lo es. Con este ejemplo ilustramos que hay aparatos que a veces tienen computadoras y otras veces, no. Para seguir trabajando sobre estas ideas se puede realizar un análisis similar sobre automóviles o ascensores, por ejemplo.



Una vez finalizado el análisis de los artefactos, hacemos hincapié en que las tres preguntas con las que se trabajó constituyen un test para descartar que un artefacto sea una computadora. Toda vez que una de esas preguntas se conteste negativamente, estamos ante un objeto que no es una computadora. Repartimos la ficha de la actividad y les pedimos a los estudiantes que completen las consignas. A continuación se muestra una tabla con algunas respuestas posibles.

TIPO DE APARATO	¿TOMA DATOS, ACEPTA ENTRADAS, RECIBE ESTÍMULOS?	¿A PARTIR DE LOS DATOS DE ENTRADA, TOMA DECISIONES, CALCULA Y/O MEMORIZA ALGO?	¿GENERA UNA SALIDA O MUEVE ALGUNA COSA?	¿ES PROBABLE QUE TENGA UNA COMPUTADORA EN SU INTERIOR?
Órgano eléctrico 	Sí, tiene un teclado.	Sí, sintetiza sonidos artificiales, puede memorizar secuencias.	Sí, activa un parlante (se escucha sonido), prende luces.	Sí.
Piano 	Sí, tiene un teclado.	No, no toma decisiones ni memoriza datos.	Sí, se genera un sonido.	No.
Ascensor antiguo 	Sí, tiene una botonera para indicar el piso de destino.	No, solo se traslada verticalmente hasta un piso indicado.	Sí, mueve una estructura de hierro.	No.
Ascensor moderno 	Sí, tiene una botonera para indicar el piso de destino. También cuenta con un sensor que chequea si puede o no cerrar la puerta.	Sí. Por ejemplo, cuando se marcan muchos pisos, hay un programa que organiza cómo se recorren para minimizar, por ejemplo, el tiempo de espera y el consumo de energía.	Sí, mueve una estructura de hierro. Además, muestra datos en una pantalla.	Sí.
Lector de tarjetas del colectivo 	Sí, lee información de una tarjeta sin contacto.	Sí, controla que el saldo del pasajero alcance para pagar el boleto.	Sí, muestra en una pantalla el saldo de la tarjeta.	Sí.

CIERRE

Concluimos subrayando que, en nuestras vidas, las computadoras están mucho más presentes de lo que solemos creer. El uso de computadoras es una tendencia que creció mucho en la última década y se encuentra en plena expansión, cada vez a mayor velocidad.

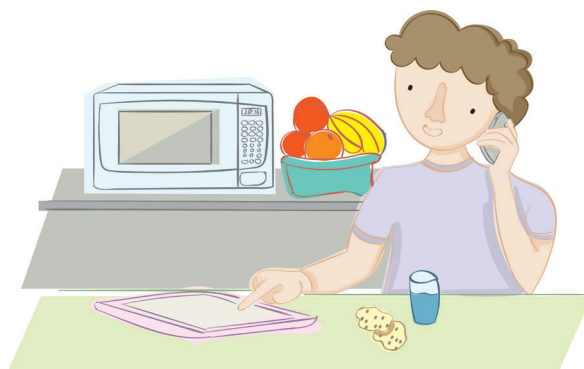
NOMBRE Y APELLIDO:

CURSO:

FECHA:

COMPUTADORAS POR TODOS LADOS

¡Las computadoras están por todas partes! Cada vez nos rodean más aparatos que contienen una computadora que los hace funcionar. Ahora vas a salir a cazar computadoras.



1. Anotá todos los aparatos que encuentres a tu alrededor. Puede ser en la escuela, en tu casa o donde vos quieras. Escribí en la primera columna todos los aparatos que veas. Por ejemplo, un televisor, un lavarropas o un cajero automático.

TIPO DE APARATO	¿ACEPTA ENTRADAS?	A PARTIR DE LOS DATOS DE ENTRADA, ¿TOMA DECISIONES, CALCULA O MEMORIZA ALGO?	¿GENERA UNA SALIDA O MUEVE ALGUNA COSA?	¿ES PROBABLE QUE TENGA UNA COMPUTADORA EN SU INTERIOR?

2. Para cada uno de los aparatos que elegiste, tenés que responder con **sí** o **no** las preguntas de la segunda a la cuarta columna. No olvides justificar cada respuesta.
3. Le llegó el turno a la última columna. El artefacto, ¿puede contener una computadora?

PARA QUE TENGAS EN CUENTA

En la evaluación, la calificación será más alta cuantos más ejemplos de computadora encuentres y cuanto más completas sean tus respuestas.

Actividad 3

Las partes de toda computadora

INDIVIDUAL

OBJETIVO

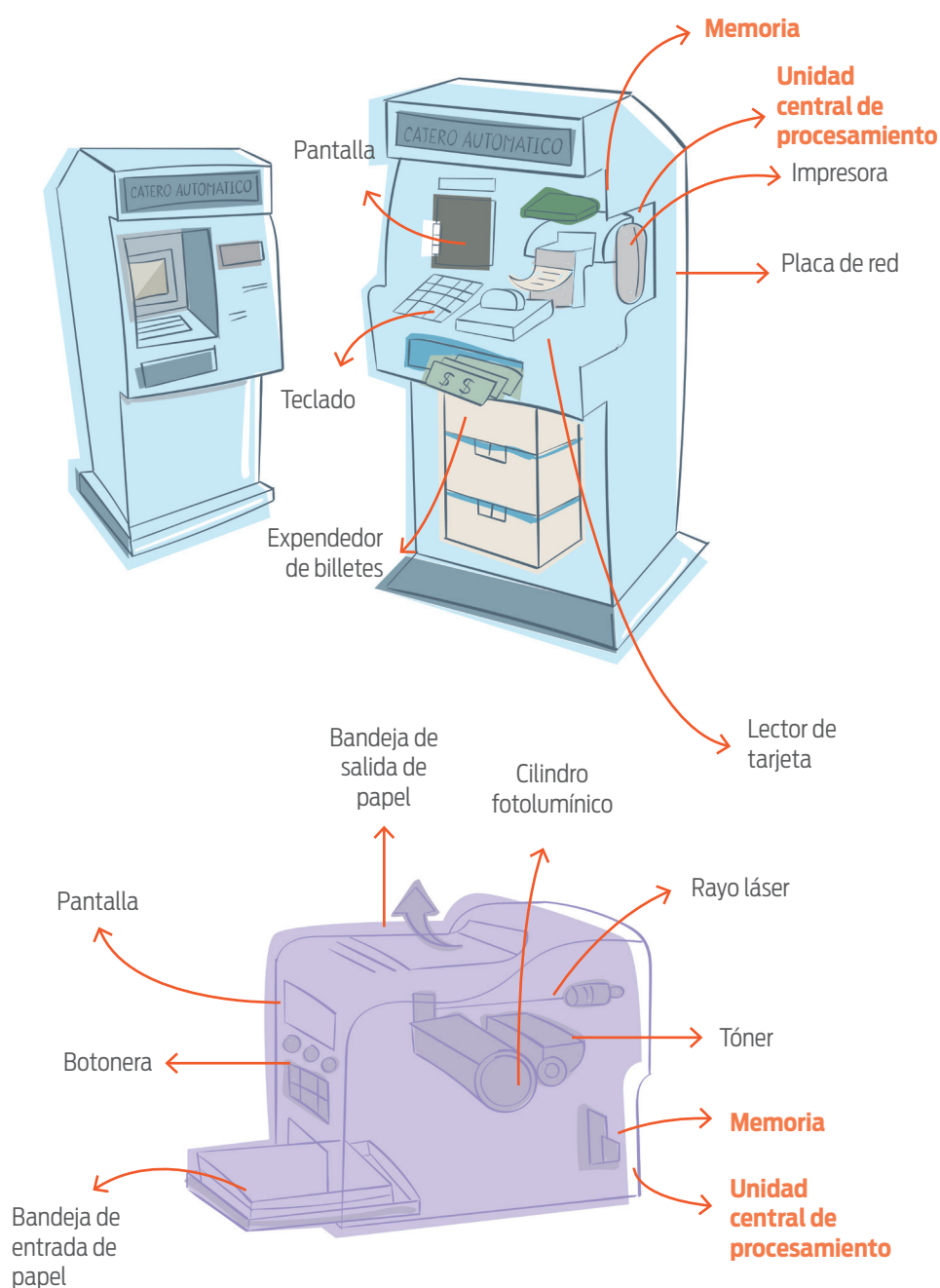
- Identificar los componentes que se encuentran presentes en todas las computadoras.

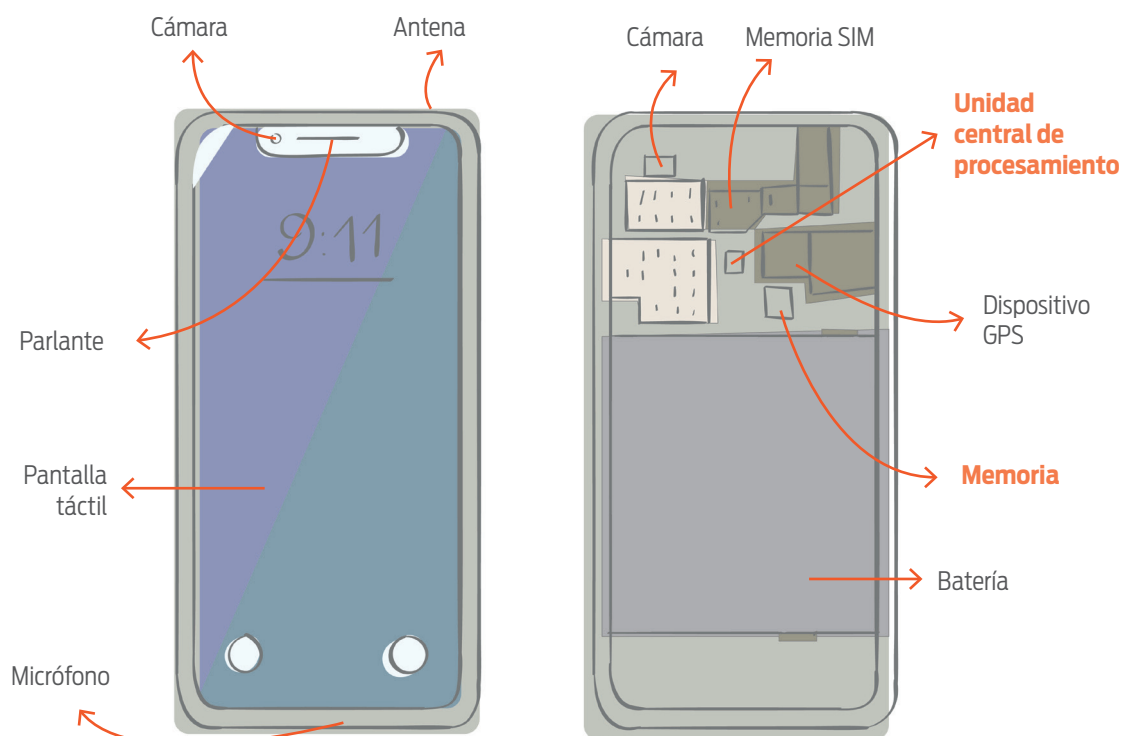
MATERIALES

- Ficha para estudiantes

DESARROLLO

Un buen ejercicio para seguir aproximándonos a una noción precisa de computadora consiste en identificar los componentes que todas tienen en común, cualquiera sea su aspecto. En el desarrollo de esta actividad, usamos como ilustración versiones muy simplificadas de un teléfono inteligente, una impresora láser y un cajero automático. Sin embargo, conviene recordar que estos dispositivos son simplemente una excusa para reconocer aquellos elementos indispensables en cualquier computadora.





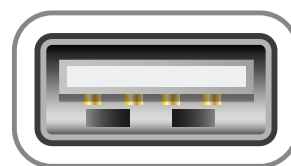
Repartimos la ficha a los estudiantes. Allí van a encontrar las tres computadoras usadas en esta actividad junto con la indicación de sus diferentes partes, tanto internas como externas. En primer lugar deben localizar los componentes comunes a todas ellas.

Una vez que los alumnos analicen las imágenes, detectarán que en todos los dispositivos hay una unidad central de procesamiento (CPU) y una memoria. Nos detenemos brevemente sobre estos componentes. Contamos que la CPU realiza cálculos que le permiten procesar información y que, además, toda esa información se encuentra almacenada en la memoria en todo momento.

Es probable que los estudiantes también noten que las tres computadoras tienen una pantalla. En primer lugar, vale la pena aclararles que no todas las computadoras tienen una; por ejemplo, pueden pensar en un típico *router* wifi, ya mencionado en una actividad anterior. Además, es interesante señalar que las tres pantallas muestran información. Por ejemplo, en la del teléfono vemos íconos de las aplicaciones instaladas, la hora, etc.; en la de la impresora, información sobre el estado del tóner, si hay o no papel en la bandeja de entrada, etc.; y en la del cajero automático, observamos indicaciones para operar el dispositivo, datos de cuentas bancarias, etc. Esto convierte a estas pantallas en dispositivos de salida. Además, en el caso del móvil permite el ingreso de datos. Por ejemplo, un número de teléfono para realizar una llamada; es decir que, en este caso, la pantalla es también un dispositivo de entrada. A los elementos que cumplen ambas funciones se los conoce como *dispositivos de entrada y salida*.

Por último, les pedimos que identifiquen en las computadoras otros periféricos, tanto de entrada como de salida, aun cuando sean diferentes en cada una de ellas. Podrían mencionar que el teléfono cuenta con un micrófono y una cámara como entradas y un parlante como salida; que la impresora tiene un teclado como entrada y que el cajero tiene como entrada un teclado y como salidas una impresora de tickets y un dispensador de billetes.

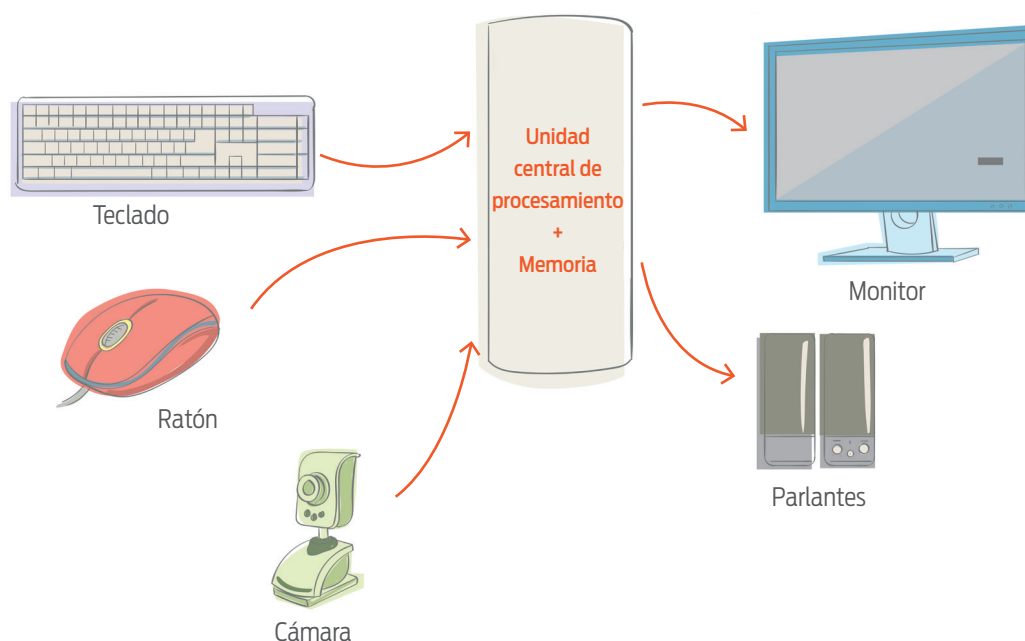
Continuamos la actividad con la siguiente reflexión, que va más allá de las computadoras que ilustran la actividad. Actualmente, es habitual que las computadoras portátiles y de escritorio posean puertos USB¹ que permiten conectar una gran cantidad de periféricos; por ejemplo, ratones, discos externos, teléfonos, etc. Por lo tanto, estos puertos nos posibilitan ampliar los dispositivos de entrada y salida que interactúan con la computadora. La versatilidad provista por estos canales se basa en una técnica conocida como *enchufar y usar* (*plug and play*, en inglés).



Puerto USB

DISPOSITIVOS DE ENTRADA

DISPOSITIVOS DE SALIDA



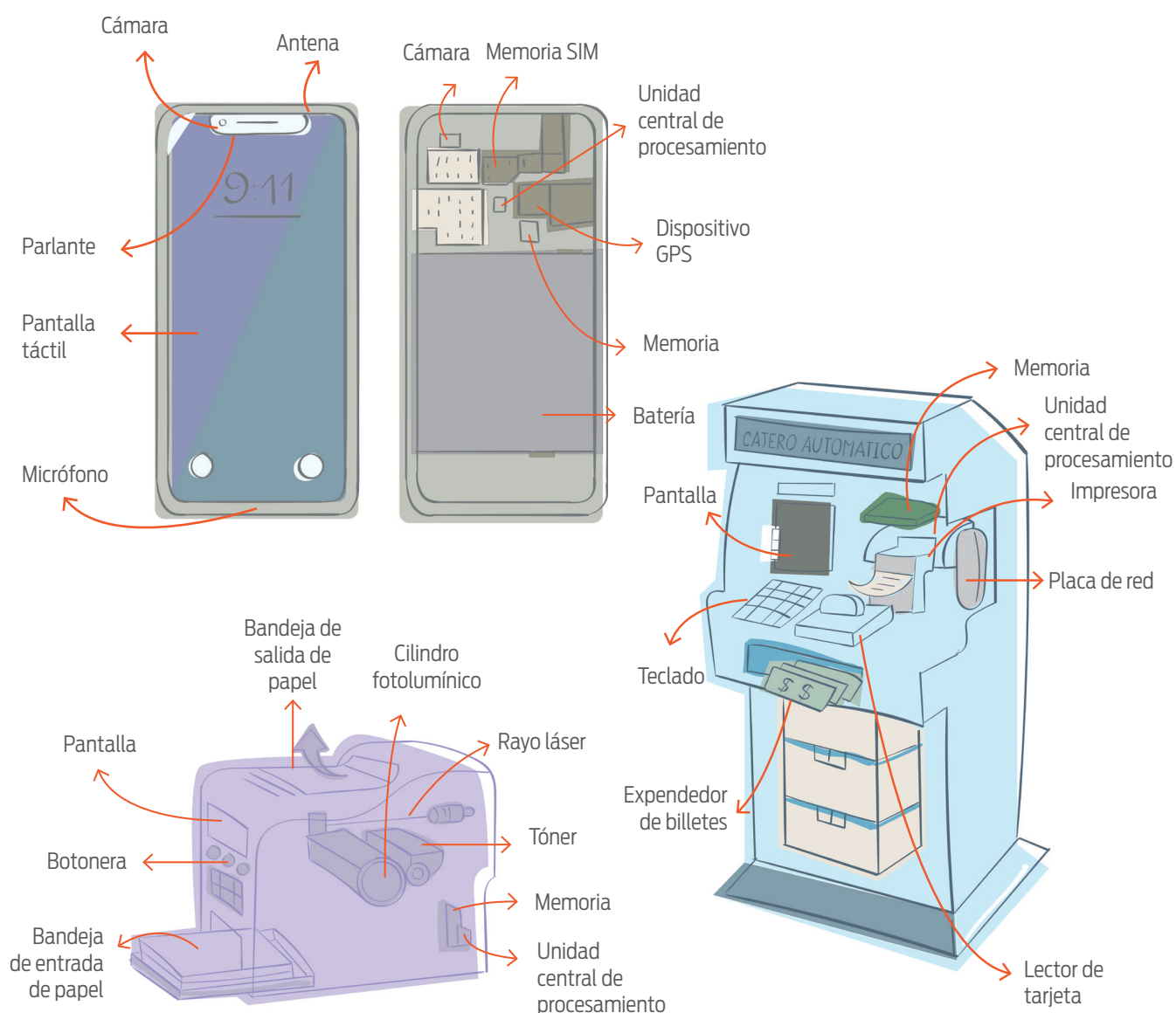
CIERRE

Cerramos la actividad contándoles a los estudiantes que, en 1945, el matemático de origen austrohúngaro John von Neumann presentó un modelo teórico de computadora que fue la base de las computadoras modernas. Von Neumann proponía como partes constitutivas de una computadora una unidad central de procesamiento, una memoria y dispositivos de entrada y de salida.

¹Del inglés, *Universal Serial Bus*.

LAS PARTES DE TODA COMPUTADORA

¿Sabés qué tienen en común un teléfono inteligente, una impresora láser y un cajero automático? Son computadoras. Tienen diferentes diseños, pero hay componentes que se repiten en los tres.



1. Observá las partes de un teléfono inteligente, una impresora láser y un cajero automático. ¿Qué componentes tienen en común?

NOMBRE Y APELLIDO:

CURSO:

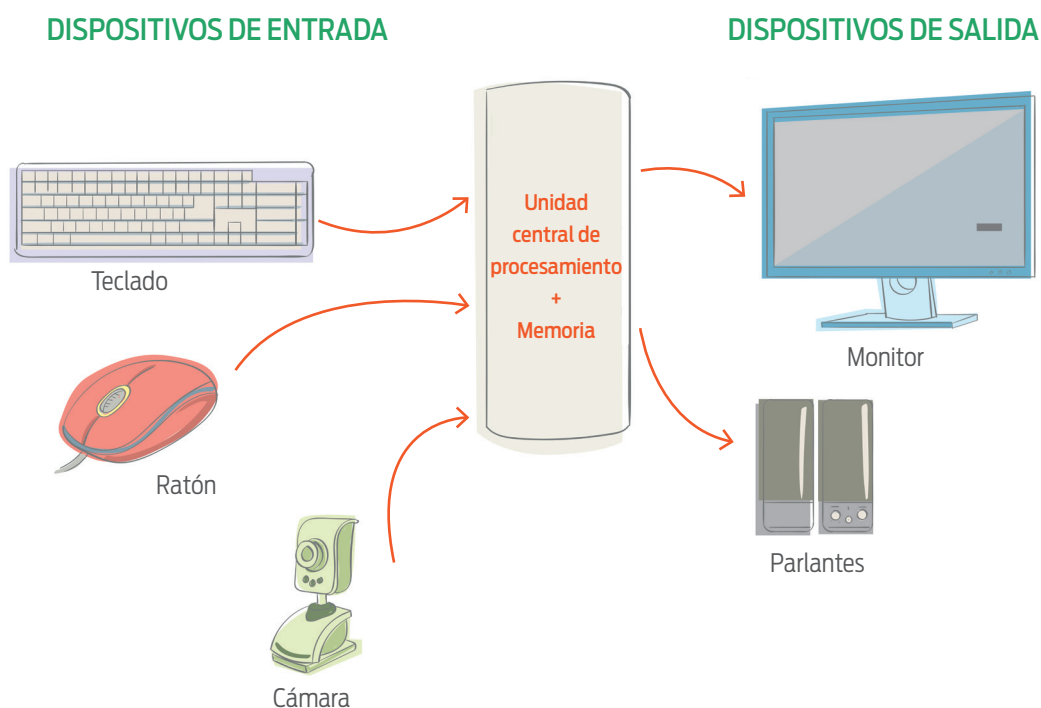
FECHA:

2. En cada uno de ellos, ¿cómo se hace para ingresar información?
¿Qué tipo de salida producen?

DISPOSITIVO	¿CÓMO SE INGRESA INFORMACIÓN?	¿QUÉ TIPO DE SALIDA PRODUCEN?
Teléfono inteligente		
Cajero automático		
Impresora láser		

JOHN VON NEUMANN

John von Neumann fue un matemático austrohúngaro que, en el año 1945, presentó un modelo teórico de computadora que fue la base de todas las computadoras modernas. Von Neumann proponía como partes constitutivas de una computadora una unidad central de procesamiento, una memoria y dispositivos de entrada y salida.



Actividad 4




Una nueva máquina

GRUPAL (4)

OBJETIVOS

- Diseñar una computadora de propósito específico.
- Mostrar que la tecnología interviene en la vida de las personas.
- Ejercitar el pensamiento creativo.
- Practicar la exposición de ideas.

MATERIALES

-  Cartulina
-  Fibras de colores
-  Ficha para estudiantes

DESARROLLO

Es el turno de los estudiantes para proponer y diseñar una computadora. Comenzarán identificando un problema y proponiendo una solución computacional para resolverlo.

Comenzamos diciéndoles a los estudiantes: “¡Que levante la mano aquel al que le guste lavar los platos!”. Es esperable que nadie en la clase lo haga. “Muy bien, parece que en esta clase son todos perezosos. Por suerte existe el lavavajillas, una máquina que se encarga de lavar los platos por nosotros. También limpia vasos y sartenes. ¿Pueden creerlo? Una verdadera maravilla”. Repartimos la ficha a los estudiantes y los alentamos a que completen las consignas. En la primera se les pide que describan un problema que podría resolverse con la ayuda de una computadora. Les planteamos la pregunta: “¿Qué problema querrían resolver?”. Es importante que todos se tomen un tiempo para identificar claramente el problema que se propongan resolver. En el caso del ejemplo con el que iniciamos la actividad, el problema consiste en conseguir limpiar la vajilla sin hacer demasiado esfuerzo.

La segunda consigna consiste en describir la arquitectura de un artefacto que solucione el problema identificado. La propuesta debe involucrar el uso de algún tipo de dispositivo computacional. Siguiendo con el lavavajillas, podemos decirles a los estudiantes que es un artefacto parecido a un lavarropas, pero que en lugar de ropa lava platos, vasos, cubiertos y otros utensilios de cocina. Tiene una computadora a la que es posible dar instrucciones a través de algunos botones. Se puede, por ejemplo, elegir entre distintos tipos de programas de lavado o seleccionar la temperatura del agua. Además, tiene una pantalla que muestra diferentes mensajes, como la hora o el tiempo restante para completar un lavado. El artefacto está conectado a una canilla y a un desagüe, por los que circula el agua que usa. Cuenta también con pequeños recipientes en los que se deposita jabón en polvo, abrillantador y sal antisarro. Por último, tiene una puerta delantera que abrimos y cerramos para poner y sacar los platos, entre otras cosas.

La tercera consigna indaga sobre las entradas y salidas del dispositivo. Continuando con el ejemplo, como entrada usamos botones, que nos dan la posibilidad, por ejemplo, de indicar si se hará un lavado de media carga. Hay entradas menos evidentes, como los sensores de control de nivel de agua y de temperatura. La salida se



muestra en la pantalla, donde aparece cierta información, como los niveles de consumo de energía o la tasa de platos lavados por semana. Algunos artefactos tienen además pequeñas luces led, que se usan como alarmas ante alguna situación anómala como, por ejemplo, el suministro insuficiente de agua. Tanto la pantalla como los ledes son salidas.

La cuarta consigna pide identificar el procesamiento que realiza la computadora dentro del artefacto diseñado. Para el electrodoméstico que venimos analizando, alcanza con respuestas tales como: “La computadora ejecuta un programa que, de acuerdo a lo ingresado por el usuario, realiza uno u otro lavado”.

La quinta y última consigna invita a los estudiantes a diseñar un afiche que exponga cómo es y qué hace la máquina inventada. La actividad concluye con la exposición de cada grupo sobre la computadora que diseñaron.

RÚBRICAS

Al final de la ficha de los estudiantes, incluimos una rúbrica para que los alumnos sepan qué estaremos evaluando y qué criterios utilizaremos.

CIERRE

Comentamos a los estudiantes que, en la industria pesada, suelen usarse computadoras de propósito específico que se conocen como PLC (las siglas se deben a su nombre en inglés, *Programmable Logic Controller*). Son muy usadas en las industrias mecánica y electrónica, donde controlan el comportamiento de distintos tipos de máquinas. Podemos mencionar como ejemplo una computadora que controla el brazo que coloca los corchos en las botellas de vino.

Para la clase siguiente, les pedimos que traigan de sus casas computadoras portátiles, de escritorio o routers en desuso.

OTRAS OPCIONES

Los estudiantes también pueden traer artefactos en desuso que contengan computadoras tales como teléfonos celulares, reproductores de DVD, decodificadores para televisión por cable, entre otros. Debido a la gran diversidad de modelos existentes, recomendamos averiguar marcas, años y demás datos relevantes de cada uno de los artefactos. De este modo, es posible encontrar la forma adecuada de abrirlos y prever los elementos que los estudiantes encontrarán en su interior (ver ficha 5 de la próxima actividad).

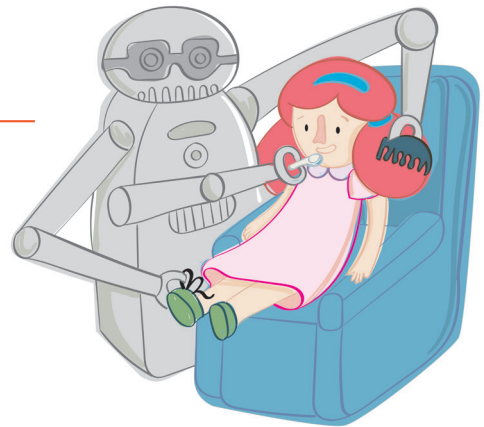
NOMBRE Y APELLIDO:

CURSO:

FECHA:

UNA NUEVA MÁQUINA

Cada ser de este planeta tiene algún problema. Más grande o más pequeño, algún problema siempre hay. En algunos casos, existen dispositivos específicos para resolverlos. Pensemos, por ejemplo, en el lavavajillas. Para poder satisfacer nuestros deseos de descansar más, contamos con un artefacto que lava los platos por nosotros. ¡Es sensacional!



1. Seleccioná a un ser humano del planeta Tierra y contá qué problema necesita resolver.

2. Diseñá un artefacto que lo ayude a resolver el problema. Tiene que contener una computadora. Indicá qué hace el artefacto y qué partes tiene.

3. ¿Qué información recibe como entrada? Y la salida, ¿cómo la expresa?

4. ¿Procesa los datos de entrada para producir una salida? ¿Qué hace?

NOMBRE Y APELLIDO:

CURSO:

FECHA:

5. Armá un afiche de la computadora que diseñaste en los puntos anteriores, que muestre cómo es e indique sus partes. El propósito es que, cuando lo vea un amigo tuyo, pueda entender sin problemas de qué se trata tu invento.

EVALUACIÓN

Estas son las tareas y la forma de calificar que tu docente considerará para evaluar cómo resolvés las actividades.

CALIFICACIÓN	ES NECESARIO TRABAJAR MÁS	BUENO-MUY BUENO	EXCELENTE
Planteo de un problema y su solución	La computación escasamente puede contribuir a resolver el problema identificado por el estudiante.	El estudiante identifica un problema, aunque la solución computacional que diseña no lo resuelve del todo.	El estudiante identifica claramente el problema y plantea un diseño computacional adecuado para resolverlo.
Componentes de la máquina inventada	No se identifica la mayoría de los componentes del dispositivo diseñado: unidad central de procesamiento, memoria, periféricos de entrada y de salida específicos.	Se identifica la mayoría de los componentes: unidad central de procesamiento, memoria, periféricos de entrada y de salida.	Se identifican todos los componentes del dispositivo diseñado. La máquina tiene un diseño simple y efectivo.

Actividad 5


¡A conocer la computadora por dentro!

GRUPAL (4)

OBJETIVOS

- Conocer las computadoras por dentro.
- Identificar los componentes de una computadora.

MATERIALES

-  Computadoras de escritorio
-  Computadoras portátiles
-  Routers wifi
-  Destornilladores
-  Ficha para estudiantes

DESARROLLO

En esta actividad buscamos que los estudiantes pierdan el miedo a desarmar computadoras. Al finalizar la clase anterior, se les pidió que trajeran artefactos en desuso que contuvieran computadoras tales como computadoras de escritorio, portátiles y routers wifi (que son los artefactos con los que ilustramos la actividad).

A los estudiantes que trajeron teléfonos celulares, reproductores de DVD, decodificadores para televisión por cable, entre otros posibles artefactos en desuso, debemos entregarles material complementario al de la ficha. Este debería incluir las instrucciones para desarmar esos artefactos y una ilustración que muestre las partes que encontrarán en su interior.

En algún momento, las computadoras presentadas en esta actividad quedarán obsoletas. Este parece ser el destino de todos los dispositivos tecnológicos. Si ya hubieran caducado, sugerimos adaptar la actividad usando otros dispositivos.

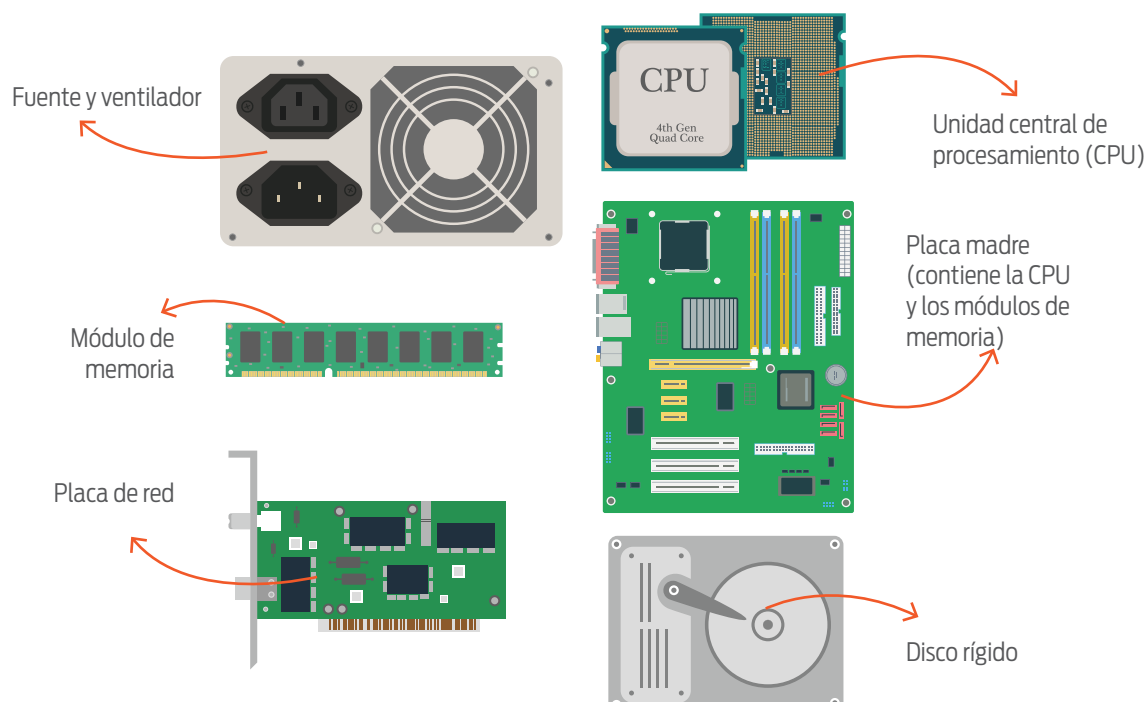
Les pedimos a los estudiantes que formen grupos de 4 integrantes y les repartimos la ficha. Allí se presentan instrucciones para desarmar una computadora de escritorio, una portátil y un router wifi. A continuación se describe cómo trabajar con cada uno de estos artefactos.

Computadora de escritorio

Si la computadora está encendida, la apagamos. Luego, la desenchufamos del tomacorriente. Desconectamos todos los cables que se encuentran enchufados al gabinete, tanto en el frente como en la parte posterior. De este modo, habremos desconectado todos los periféricos, lo cual facilitará la inspección del interior de la máquina.

Desatornillamos la tapa del gabinete y la retiramos. Allí nos encontraremos con la placa madre, sobre la cual está montada la unidad central de procesamiento. Además, cuenta con muescas en las que se encastran los módulos de memoria.

Las computadoras están equipadas con un ventilador que disipa el exceso de calor que genera el funcionamiento del equipo. Presumiblemente, también nos encontraremos con un disco rígido. Es el soporte físico en el que guardamos toda nuestra información en forma persistente, es decir que ahí se conserva aun cuando apaguemos el equipo. Otro componente habitual es la placa de red. Permite que el dispositivo se conecte con otras computadoras. Finalmente, se encuentra la fuente, que suministra energía eléctrica a todos los componentes internos de la computadora.



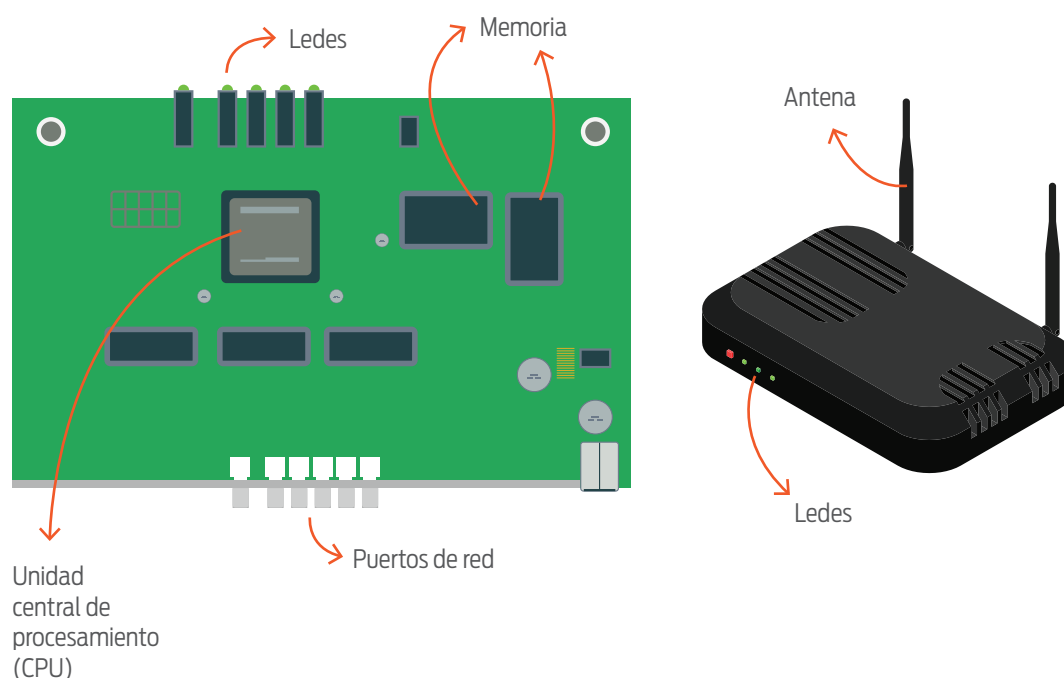
Computadora portátil

En este caso no podremos observar todos los componentes. Desarmar la carcasa de estos dispositivos suele ser complicado, además de que se necesitan destornilladores adecuados. Comenzamos corroborando que se encuentre apagada y desenchufamos todos los cables, incluyendo los de suministro de energía y los de los periféricos. La cerramos y, con mucho cuidado, la damos vuelta y la apoyamos sobre alguna superficie limpia. En algunos modelos es posible retirar las tapas que protegen los módulos de memoria y el disco rígido.



Router wifi

Un **router** wifi es una computadora que no posee teclado ni monitor. Tiene puertos de entrada y salida para conexiones alámbricas y antenas para enviar y recibir información de modo inalámbrico. Además, estos dispositivos suelen venir equipados con monitores lumínicos. Se trata de pequeños ledes que brindan información sobre aspectos tales como la transferencia de datos y el estado de la conexión a la red.



Comenzamos retirando todos los cables que tenga enchufados. Si es posible, desenroscamos las antenas. Si la carcasa se encuentra cerrada con tornillos, los retiramos. Algunos modelos simplemente se separan tirando del frente. Una vez abierto el artefacto, retiramos la placa principal. Allí encontraremos la unidad central de procesamiento. En algunos ejemplares, la memoria está integrada a la unidad central de procesamiento; si este es el caso, no podremos observarla.

RÚBRICAS

Al final de la ficha de los estudiantes, incluimos una rúbrica para que los alumnos sepan qué estaremos evaluando y qué criterios utilizaremos.

CIERRE

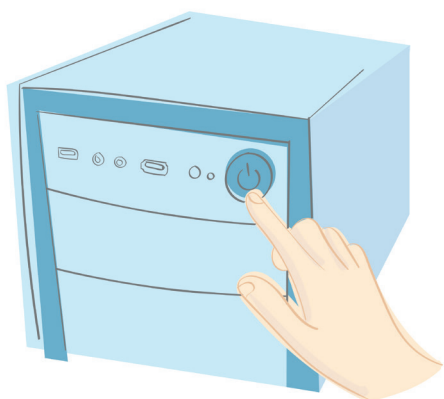
A modo de cierre, les comentamos a los estudiantes que en una computadora, en general, los distintos componentes pueden encastrarse solo en pocos lugares. Por lo tanto, si los desconectamos para observarlos de cerca, luego será sencillo volver a dejar todo en su lugar.

¡A CONOCER LA COMPUTADORA POR DENTRO!

¡En esta actividad vamos a desarmar computadoras! Podés trabajar con una de escritorio, una portátil o un router wifi, así como con un teléfono celular, un reproductor de DVD o un decodificador de televisión por cable. Seguí las instrucciones y animate a explorar estos artefactos por dentro. Así también se aprende. Al final de la ficha hay unas imágenes que te pueden servir de referencia.



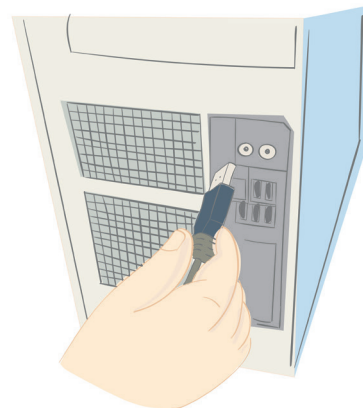
COMPUTADORA DE ESCRITORIO



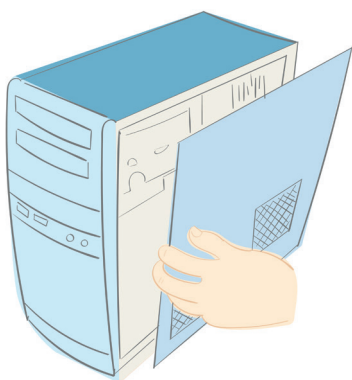
1) Apagá la computadora.



2) Desenchufá el cable de alimentación eléctrica.



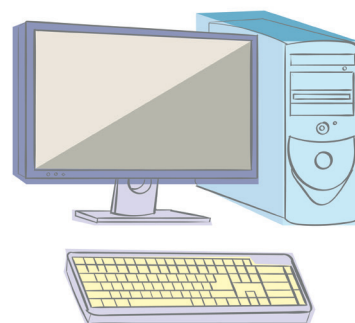
3) Desenchufá el resto de los cables, si los hay.



4) Sacá la carcasa del gabinete. Si sacaste tornillos, guardalos en algún lugar para no perderlos.



5) Buscá la memoria y el la unidad central de procesamiento.



6) Volvé a armar la computadora.

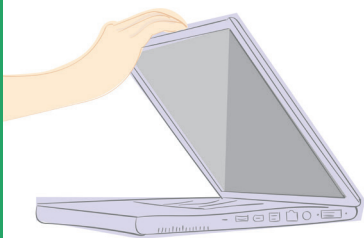
¿Identificás otros elementos? ¿Un disco rígido?
¿La placa madre? ¿Alguna otra cosa?

NOMBRE Y APELLIDO:

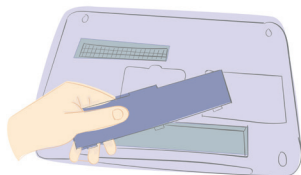
CURSO:

FECHA:

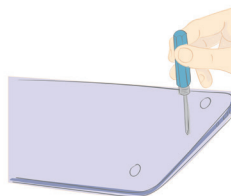
COMPUTADORA PORTÁTIL



1) Apagá la computadora, cerrá la tapa y desenchufá el cargador.



2) Dala vuelta y sacale la batería.

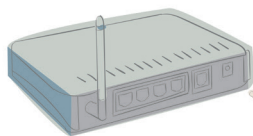


3) Desatornillá las tapitas. Te vas a encontrar con el disco rígido y la memoria.

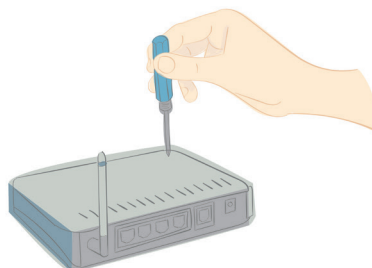


4) Armala de nuevo.

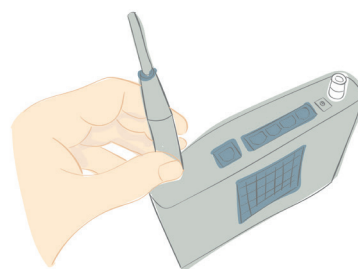
ROUTER WIFI



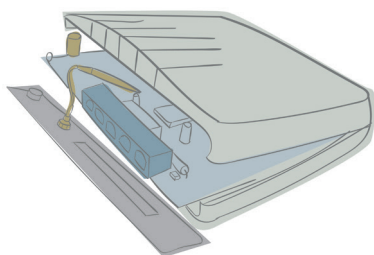
1) Desenchufalo y sacale el cable.



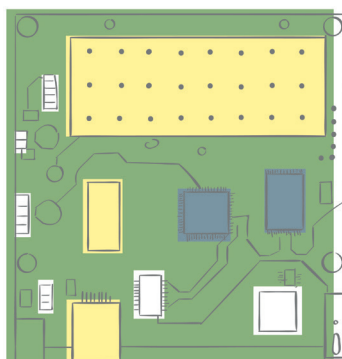
2) Sacá todos los tornillos. Puede haber alguno debajo de las patas de goma.



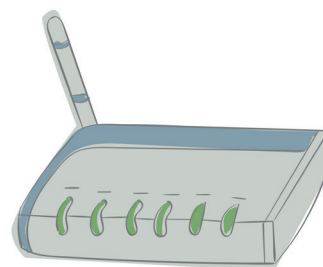
3) Desconectá las antenas.



4) Sacá la placa principal.



5) Buscá la unidad central de procesamiento y la memoria.



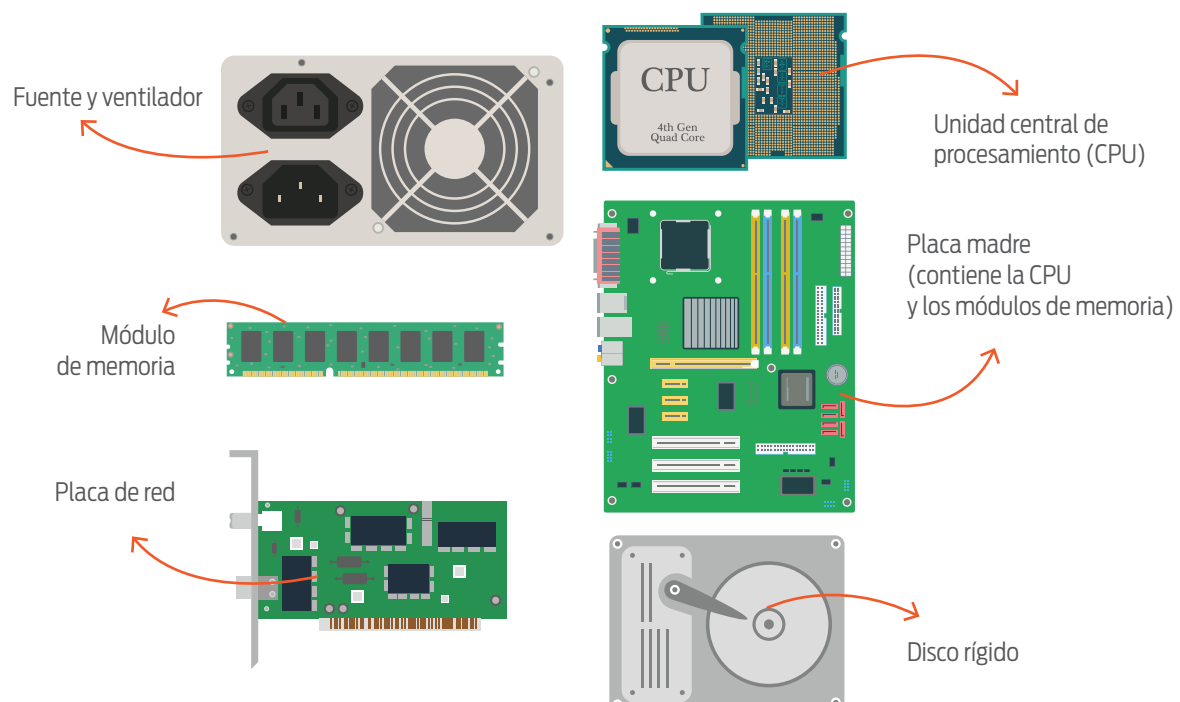
6) Volvé a armarlo.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

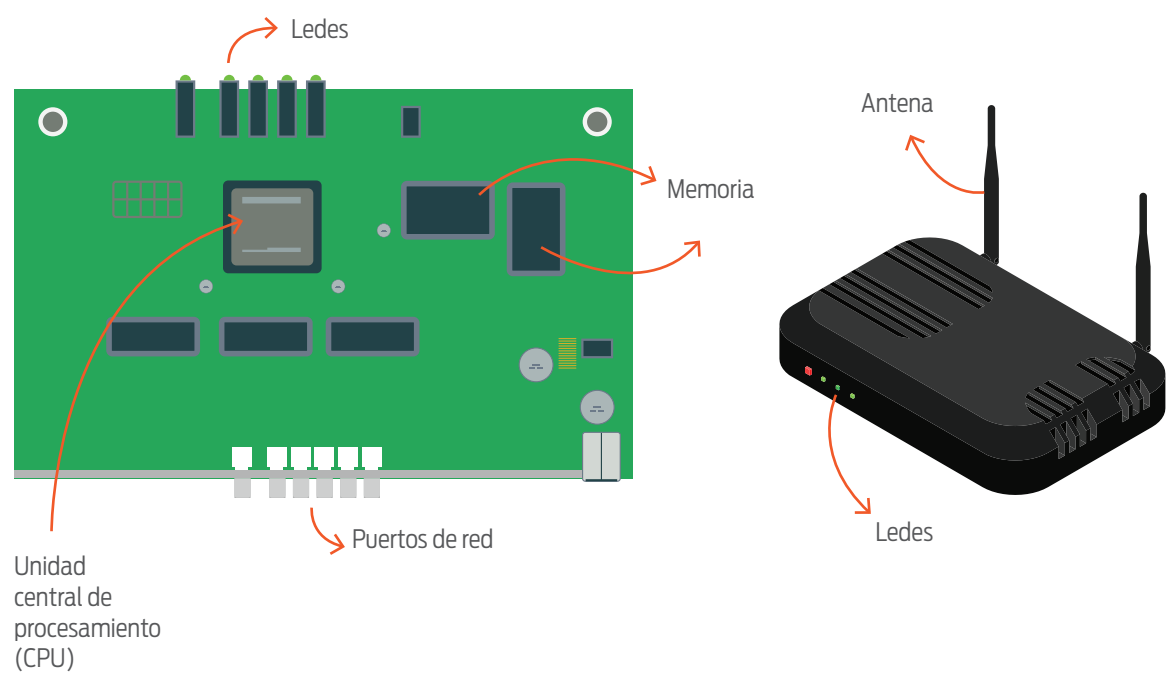
Componentes de una computadora de escritorio



Componentes de una laptop



Componentes de un router wifi.



EVALUACIÓN

Estas son las tareas y la forma de calificar que tu docente considerará para evaluar cómo resolvés las actividades.

CALIFICACIÓN	ES NECESARIO TRABAJAR MÁS	BUENO-MUY BUENO	EXCELENTE
Reconocimiento de los componentes de la máquina	El estudiante reconoce pocos componentes de la computadora.	El estudiante reconoce la mayoría de los componentes mencionados en la ficha.	El estudiante reconoce todos los componentes de la computadora.

LA MEMORIA Y LA UNIDAD CENTRAL DE PROCESAMIENTO

SECUENCIA DIDÁCTICA 1

LA MEMORIA

¿Qué hay en la memoria?

La memoria rápida y la memoria grande

SECUENCIA DIDÁCTICA 2

EL FUNCIONAMIENTO INTERNO

La máquina criptográfica

La materia prima con la que trabajan las computadoras es la información. Para que esta sea procesada por un dispositivo computacional es indispensable que exista un componente físico donde representarla. En la primera secuencia didáctica, se propone una serie de actividades que permiten comprender ciertos aspectos del funcionamiento de la memoria.

Por su parte, la unidad central de procesamiento (CPU) es el componente de *hardware* que se encarga de procesar, transformar y producir información. La segunda secuencia didáctica permitirá a los estudiantes observar cómo la CPU interactúa con otros componentes de una computadora para ejecutar las instrucciones de un programa.



Secuencia Didáctica 1

LA MEMORIA

Cuando hablamos de memoria, nos referimos a una serie de componentes físicos que tienen la capacidad de representar información. La secuencia didáctica comienza con una actividad que muestra la organización más básica de la memoria y nos permite reflexionar sobre su contenido. La segunda y última actividad revelan que hay memorias de distinto tipo y que se las puede clasificar de acuerdo con aspectos tales como la velocidad y el tamaño. Se trata de una tarea para que los estudiantes comprueben en sus hogares qué información guardan algunas computadoras en cada tipo de memoria.

.....

OBJETIVOS

- Comprender cómo es el funcionamiento básico de la memoria.
- Diferenciar la memoria rápida de la memoria grande.

.....

Actividad 1





¿Qué hay en la memoria?

INDIVIDUAL

OBJETIVOS

- Representar la memoria como una tira de celdas.
- Observar que en cada celda de memoria se almacena uno de dos valores posibles.
- Destacar que la memoria no interpreta el contenido que almacena.

MATERIALES

-  Lápices de colores
-  Computadora
-  Conexión a Internet
-  Ficha para estudiantes

DESARROLLO

En esta actividad retomamos ideas sobre la representación de datos para mostrar que la memoria no le atribuye significado a la información que almacena. Comenzamos repartiendo la ficha a los estudiantes, en la que encontrarán la siguiente cuadrícula:

0	0	1	1	1	1	0	0	1	0	1	0	0	0	0	0
0	1	0	1	0	1	0	0	1	1	1	1	0	1	0	0
1	1	0	0	0	0	1	0	0	0	0	0	1	1	0	1
1	1	0	1	1	0	1	1	0	0	1	1	0	0	1	0
0	0	1	1	0	1	1	0	0	1	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	1	1	0	1	1	0	1

Les contamos que lo que observan en la ficha es una representación de la memoria. Si bien (como ya vimos en la secuencia didáctica 1 del capítulo 5) el componente físico utiliza dos niveles de voltaje distintos, para representar los posibles valores de un bit nosotros usamos los símbolos 0 y 1. Les explicamos a los estudiantes: “Vamos a pensar la memoria como una tira de celdas. En cada celda se almacena, o bien un cero, o bien un uno. Los programas, al ejecutarse, tienen la posibilidad de leer el contenido de las celdas y también de modificarlo”. Les preguntamos: “¿Qué información esconden estos números?”. Es probable que no estén en condiciones de responder la pregunta. Les comentamos que a lo largo de esta actividad trataremos de encontrar una respuesta.

La manipulación de bits suele ser engorrosa. Habitualmente se los agrupa de a 8 y a cada grupo se lo conoce como **byte**. Con un byte se pueden representar números entre 0 y 255. Les pedimos a los estudiantes que resuelvan la primera consigna de la ficha. Se les pide que escriban el contenido de la memoria como secuencia de bytes. Para facilitar la resolución de esta parte de la tarea, podemos indicarles que usen un conversor de binario a decimal en línea.¹ Deberían arribar a los siguientes valores:

60	160	84	244	194	13	219	50	54	72	160	237
----	-----	----	-----	-----	----	-----	----	----	----	-----	-----

1. Se puede usar, por ejemplo, el conversor disponible en <https://goo.gl/C68hEt>. Si en el aula no hay acceso a Internet, recomendamos buscar previamente una tabla completa que incluya las representaciones binarias de los números del 0 al 255, e imprimir varias copias para hacer circular entre los estudiantes.

¿CÓMO SE TRADUCÍA DE UN SISTEMA A OTRO?

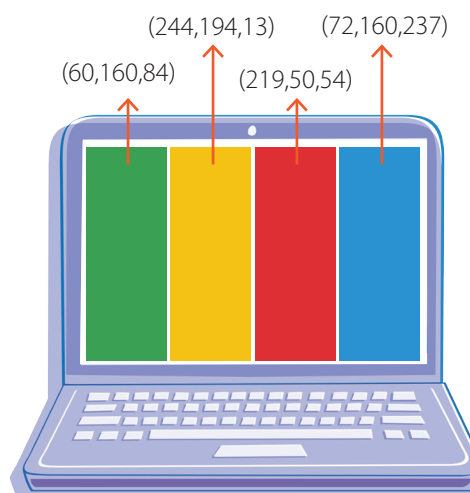
En el capítulo 5, “Representación de datos”, vimos cómo se puede representar números usando los símbolos 1 y 0, es decir, un sistema binario. Si realizamos las actividades en clase, podemos mencionar el juego de las cartas (ver la secuencia didáctica 1).

A continuación, les preguntamos a los estudiantes: “¿Cómo hacen las computadoras para trabajar con colores?”. Es esperable que alguien conteste que usan la codificación RGB. “Efectivamente, es habitual que los programas que manipulan imágenes usen el modelo de color RGB. Recordemos que, en este tipo de codificación, se usan tres bytes para representar un color. Los bytes indican la intensidad con que debe encenderse cada una de las luces que componen un píxel”.

Les pedimos que trabajen sobre la segunda consigna de la ficha. Allí se presenta el programa ficticio Coloreame. Este toma datos almacenados en la memoria, los interpreta como cuatro colores codificados en RGB y muestra cuatro tiras verticales en la pantalla, una de cada color identificado. Para resolver la consigna, en primer lugar tienen que investigar qué colores representan los números almacenados en la memoria. Deben agrupar los bytes de a tres y, usando alguna aplicación adecuada¹, reconocer que se trata de un verde, un amarillo, un rojo y un azul.

Una vez identificados los colores, los estudiantes tienen que usarlos para pintar las franjas del monitor que se encuentra en la ficha.

Cuando hayan completado el desafío, les contamos que ahora trabajaremos con el programa imaginario Textito. Se trata de una aplicación que lee contenido de la memoria, lo interpreta como texto codificado, lo decodifica y lo muestra en la pantalla. Usa la codificación de letras Letrados que se muestra a continuación.



13	244	7	89	72	98	112	75	60	202	66	15	84	219	54
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
178	160	33	214	84	151	39	50	93	67	113	107	57	194	237
Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	,	.

¹ Por ejemplo, se puede usar la que está disponible en <https://goo.gl/Z1z1hj>.

Les pedimos a los estudiantes que resuelvan la tercera consigna. Tienen que averiguar qué mostraría Textito si leyese la misma porción de memoria que Coloreame. Es esperable que, luego de unos minutos, descubran el texto: “Hola, mundo”.

Finalmente, preguntamos a los estudiantes: “Entonces, ¿qué era el contenido almacenado en la memoria?”. A partir de la información brindada en la actividad, no podemos saberlo. Podría tratarse de colores, de texto o de otro tipo de información.

Les presentamos la consigna 4: “¿La memoria entiende la información que almacena?”. Guiamos la reflexión de modo tal de concluir que la memoria no les atribuye significado a los datos que almacena. Son los programas los encargados de interpretarlos.



CIERRE

A modo de cierre, comentamos con los estudiantes que, en la práctica, a diferencia de lo que sucedió en esta actividad, es muy improbable que dos programas que manipulan información de distinta naturaleza produzcan resultados significativos leyendo la misma porción de memoria. En este ejercicio los valores fueron cuidadosamente escogidos para subrayar que el contenido de la memoria no tiene un significado en sí mismo.

NOMBRE Y APELLIDO:

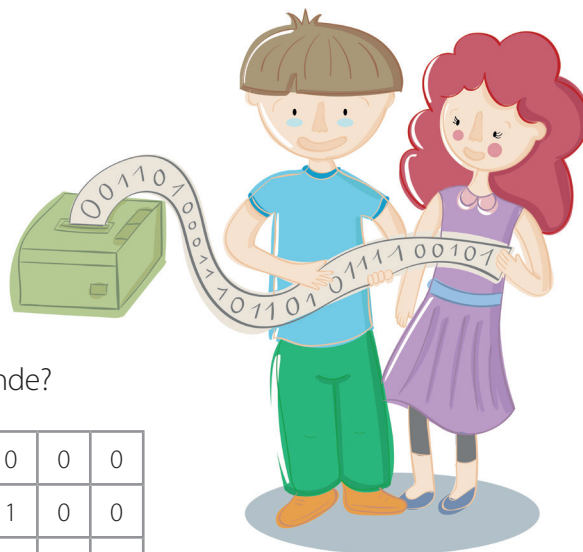
CURSO:

FECHA:

¿QUÉ HAY EN LA MEMORIA?

¿Sabías que la memoria es como una larga cinta con casilleros en los que se guarda un 0 o un 1? Mirá, por ejemplo, el siguiente fragmento. ¿Qué información esconde?

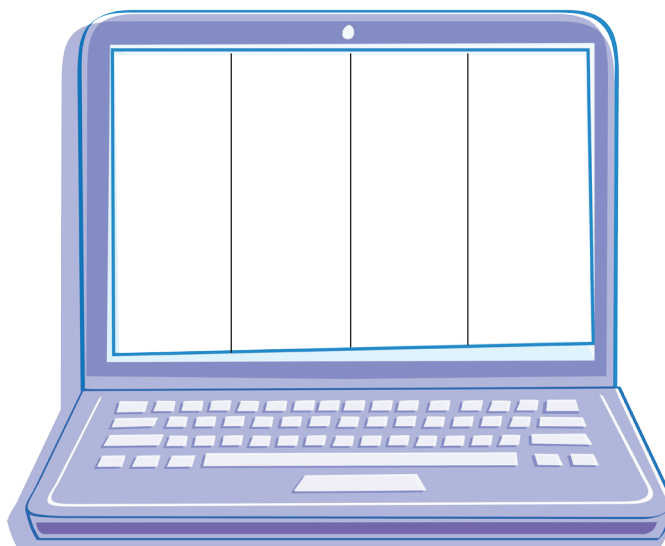
0	0	1	1	1	1	0	0	1	0	1	0	0	0	0	0
0	1	0	1	0	1	0	0	1	1	1	1	0	1	0	0
1	1	0	0	0	0	1	0	0	0	0	0	1	1	0	1
1	1	0	1	1	0	1	1	0	0	1	1	0	0	1	0
0	0	1	1	0	1	1	0	0	1	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	1	1	0	1	1	0	1



1. ¡Trabajar con bits es muy complicado! Agrupalos de a 8 y mostrá el contenido de la memoria como una tira de bytes. Podés usar el conversor disponible en <https://goo.gl/C68hEt>.

--	--	--	--	--	--	--	--	--	--	--	--

2. Aquí te presentamos el programa Coloreame. Se trata de una aplicación que lee la memoria, interpreta los valores almacenados como la codificación RGB de cuatro colores, y los usa para pintar cuatro franjas verticales en la pantalla. ¡Tomá lápices de colores y pintá las franjas tal como lo haría Coloreame a partir de los bytes del punto anterior! Para descubrir los colores codificados, podés usar la herramienta disponible en <https://goo.gl/Z1z1hj>.



NOMBRE Y APELLIDO:

CURSO:

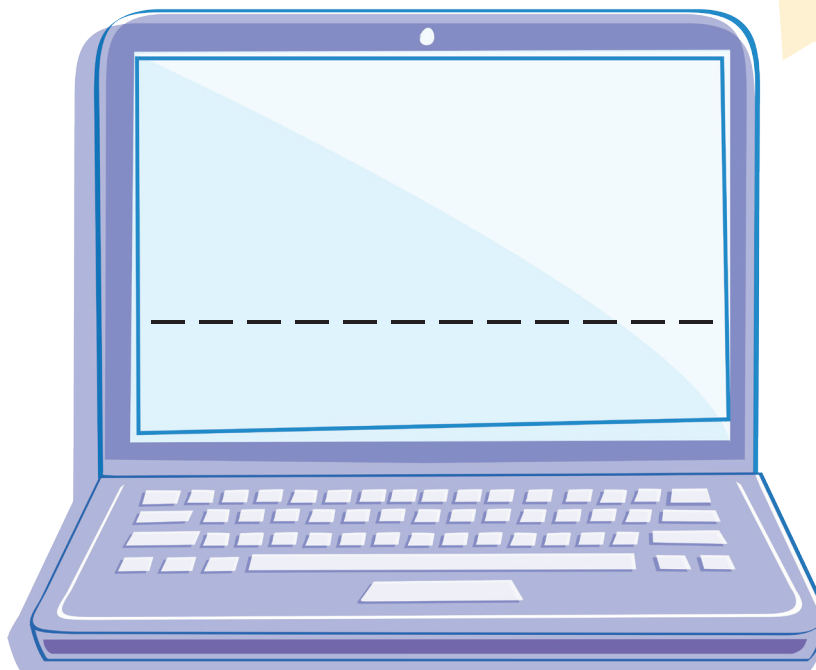
FECHA:

3. ¿Alguna vez escuchaste hablar de Textito? Es un programa que interpreta como texto los datos almacenados en la memoria y los muestra en la pantalla usando el sistema de codificación Letrados:

13	244	7	89	72	98	112	75	60	202	66	15	84	219	54
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
178	160	33	214	84	151	39	50	93	67	113	107	57	194	237
Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	,	.

Escribí en el monitor que aparece abajo el texto que mostraría Textito si leyese los mismos bytes que usaste con Coloreame.

¡ATENCIÓN!
En Letrados,
el 13 codifica al
espacio en blanco.



4. Entonces, la memoria, ¿puede entender de qué se trata la información que almacena?
Tachá lo que no corresponda.

SÍ

No

Actividad 2

La memoria rápida y la memoria grande



GRUPAL (4)

OBJETIVOS

- Diferenciar memorias volátiles de permanentes.
- Distinguir memorias pequeñas de grandes.
- Discriminar memorias rápidas de lentas.

MATERIALES



Papel



Lápiz

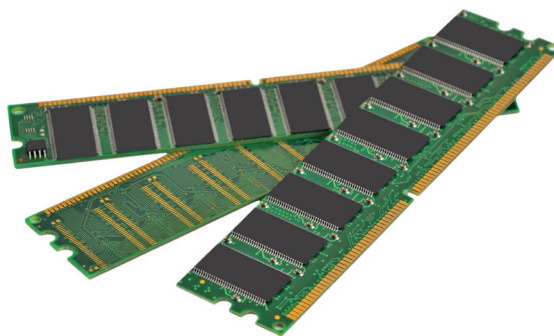


Ficha para estudiantes

DESARROLLO

Comenzamos la actividad preguntando a los estudiantes: “¿Qué pasa si estamos trabajando con una computadora de escritorio y súbitamente se interrumpe el suministro de energía eléctrica? Por ejemplo, si se corta la luz o si accidentalmente pasa corriendo el perro y se lleva por delante el cable”. Guiamos el intercambio con los estudiantes para llegar a la conclusión de que en esos casos, si trabajamos con programas fuera de línea¹, es probable que se pierda parte de la información. “Por ejemplo, si estuviésemos escribiendo en un procesador de texto, se perderían los cambios hechos en el documento desde la última vez que lo guardamos”.

Les explicamos a los estudiantes que, en esas situaciones, la información que se pierde se encuentra almacenada en la **memoria rápida**. Como representa valores con niveles de voltaje, para funcionar necesita energía. Sin electricidad, no preserva su contenido. Por este motivo se dice que es una **memoria volátil**. Usualmente se la conoce como memoria **RAM** (por el acrónimo del inglés de *Random Access Memory*).

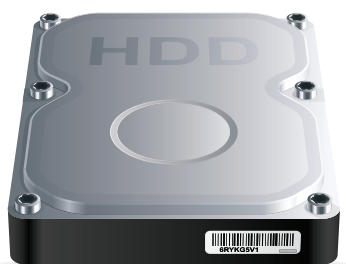


Módulos de memoria RAM

Siguiendo con el ejemplo, podemos preguntar: “¿Por qué no se perdieron los cambios previos a la última vez que guardamos el documento?”. Esto se debe a que se encontraban almacenados en la **memoria grande**. En contraposición a la memoria rápida, los datos que se guardan en ella se conservan aun cuando el componente deja de recibir energía. Por eso se dice que es una memoria persistente. Para llevar a los estudiantes a un terreno conocido, les comentamos que tanto los discos rígidos como las memorias USB pertenecen a esta clase de me-

¹Programas que no necesitan una conexión a Internet para funcionar.

morias. “Nunca enchufamos las memorias USB a la corriente eléctrica y, sin embargo, no perdemos la información que almacenan, ¿se dieron cuenta?”. Les contamos también que existen dispositivos que no cuentan con este tipo de memoria y que, por lo tanto, al desconectarse del suministro de energía, pierden toda la información que estaban utilizando.



Disco rígido



Memoria USB

Repartimos la ficha a los estudiantes y les comentamos que se trata de una tarea para el hogar. Tienen que completar una tabla indicando qué información guardan distintos dispositivos computacionales en cada una de las memorias, rápida y grande. Por ejemplo, si le quitamos la batería a un teléfono celular, ¿las aplicaciones abiertas continúan abiertas al volver a encenderlo? ¿Qué sucede si pusimos una foto como fondo de pantalla? ¿Sigue apareciendo la foto o muestra el fondo de pantalla que trajo de fábrica? Pueden recopilar información para completar la actividad desenchufando cables y quitando baterías. Es conveniente aclararles que no todos los artefactos de un mismo tipo administran de igual forma la información. Siguiendo con el ejemplo, podría haber teléfonos que recuerden las aplicaciones abiertas aun cuando les saquemos la batería, y otros que no lo hagan. A continuación se exhiben posibles respuestas.

DISPOSITIVO	SI DESENCHUFO O QUITO LA BATERÍA SIN PREVIO AVISO...	
	¿Qué olvida?	¿Qué recuerda?
Teléfono celular	Aplicaciones abiertas, mensaje que estaba mirando en la red social.	Todos los programas instalados, el fondo de pantalla.
Televisor inteligente	Canal que estaba viendo, nivel de volumen y brillo.	Canales sintonizados.
Computadoras de escritorio y portátiles	Posición del ratón, ventanas abiertas.	Programas instalados, páginas web visitadas, idioma del teclado.
Microondas	Potencia, tiempo de cocción.	Nada.
Consola de videojuegos	Posición en el juego.	Juego, mejores puntajes.

Continuamos preguntando: “¿Por qué creen que las memorias de la computadora se clasifican como *memoria rápida* y *memoria grande*? ¿Por qué no rápida y lenta o chica y grande?”. Esta clasificación se debe a que hay una relación inversa entre el tamaño de la memoria y el tiempo que nos demanda leer la información que almacena o ingresar en ella nuevos datos. A menor tamaño, mayor velocidad de acceso. A mayor tamaño, menor rapidez.¹

Para comprender la diferencia entre las velocidades de acceso, se puede usar la siguiente analogía: si la memoria RAM fuera un avión supersónico, un disco rígido sería una bicicleta a toda velocidad y una memoria USB tendría la velocidad de un caracol. Es decir, funcionan en escalas de velocidad totalmente diferentes.

También hay una relación directa entre la velocidad de la memoria y su precio. Cuanto más rápida, más cara; y cuanto más lenta, más barata. Es por eso que la memoria RAM tiene mucha menos capacidad que un disco rígido, aunque los precios sean parecidos.

CIERRE

Como cierre se puede destacar que, al trabajar en la nube (como, por ejemplo, al hacerlo con editores de texto en línea), no perdemos información cuando el dispositivo se queda sin suministro de energía. Esto se debe a que los datos no se encuentran almacenados en la memoria permanente de nuestras computadoras, sino en la memoria permanente de terceros. Como dice una campaña de la Fundación de Software Libre de Europa: “No hay computación en la nube, hay computadoras de otros”.

¹ La nomenclatura *rápida* y *grande* es la que se usaba con la computadora Ferranti Mercury de la década del sesenta.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

LA MEMORIA RÁPIDA Y LA MEMORIA GRANDE



Quando guardamos un documento, la computadora copia información de la memoria rápida a la grande. De este modo, podemos recuperar lo hecho incluso si se corta la luz. ¿Sabés por qué el ícono suele ser así? Porque imita la forma que tenían los viejos disquetes (o discos flexibles) que se usaban en las computadoras de los años ochenta. Estos discos eran transportables y permitían almacenar datos desde 79 KB hasta 240 MB!

Las computadoras, por lo general, tienen dos tipos de memoria: la **memoria rápida** y la **memoria grande**. Una de las principales diferencias es que la memoria rápida es **volátil**, lo que significa que la información que almacena se pierde cuando se apaga la computadora. En cambio, la memoria grande es **permanente**, es decir, conserva la información aunque deje de recibir energía eléctrica.

1. Investigá qué información se conserva y cuál se pierde al cortar el suministro de energía de los dispositivos computacionales que tenés en tu casa. A algunos los vas a tener que desenchufar y a otros, sacarles la batería.

2. ¡Escribí tus descubrimientos en la tabla!

DISPOSITIVO	SI DESENCHUFO O QUITO LA BATERÍA SIN PREVIO AVISO...	
	¿Qué olvida?	¿Qué recuerda?
Teléfono celular		
Televisor inteligente		
Computadoras de escritorio y portátiles		
Microondas		
Consola de videojuegos		

¿Sabías que hay una relación entre la volatilidad, el tamaño, la velocidad y el precio de las memorias?

TIPO DE MEMORIA	¿VOLÁTIL O PERSISTENTE?	TAMAÑO	VELOCIDAD	COSTO
Rápida	Volátil	Chica	Rápida	Cara
Grande	Persistente	Grande	Lenta	Barata



Secuencia Didáctica 2

EL FUNCIONAMIENTO INTERNO

Esta secuencia didáctica está compuesta por una sola actividad llamada “La máquina criptográfica”. En ella se presenta la arquitectura de una computadora imaginaria y se describe cada uno de sus componentes y sus respectivas funciones. Los componentes del *hardware* son la memoria rápida, la memoria grande, la unidad central de procesamiento y el puntero de instrucción.

Luego se simula su funcionamiento interno con un juego de rol, en el que van a participar distintos estudiantes del curso. Debido a la extensión de la actividad, puede hacer falta más de una clase para completarla.

.....

OBJETIVOS

- Exhibir el funcionamiento interno de una computadora.
- Mostrar que las computadoras solo ejecutan un conjunto de instrucciones muy simples.

.....



Actividad 1

La máquina criptográfica



GRUPAL (4)

OBJETIVOS

- Emular el funcionamiento de bajo nivel de una computadora.
- Presentar el funcionamiento de la unidad central de procesamiento.

MATERIALES



Tizas o marcador de pizarra



Borrador



Cartulina



Ficha para estudiantes

DESARROLLO

En esta actividad presentamos la máquina criptográfica, una computadora especialmente diseñada para poder cifrar mensajes. Mostramos cada uno de sus componentes y explicamos sus funciones. Además, observamos cómo se integran dentro de la arquitectura general de una computadora imaginaria.

Comenzamos la actividad repartiendo la ficha a los estudiantes. Allí se encuentra una descripción de la máquina criptográfica. Repasamos con los estudiantes todo lo referente a sus componentes y su funcionamiento. Además, les presentamos las reglas del juego de rol con el que simularán el comportamiento interno de esta particular computadora. Es importante que nos tomemos el tiempo necesario para que los estudiantes tengan un panorama claro sobre lo que harán. Sin un acuerdo en este sentido, difícilmente la actividad pueda llegar a buen puerto. A continuación, se ofrecen descripciones que pueden usarse como disparadores en el intercambio con los estudiantes.

EL HARDWARE DE LA MÁQUINA CRIPTOGRÁFICA

1. Memoria rápida

La máquina criptográfica tiene dos memorias: una rápida y una grande. La memoria rápida consiste en un único casillero y la representamos como un cuadrado dibujado en el pizarrón. Allí se puede escribir de a un símbolo a la vez. El alfabeto disponible incluye las letras del abecedario, los dígitos del 0 al 9, el espacio, y los signos de exclamación e interrogación, tanto de apertura como de cierre.



Memoria rápida de un casillero

A, B, C, D, E, F, G, H, I, J, K, L, M, N, Ñ, O, P,
Q, R, S, T, U, V, W, X, Y, Z, 0, 1, 2, 3, 4, 5, 6,
7, 8, 9, , !, ., ?,

Alfabeto de la máquina criptográfica

2. Memoria grande

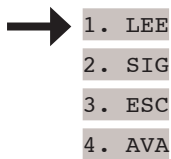
También podemos dibujarla en el pizarrón. Consiste en una tira de 10 celdas. En ellas se pueden guardar los mismos símbolos que en el casillero de la memoria rápida. Además, el componente tiene un cabezal que siempre se encuentra posicionado sobre una celda. Se puede leer el carácter almacenado en ella o escribir allí uno nuevo.



Memoria grande

3. Puntero de instrucción

En el pizarrón deberemos copiar el programa que ejecutará la máquina criptográfica. En esta máquina, un programa es una lista numerada de instrucciones (la descripción de las instrucciones se encuentra más adelante). Las escribiremos en el pizarrón, una debajo de la otra. La computadora cuenta con un puntero de instrucción, que representaremos con una flecha. Siempre apunta a una instrucción, que es la que la máquina tiene que ejecutar.



```
1. LEE
2. SIG
3. ESC
4. AVA
```

Puntero de instrucción

4. Unidad central de procesamiento

Por último, tenemos la unidad central de procesamiento (CPU), que dirige todo el funcionamiento. Es el componente que se encarga de ejecutar una a una las instrucciones de un programa. Para hacerlo, repite lo que se conoce como **ciclo de instrucción**: (i) lee la instrucción apuntada por el puntero de instrucción; (ii) la ejecuta; y (iii) mueve el puntero para que pase a apuntar a la próxima instrucción que el programa tiene que ejecutar. Estos ciclos se interrumpen solo cuando se ha completado una corrida del programa.

Dinámica de funcionamiento del juego

Para poner a funcionar la máquina criptográfica, hace falta que cuatro estudiantes pasen al frente. Cada uno de ellos interpretará un rol y recibirá uno o varios elementos específicos.

OBJETOS NECESARIOS PARA EL JUEGO

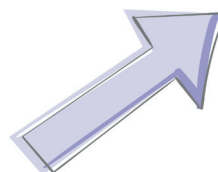
Recomendamos preparar previamente los siguientes elementos para que los estudiantes puedan interpretar cada uno de los roles:

**UNIDAD
CENTRAL
PROCESAMIENTO**



Rollo de cartulina a
modo de megáfono

**PUNTERO DE
INSTRUCCIÓN**

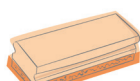


Flecha de cartulina
para apuntar
instrucción

MEMORIA RÁPIDA

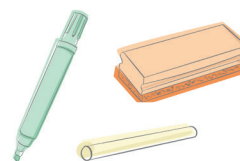


Marcador o tiza
(según tipo
de pizarrón)



Borrador

MEMORIA GRANDE



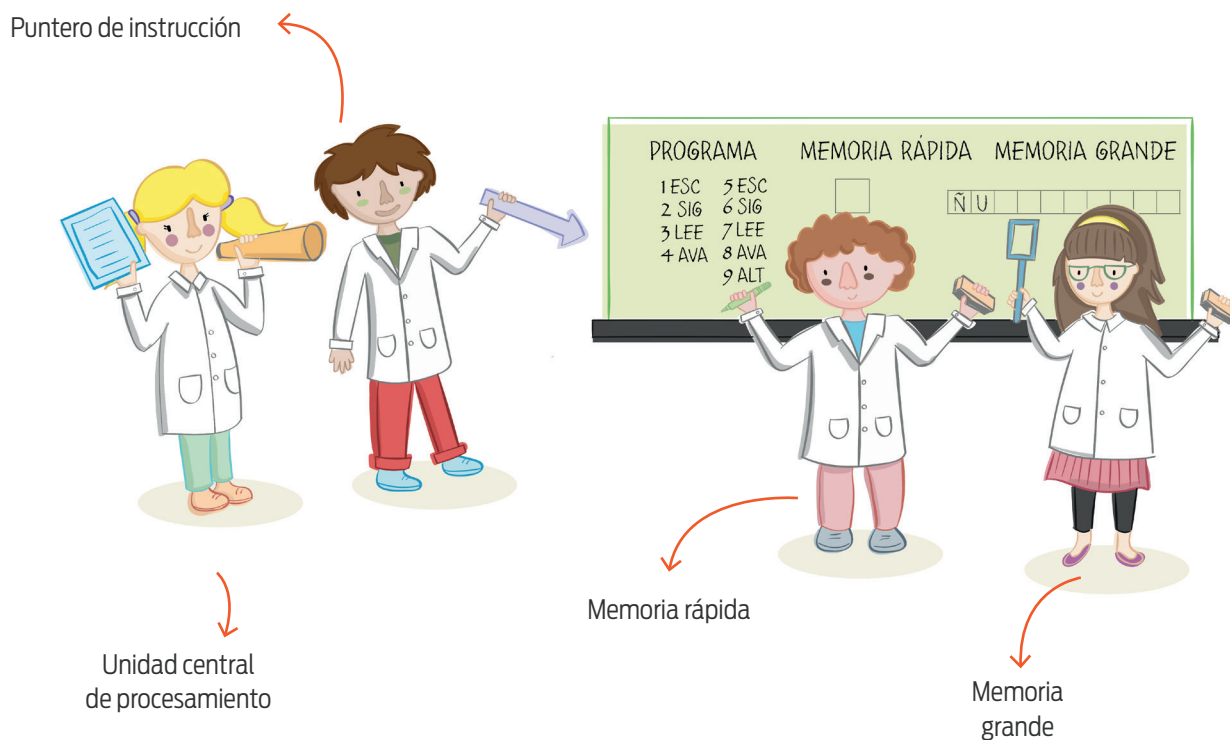
Marcador o tiza
+ borrador



Cabezal para indicar
celda (puede ser
una vara, un rollo de
cartulina, etc.)

Las tareas correspondientes al rol que desempeñará cada estudiante son las siguientes:

1. **Puntero de instrucción:** debe sostener la flecha que representa el puntero de instrucción. Al comenzar la ejecución de un programa, debe señalar la primera instrucción.
2. **Unidad central de procesamiento:** su función es ir haciendo avanzar sucesivas veces el ciclo de instrucción. Cada vez, debe (i) consultar al puntero de instrucción cuál es la instrucción del programa que tiene que ejecutarse; (ii) dar indicaciones al resto de los componentes para llevarla a cabo; y (iii) indicarle al puntero de instrucción a dónde apuntar, de forma tal de que quede señalando la próxima instrucción del programa que tiene que ejecutarse.
3. **Memoria rápida:** siguiendo las instrucciones que recibe de la unidad central de procesamiento, lee el contenido del casillero o escribe allí un nuevo símbolo.
4. **Memoria grande:** su tarea consiste en desplazar el cabezal y leer y escribir símbolos en las celdas. Siempre debe hacerlo siguiendo instrucciones impartidas por quien interprete el rol de la unidad central de procesamiento. Al comenzar una ejecución, el cabezal tiene que estar sobre la celda ubicada en el extremo izquierdo.



El conjunto de instrucciones

Estas son las instrucciones que pueden formar parte de los programas de la máquina criptográfica.

AVA

Mueve el cabezal de la memoria grande una posición hacia la derecha. Si el cabezal se encuentra en la última celda, vuelve a posicionarse sobre la primera.

ESC

Escribe en la memoria rápida el contenido de la celda de la memoria grande apuntada por el cabezal.

LEE

Lee el contenido de la memoria rápida y lo escribe en la celda de la memoria grande apuntada por el cabezal.

SIG

Reemplaza el símbolo de la memoria rápida por el siguiente, de acuerdo con el orden del alfabeto que usa la máquina. Al aplicarlo sobre el símbolo ?, el contenido pasa a ser A.

ANT

Reemplaza el símbolo de la memoria rápida por el anterior, de acuerdo con el orden del alfabeto que usa la máquina. Al aplicarlo sobre el símbolo A, el contenido pasa a ser ?.

SAL [nro de línea]

Hace apuntar al puntero de instrucción hacia la línea **nro de línea**, que es un parámetro de la instrucción.

SI MR = [valor] SAL [nro de línea]

Si el contenido de la memoria rápida es **valor**, hace apuntar al puntero de instrucción a la línea **nro de línea**.

ALT

Termina la ejecución del programa.

INSTRUCCIONES DE LOS LENGUAJES DE MÁQUINA

Los lenguajes de máquina provistos por computadoras reales están compuestos por unas pocas instrucciones, cada una de ellas de tres letras. Por tal motivo, en esta actividad usamos esa misma cantidad de letras para el nombre de cada instrucción.

Una vez que todo el grupo haya comprendido las características de la máquina criptográfica, pasamos a realizar las consignas de la ficha de la actividad. En ellas hay distintos programas que serán emulados por los grupos de estudiantes. Sugerimos que cada programa sea resuelto por un grupo distinto, de forma tal de que varios tengan la oportunidad tanto de interpretar a un componente de la máquina como de ser testigos oculares de la ejecución de un programa llevada a cabo por otros. Mirar la máquina desde afuera da una buena perspectiva sobre cómo funciona internamente una computadora.

Primera ejecución

La primera consigna presenta el programa Encripto 2. Al ejecutarse, modifica el contenido de las dos primeras posiciones de la memoria grande. Reemplaza lo almacenado en cada celda por el símbolo siguiente de acuerdo con el orden del alfabeto usado en la actividad. Por lo tanto, hace que la máquina criptográfica ejecute el algoritmo César estudiado en el capítulo 1, esta vez sobre mensajes de dos símbolos.

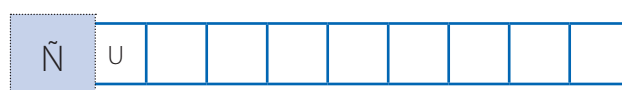
PARA REFRESCAR EL MÉTODO DE CIFRADO CÉSAR

Como vimos en la secuencia didáctica 3 del capítulo 1, “Ciudadanía digital y seguridad”, existe una técnica para enviar mensajes secretos llamada **cifrado César** o **cifrado de desplazamiento**. En esta técnica se sustituye una letra por otra que se encuentra más adelante en el alfabeto, según el desplazamiento elegido. Por ejemplo, usando un desplazamiento de una sola posición, la A se reemplaza por la B, la B por la C y así sucesivamente.

Copiamos en el pizarrón el siguiente programa:

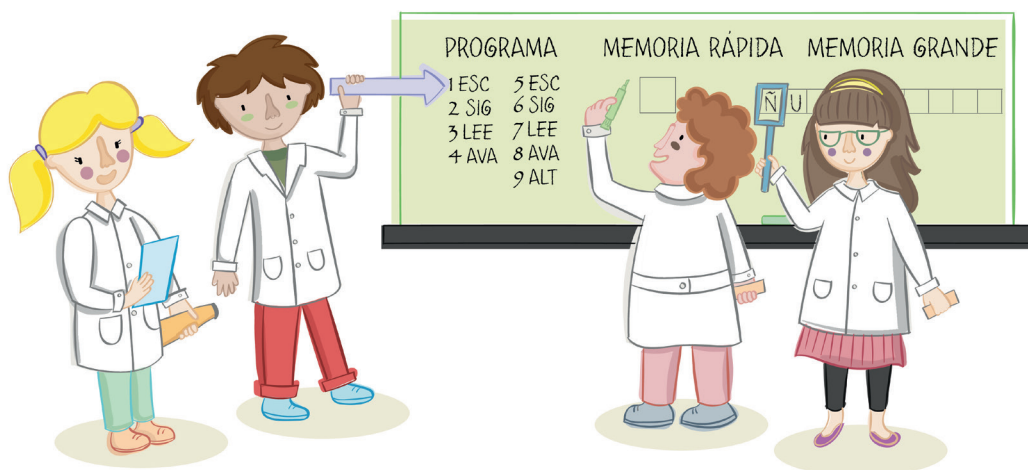
1 ESC	5 ESC
2 SIG	6 SIG
3 LEE	7 LEE
4 AVA	8 AVA
	9 ALT

Inicialmente, en la memoria grande se encuentra escrita la palabra ÑU seguida de ocho espacios en blanco. Por su parte, el cabezal se encuentra sobre la celda que almacena la Ñ.



Al finalizar una ejecución, el contenido original de la primera celda habrá sido reemplazado por el símbolo O y el de la segunda, por V. El resto se mantendrá inalterado.

Hacemos pasar a cuatro estudiantes al frente y le asignamos un rol a cada uno. El primero debe sostener la flecha del **puntero de instrucción**, que comenzará apuntando a la primera instrucción. Otro tiene que leer y escribir la **memoria rápida**, que al empezar contiene un espacio en blanco. El tercero debe representar a la **memoria grande** manipulando el cabezal, que al iniciar un programa se encuentra posicionado sobre la celda del extremo izquierdo. Además, a medida que avance el juego, irá leyendo y escribiendo celdas siguiendo instrucciones impartidas por quien asuma el rol de la **unidad central de procesamiento**. Es conveniente que este último tenga a mano la ficha con la descripción del lenguaje de la máquina criptográfica para chequear rápidamente qué es lo que hace cada instrucción.



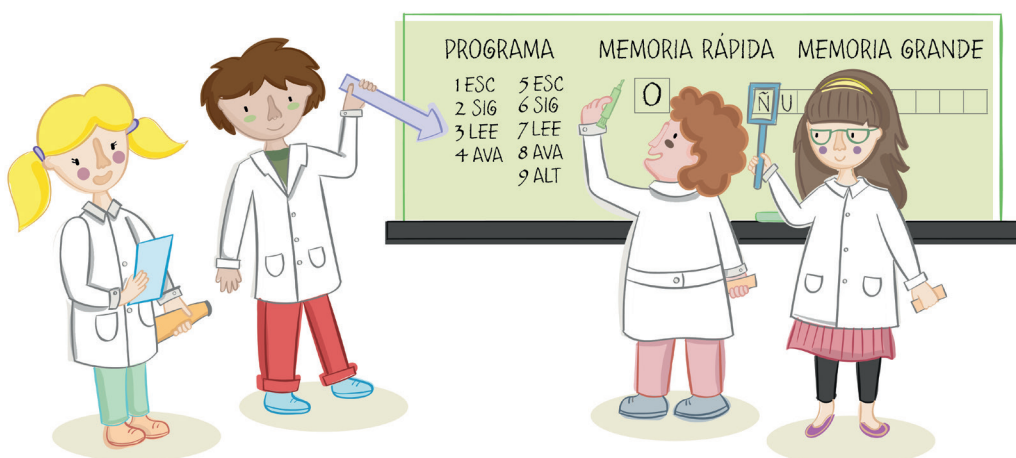
Estado inicial de la máquina criptográfica

Para dar comienzo a la ejecución del programa, el estudiante que interpreta a la unidad central de procesamiento le pregunta al puntero de instrucción qué instrucción del programa tiene que ejecutarse. En este caso, recibe como respuesta: “La instrucción actual es **ESC**”. A continuación, le pregunta a quien representa el rol de la memoria grande: “¿Qué hay en la celda bajo el cabezal?”. La memoria grande contesta: “La letra **Ñ**”. Luego, le dice a la memoria rápida: “Vos, escribí una **Ñ**.” El participante que actúa de memoria rápida escribe entonces el símbolo **Ñ** en la memoria monoceldaica. Finalmente, quien dirige la ejecución pide a quien interpreta al puntero de instrucción que señale la siguiente línea del programa. Con esto, completamos el primer ciclo de instrucción. O sea, llevamos a cabo la ejecución de la primera instrucción de Encripto 2.



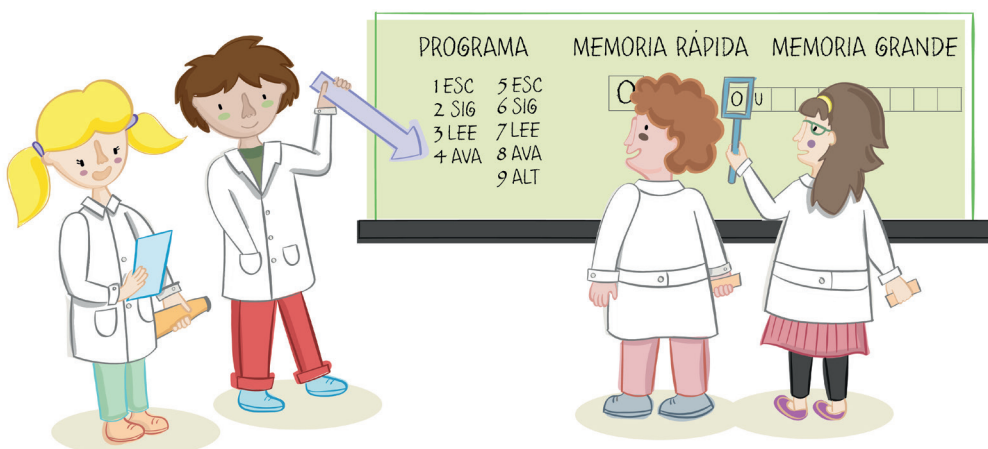
Estado de la máquina criptográfica luego de completar el primer ciclo de instrucción

El segundo ciclo de instrucción comienza con la unidad central de procesamiento preguntando a viva voz: “¿Cuál es la instrucción actual?”. El puntero de instrucción contesta: “La instrucción actual es **SIG**”. Entonces, le indica a la memoria rápida: “Pasá al siguiente símbolo”. Acto seguido, borra la Ñ y escribe una O. Entonces, el que interpreta a la CPU indica al puntero de instrucción que pase a apuntar a la siguiente instrucción, cosa que hace inmediatamente.



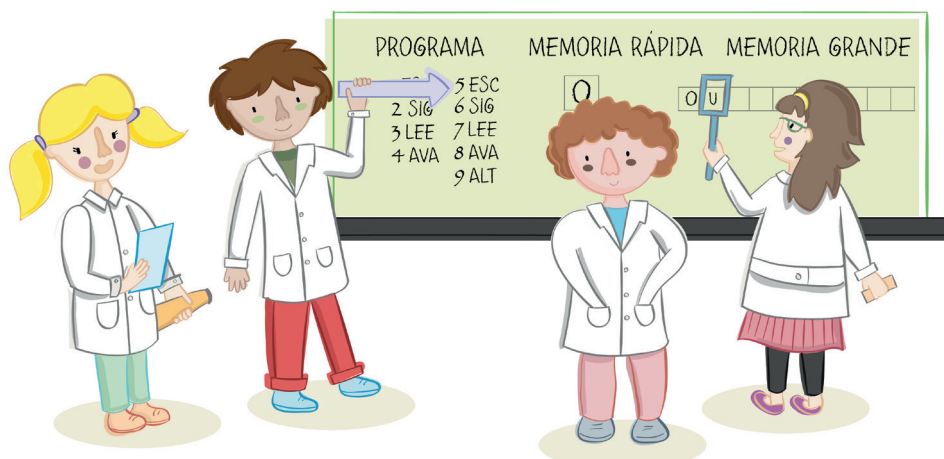
Estado de la máquina criptográfica luego de completar el segundo ciclo de instrucción

Al comenzar el tercer ciclo, la unidad central le consulta al puntero de instrucción: “¿Qué instrucción tenemos que ejecutar ahora?”. Este contesta: “La instrucción actual es **LEE**”. A continuación, le dice a la más pequeña de las memorias: “¿Qué hay allí?”, a lo que esta responde: “Una O”. Luego, le dice a la memoria grande: “Escribí una O”. La orden es obedecida inmediatamente. Finalmente, guía al puntero de instrucción para que avance un renglón.



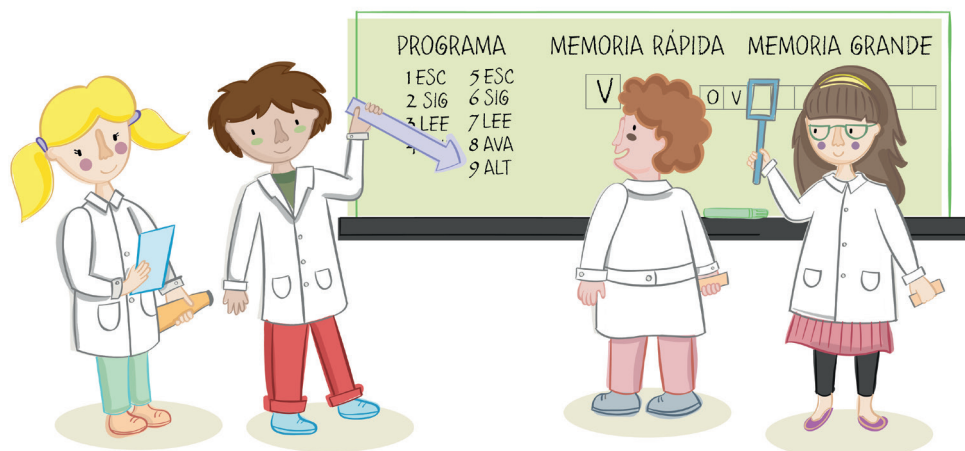
Estado de la máquina criptográfica luego de completar el tercer ciclo de instrucción

El cuarto ciclo de instrucción arranca con la CPU preguntando: “¿Cuál es la instrucción actual?”, a lo que el puntero de instrucción responde: “La instrucción actual es **AVA**”. Acto seguido, le dice a la memoria grande: “Avanzá el cabezal una posición”. El cabezal se mueve una celda hacia la derecha y queda apuntando a la que contiene la letra **U**. Como antes, esta vuelta finaliza indicando al puntero que señale la siguiente instrucción.



Estado de la máquina criptográfica luego de completar el cuarto ciclo de instrucción

Si observamos el programa con atención, podemos notar que las instrucciones entre las líneas 1 y 4 son las mismas que las que están entre las líneas 5 y 8. La máquina, entonces, reproducirá a continuación los mismos pasos ya hechos. Al finalizarlos, la segunda celda de la memoria grande contendrá el símbolo **V** y el cabezal estará sobre la celda de la tercera posición. Además, el puntero de instrucción señalará la instrucción **ALT**. Por lo tanto, la máquina se detendrá y habrá finalizado la ejecución del programa.



Estado de la máquina criptográfica al finalizar la ejecución del programa Encrypto 2

Reflexión intermedia

Una vez concluida la ejecución de Encripto 2, es el momento apropiado para reflexionar sobre lo ocurrido.

En primer lugar, podemos hacer notar a los estudiantes que la unidad central de procesamiento lo único que hace es repetir mecánicamente una serie de pasos hasta que la ejecución de un programa finaliza. A su manera, este componente también ejecuta un programa que viene directamente incorporado en él. Es decir, viene programado de fábrica para repetir una y otra vez un ciclo de instrucción.

A continuación, subrayamos que cada uno de los pasos que realiza la CPU es extremadamente simple. Aun así, pudimos ver que un programa escrito en lenguaje de máquina puede llevar a cabo una tarea que tiene un interés práctico: encriptar mensajes. En general, las computadoras proveen un lenguaje con el que se puede programar su comportamiento. Suele tratarse de lenguajes con instrucciones muy rudimentarias, que permiten establecer cómo se procesan los datos mientras viajan por los componentes de la máquina. Vulgarmente, se los denomina *lenguajes de bajo nivel*.

Resulta oportuno aprovechar la ocasión para tender un puente entre lo hecho durante el desarrollo de la actividad y lo ejercitado en capítulos anteriores. Hasta aquí, cada vez que construimos programas para ejecutar en la computadora, hemos usado Scratch. Considerando que las computadoras solo cuentan con un conjunto de instrucciones muy básicas, ¿cómo hacen realmente para ejecutar nuestras producciones?

Scratch, como casi todos los lenguajes de programación de uso corriente, pertenece al grupo de los llamados *lenguajes de alto nivel*. Son lenguajes considerablemente más expresivos que los que las computadoras traen de fábrica. Casi todos los lenguajes de alto nivel omiten por completo cómo se procesa y circula la información entre los componentes físicos de la computadora. Al escribir un programa usando cualquiera de ellos, se produce un proceso de traducción. Este proceso se llama *compilación* y lo lleva adelante un programa llamado *compilador*. Toma como entrada un programa escrito en un lenguaje de alto nivel y produce como salida un programa en lenguaje de máquina, cuyo comportamiento es idéntico al del programa recibido como entrada.

Existe un mecanismo alternativo al de la compilación. En este caso, contamos con un programa llamado *intérprete*, generalmente programado en el lenguaje de la máquina. Al igual que el compilador, recibe un programa como entrada. Sin embargo, en lugar de generar una salida, emula paso a paso la ejecución del programa recibido.

Segunda ejecución

La segunda consigna plantea hacer una ejecución de Encripto 4. Si bien es parecido a Encripto 2, no es idéntico. La secuencia de cuatro instrucciones que observamos que se repetían, en este programa aparece 4 veces.

1 ESC	5 ESC	9 ESC	13 ESC
2 SIG	6 SIG	10 SIG	14 SIG
3 LEE	7 LEE	11 LEE	15 LEE
4 AVA	8 AVA	12 AVA	16 AVA
			17 ALT

En este caso, una ejecución del programa aplicará el algoritmo César sobre cuatro caracteres. Al momento de arrancar, en la memoria grande se encuentra almacenada la palabra *PELO*. Al finalizar, *QFMP*.

Tercera ejecución

La novedad de la tercera consigna es que el programa propuesto incorpora instrucciones del lenguaje que inspeccionan condiciones y permiten provocar saltos. Al igual que al programar en Scratch, un programa de la máquina criptográfica puede evaluar una condición y, de acuerdo a si es cierta o no, hacer una u otra cosa. En esta máquina, lo que puede evaluarse es si el contenido de la memoria rápida es igual a un determinado símbolo, y en caso de que así sea, hacer apuntar al puntero de instrucción a una instrucción arbitraria; caso contrario, el puntero avanza a la siguiente instrucción del programa.

```

1 ESC
2 SI MR = ! SAL 7
3 SIG
4 LEE
5 AVA
6 SAL 1
7 ALT

```

El programa propuesto se llama Corriendo hasta la exclamación, e inicialmente, el contenido de la memoria grande es "¡TERMINAR!". Al correr, el programa aplicará el algoritmo César sobre todos los símbolos que aparezcan antes del símbolo *!*. Por lo tanto, al finalizar la corrida la memoria grande contendrá el texto "¡UFSNJÑBS!".

El programa comienza copiando el contenido de la memoria grande bajo el cabezal en el casillero de la memoria rápida. La siguiente es una instrucción condicional que pregunta si el contenido de la memoria rápida es el signo de exclamación de cierre. En caso afirmativo, salta hasta la última instrucción del programa, que detiene la máquina. En caso contrario, pasa a la siguiente instrucción. Las instrucciones entre las líneas 3 y 5 se ejecutan una tras otra y no presentan innovación alguna en relación con las de

ambas versiones de Encrypto: reemplazan el símbolo de la celda bajo el cabezal de la memoria grande por el siguiente, de acuerdo con el orden del alfabeto de la máquina.

La instrucción de la línea 6, una vez alcanzada, provoca que el puntero de instrucción pase a señalar la primera instrucción, de modo que hace comenzar todo otra vez. Este proceso se repetirá hasta que el cabezal quede posicionado sobre una celda que contenga el cierre de una exclamación. Además, este es el único modo de alcanzar la última instrucción del programa. Si tal símbolo no se encontrase en la memoria al comenzar, el programa no se detendría, se ejecutaría hasta el fin de los tiempos.

Reflexión de cierre

Combinando un salto condicional con uno incondicional conseguimos el mismo efecto que se produce con las instrucciones de un ciclo de cualquier lenguaje de alto nivel, como por ejemplo, Scratch. Al ejecutarse, el programa repite un bloque de instrucciones hasta que se alcanza una cierta condición.

Resulta interesante reflexionar sobre qué ocurriría si al comenzar la ejecución ninguna celda de la memoria grande tuviese impreso el símbolo !. Como vimos, en ese caso el programa no se detendría nunca. De acuerdo con la terminología de la teoría de la computabilidad, Corriendo hasta la exclamación calcula una función parcial. Es decir, una función que no está definida frente a ciertas entradas. En el plano de esta máquina, esto quiere decir que, al correr, si recibe ciertas entradas, se conseguirá encriptar un mensaje; y si recibe otras, la maquinaria no se detendrá nunca.

Desde la perspectiva del programa Corriendo hasta la exclamación, la entrada es el contenido de la memoria grande al comenzar una ejecución y la salida (en caso de existir) es lo que en esa memoria queda escrito al finalizar.

Desde la óptica de la máquina criptográfica, además del mensaje escrito en la memoria grande, el programa que tiene que emular también forma parte de la entrada. La salida, que existirá solo en el caso de que una corrida concluya, será el contenido de la memoria grande en el momento en que la máquina se detenga. Esta computadora puede, potencialmente, transformar cualquier mensaje de diez símbolos en cualquier otro de igual longitud. Es solo cuestión de construir el programa adecuado.

LA MÁQUINA CRIPTOGRÁFICA

¿Querés saber cómo funciona por dentro una computadora? ¡Lo vas a ver con tus propios ojos! Para arrancar, andá mirando el manual de la máquina criptográfica.

HARDWARE DE LA MÁQUINA CRIPTOGRÁFICA

La máquina criptográfica tiene dos memorias: una rápida y una grande. La **memoria rápida** consiste en un único casillero. Allí se puede escribir de a un símbolo por vez. El alfabeto disponible incluye las letras del abecedario, los dígitos del 0 al 9, el espacio y los signos de exclamación e interrogación, tanto de apertura como de cierre.

C

A, B, C, D, E, F, G, H, I, J, K, L, M, N, Ñ, O, P,
Q, R, S, T, U, V, W, X, Y, Z, 0, 1, 2, 3, 4, 5, 6,
7, 8, 9, , !, ., ¿, ?

Memoria rápida Alfabeto

La **memoria grande** consiste en una tira de 10 celdas. En cada una se pueden guardar los mismos símbolos que en el casillero de la memoria rápida. Además, el componente tiene un cabezal que siempre se encuentra posicionado sobre una celda. Se puede leer el carácter allí almacenado o escribir uno nuevo.

¿	C	H	O	R	I	Z	O	S	?
---	---	---	---	---	---	---	---	---	---

Memoria grande

En esta máquina, un programa es una lista numerada de instrucciones que se escriben una debajo de la otra (las instrucciones de la máquina criptográfica se encuentran más adelante). La computadora cuenta con un **puntero de instrucción**. Se trata de una flecha que siempre apunta a la instrucción que la máquina tiene que ejecutar.

→

1. LEE
2. SIG
3. ESC
4. AVA

Puntero de instrucción

Por último, tenemos la **unidad central de procesamiento** (CPU), que dirige todo el funcionamiento. Es el componente que se encarga de ejecutar una a una las instrucciones de un programa. Para hacerlo, repite lo que se conoce como **ciclo de instrucción**: (i) consulta al puntero de instrucción cuál es la instrucción del programa que tiene que ejecutarse; (ii) da indicaciones al resto de los componentes para llevarla a cabo; y (iii) le indica al puntero de instrucción a dónde debe apuntar, de forma tal de que quede señalando la próxima instrucción del programa que tiene que ejecutarse. Estos ciclos se interrumpen solo cuando se ha completado una corrida del programa.



EL CONJUNTO DE INSTRUCCIONES

Las instrucciones que pueden formar parte de los programas de la máquina criptográfica se describen a continuación.

AVA

Mueve el cabezal de la memoria grande una posición hacia la derecha. Si el cabezal se encuentra en la última celda, vuelve a posicionarse sobre la primera.

ESC

Escribe en la memoria rápida el contenido de la celda de la memoria grande apuntada por el cabezal.

LEE

Lee el contenido de la memoria rápida y lo escribe en la celda de la memoria grande apuntada por el cabezal.

SIG

Reemplaza el símbolo de la memoria rápida por el siguiente, de acuerdo con el orden del alfabeto que usa la máquina. Al aplicarlo sobre el símbolo ?, el contenido pasa a ser A.

ANT

Reemplaza el símbolo de la memoria rápida por el anterior, de acuerdo con el orden del alfabeto que usa la máquina. Al aplicarlo sobre el símbolo A, el contenido pasa a ser ?.

SAL [nro de línea]

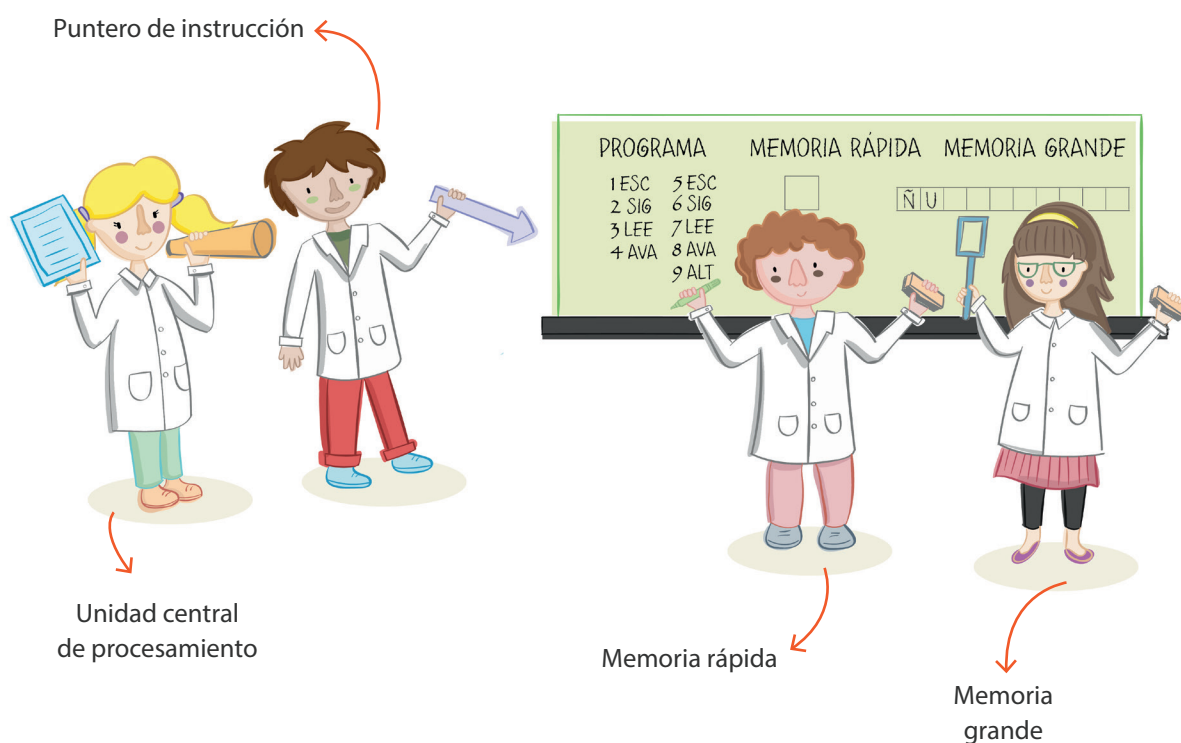
Hace apuntar al puntero de instrucción hacia la línea **nro de línea**, que es un parámetro de la instrucción.

SI MR = [valor] SAL [nro de línea]

Si el contenido de la memoria rápida es **valor**, hace apuntar al puntero de instrucción a la línea **nro de línea**.

ALT

Termina la ejecución del programa.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

Cuatro compañeros van a simular la ejecución de distintos programas sobre esta máquina. ¡Asegurate de ser alguno de ellos en alguna oportunidad!

1. Te presentamos el programa Encripto 2.

1 ESC	5 ESC
2 SIG	6 SIG
3 LEE	7 LEE
4 AVA	8 AVA
	9 ALT

Ñ	U								
---	---	--	--	--	--	--	--	--	--

¿Qué ocurre cuando corren el programa Encripto 2 si al empezar la memoria grande contiene el mensaje *ÑU*?

2. Mirá el programa Encripto 4 y observá qué sucede si al comenzar la ejecución la memoria grande tiene impresa la palabra *PELO*.

1 ESC	5 ESC	9 ESC	13 ESC
2 SIG	6 SIG	10 SIG	14 SIG
3 LEE	7 LEE	11 LEE	15 LEE
4 AVA	8 AVA	12 AVA	16 AVA
			17 ALT

P	E	L	O						
---	---	---	---	--	--	--	--	--	--

¿Qué quedó?

3. Observá atentamente el programa Corriendo hasta la exclamación y fijate qué sucede si al comenzar a ejecutarlo en la memoria grande dice *¡TERMINAR!*

1 ESC	3 SIG	5 AVA
2 SI MR = ! SAL 7	4 LEE	6 SAL 1
		7 ALT

¡	T	E	R	M	I	N	A	R	!
---	---	---	---	---	---	---	---	---	---

¿Qué texto quedó impreso? ¿Qué pasa si ejecutamos Corriendo hasta la exclamación sobre un mensaje que no contenga el símbolo *!*?

CONDICIONALES Y CICLOS

Si prestás atención, vas a ver que las instrucciones que provocan saltos permiten incorporar condicionales y ciclos a los programas de la máquina criptográfica.

GLOSARIO

abstracción. Proceso que consiste en reducir la complejidad de algo, poniendo el foco en la idea principal.

algoritmo. Serie ordenada de pasos para completar una tarea.

almacenamiento. Término que se usa para referirse tanto al proceso como al dispositivo para guardar datos digitales, de forma temporal o permanente.

autenticación. Verificación de la identidad de una persona o proceso.

bit. Acrónimo de *binary digit*, es decir, 'dígito binario'. Es un dígito del sistema de numeración binario que tiene solo dos valores, el 0 y el 1. Toda información digitalizada se representa con un conjunto de bits y por eso se lo conoce como la unidad mínima de información.

booleano. En honor al matemático británico George Boole (1815-1864). Tipo de dato o expresión con solo dos valores posibles: verdadero o falso.

bug. Error en un programa de *software*. Puede provocar que el programa finalice inesperadamente o funcione de una manera inesperada.

cadena de caracteres. También llamada *string*. Secuencia de letras, números u otros símbolos. Una cadena puede representar, por ejemplo, un nombre, una dirección o el título de una canción. Algunas funciones generalmente asociadas con cadenas son: longitud, concatenación y subcadena.

canal. En general, medio por el que se puede realizar una comunicación, por ejemplo, una llamada por teléfono celular. En particular, ondas electromagnéticas que se emiten y se reciben en una frecuencia determinada en esa comunicación.

celular. Dispositivo inalámbrico electrónico que se conecta a una red celular o está preparado para tener acceso a la telefonía celular. También se lo denomina (*teléfono*) *móvil*.

chip. Circuito electrónico en miniatura dentro de un encapsulado generalmente negro con terminales de conexión a su alrededor. Consta de transistores y otros elementos electrónicos activos y pasivos; llega a tener miles de millones de estos elementos.

ciberseguridad. Protección contra el acceso no autorizado a los recursos informáticos o su alteración, mediante el uso de tecnología, procesos y capacitación.

ciclo. Estructura de programación que repite una secuencia de instrucciones.

Ciencias de la Computación. Disciplina que estudia las computadoras y los procesos algorítmicos, incluyendo sus principios, los diseños de *hardware* y *software*, su implementación y su impacto en la sociedad.

cifrado. Procedimiento que aumenta la seguridad de datos electrónicos mediante la codificación del contenido, de manera que solo pueda leerlo la parte autorizada que cuente con la clave para decodificarlo.

ciudadanía digital. Normas de comportamiento apropiado y responsable con respecto al uso de la tecnología.

código. Conjunto de instrucciones expresadas en un lenguaje de programación.

computación. Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de computadoras. Actividad orientada a objetivos que requiere procesos algorítmicos, se beneficia de ellos o los crea.

computadora. Máquina o dispositivo físico que realiza procesos, cálculos y sigue instrucciones provistas por programas. Recibe, procesa y genera información. Son ejemplos de computadoras las PC, los *smartphones*, los chips de computadoras en otras máquinas.

concurrency. Propiedad de los sistemas de computación que permite que varios procesos se ejecuten simultáneamente.

condicional. Tipo de instrucción de un lenguaje de programación que permite realizar diferentes cálculos o acciones dependiendo de si una condición booleana que aparece en un programa es verdadera o falsa.

conectividad. Capacidad de un programa o dispositivo de conectarse con otros programas y dispositivos.

control, estructura de. En general, capacidad de dirigir el curso de las acciones. En programación, se llama así al uso de elementos de código de programación que dirigen las acciones que deben llevarse a cabo y el orden en que deben realizarse. Los condicionales y los ciclos son ejemplos de estructuras de control.

CPU. Sigla del inglés *central processing unit*, unidad central de procesamiento. Núcleo de la computadora, encargado de procesar las instrucciones de máquina, intercambiando información entre la memoria y los registros. También se lo denomina *microprocesador*.

datos. Información recopilada y usada como referencia o para hacer análisis. Los datos pueden ser digitales o no digitales, y pueden presentarse de muchas maneras, que incluyen números, texto, imágenes, sonidos o video.

debuggear, depurar. Ver *testear*.

descomponer. Desglosar un problema o un sistema en distintas partes.

digital. Característica de la tecnología electrónica que usa valores discretos, generalmente 0 y 1, para generar, almacenar y procesar datos.

disco rígido. Dispositivo magnético de almacenamiento permanente de datos. Por lo general, posee varios platos que giran a gran velocidad y un brazo con un cabezal que lee y escribe en la superficie de los platos, todo esto dentro de un contenedor hermético.

dispositivo de cómputo. Ver computadora.

ejecutar. Operar una instrucción o un conjunto de instrucciones (programa, aplicación, etc.).

encriptación. Ver *cifrado*.

entrada. Datos, señales u órdenes que recibe una computadora.

evento. Cualquier ocurrencia identificable que tenga un significado para el sistema, el *hardware* o el *software*. Los eventos generados por los usuarios incluyen pulsaciones de teclas o clics del ratón.

hardware. Conjunto de componentes físicos que conforman un sistema informático, una computadora o un dispositivo informático.

implementación. Proceso que consiste en expresar el diseño de una solución en un lenguaje de programación (código) que puede ejecutarse en un dispositivo de cómputo.

Internet. Conjunto mundial de redes de computadoras y sus conexiones, que utilizan protocolos comunes para comunicarse.

lenguaje de máquina. Conjunto de instrucciones que interpreta de manera directa el microprocesador o la CPU.

medio físico. Material de cualquier naturaleza que permite transmitir las señales que representan datos o bits y que luego se transforman en información. Por ejemplo, los cables de cobre por donde circulan los pulsos electromagnéticos, el aire por donde se transmiten las señales de radio, la fibra óptica por donde se transmiten pulsos de luz.

memoria flash. Almacenamiento permanente que utiliza tecnología electrónica de estado sólido sin partes móviles.

memoria volátil. Almacenamiento temporal utilizado por dispositivos informáticos.

módem. Acrónimo de **modulador-demodulador**. Conversor de señales digitales en señales analógicas que pueden transmitirse por el aire o por cable.

modularidad. Característica de un *software* que ha sido dividido en partes más pequeñas. Una aplicación puede tener varios procedimientos que son llamados desde el programa principal. Los procedimientos existentes pueden reutilizarse y recombinarse en una nueva aplicación.

offline. En español, 'fuera de línea'. Indica que un dispositivo, *software* o usuario está desconectado de Internet.

online. En español, 'en línea'. Indica que un dispositivo, *software* o usuario está conectado a Internet.

parámetro. Variable de un tipo especial que se usa en un procedimiento para referirse a la parte de los datos que se reciben como entradas.

pensamiento computacional. Habilidad humana para formular problemas de tal manera que las soluciones puedan representarse como pasos o algoritmos ejecutados por una computadora.

periférico. Unidad de *hardware* que provee una o más funciones de computación dentro de un sistema computacional. Puede proveer la entrada de datos a la computadora, aceptar la salida de datos o ambas.

píxel. Acrónimo del inglés *picture element*, 'elemento de imagen'. Superficie homogénea más pequeña que compone una imagen. Se define por su brillo y color. En las pantallas de las computadoras y en las impresoras, una imagen se muestra casi siempre mediante una cuadrícula de píxeles, cada uno ajustado al color requerido.

placa madre. Circuito principal de la computadora, que habitualmente contiene la CPU y la memoria RAM. También puede incorporar algunas interfaces a periféricos como discos duros, teclados, parlantes, etc.

plug-in. Programa informático que agrega a otro una función específica.

procedimiento. Módulo de código independiente que lleva a cabo una tarea concreta y es referenciado dentro de un cuerpo de código de programa más grande.

proceso de comunicación. Intercambio de datos entre un emisor y un receptor. El emisor transmite señales por un medio físico para enviar un mensaje y, cuando esas señales llegan al receptor, se interpreta el mensaje.

programa. Conjunto de instrucciones escritas en lenguaje de programación que la computadora ejecuta para lograr un objetivo particular, como el tratamiento de textos, el diseño de gráficos, la resolución de problemas matemáticos, el manejo de bancos de datos, entre otros.

protocolo. Conjunto especial de reglas que especifican el método para enviar y recibir datos entre los puntos que se comunican en una red.

punto de acceso wifi. Aparato que transforma una señal de red por cable en una señal de red inalámbrica wifi. Generalmente es una pequeña computadora con capacidad limitada de procesamiento y memoria.

RAM. Sigla del inglés *random access memory*, 'memoria de acceso aleatorio'. Memoria que permite que la CPU pueda calcular. Es una memoria no permanente o volátil que, si se interrumpe el suministro de energía, pierde su contenido.

red, red de dispositivos. Grupo de dispositivos informáticos (computadoras, teléfonos, servidores, conmutadores, *routers*, etc.) conectados por cables o inalámbricamente para el intercambio de información y recursos.

registros de CPU. Memoria pequeña y rápida dentro de la CPU que permite almacenar instrucciones y datos temporalmente mientras una instrucción se está ejecutando. Cada registro tiene un nombre particular para poder ser leído o escrito; ese nombre está formado típicamente por unas pocas letras en mayúscula, por ejemplo, registro RAX.

relé. Dispositivo electromecánico de conmutación que consta de un interruptor que cierra el circuito cuando se le suministra energía a un electroimán.

router. Dispositivo o *software* que determina la ruta de los paquetes de datos desde la fuente hasta el destino.

SoC. Sigla del inglés *system on a chip*, 'sistema en un chip'. Circuito integrado que contiene una o varias CPU, memoria volátil, memoria permanente y controladores de periféricos. Equivale a integrar una o varias CPU y una placa madre en un solo chip.

software. Conjunto de programas que se ejecutan en una computadora u otro dispositivo informático.

string. Ver *cadena de caracteres*.

testear. También *debuguear* o *depurar*. Proceso de encontrar y corregir errores (*bugs*) en los programas.

tipo de dato. Clasificación del dato según sus atributos y los tipos de operaciones que se pueden realizar con él. Algunos tipos comunes de datos son, por ejemplo, entero, cadena de caracteres (*string*), booleano (verdadero o falso) y punto flotante.

transistor. Dispositivo electrónico de conmutación de estado sólido sin partes móviles ni gases. En el funcionamiento de una computadora es como un relé o una válvula, pero mucho más confiable, de bajo consumo y tamaño muy reducido.

usuario final. También *usuario*. Persona para quien está diseñado el producto de *hardware* o *software*.

válvula. Dispositivo electrónico de conmutación, similar a una lámpara incandescente, que tiene un ánodo, un cátodo y una rejilla de control dentro de un bulbo lleno de gas. En el funcionamiento de una computadora, la válvula funciona como un relé, pero no tiene partes móviles y esto le permite trabajar más rápido y con menos fallas.

variable. Nombre simbólico que se utiliza para realizar el seguimiento de un valor que puede cambiar mientras se está ejecutando un programa. Una variable tiene un tipo de datos y está asociada con una ubicación de almacenamiento de datos.

web. Forma simplificada de referirse a la *World Wide Web* (WWW), 'red de alcance mundial'. Sistema de distribución de documentos accesibles vía Internet. Mediante un navegador se visualizan sitios web con textos, imágenes, videos u otros contenidos multimedia, y se navega a través de esas páginas usando hipervínculos.

wifi. Sistema de conexión inalámbrica entre dispositivos electrónicos, dentro de un área determinada; frecuentemente se usa para tener acceso a Internet.

FOTOS E ILUSTRACIONES TÉCNICAS

Shutterstock: cobalt88, Sergiy Kuzmin, Lek Changply, Orkidia, Parkheta, Suradech Prapairat, Abscent, Titov Nikolai, Unkas Photo, CapturePB, zentilia.

Wikipedia: John von Neumann (p21): <https://commons.wikimedia.org/wiki/File:JohnvonNeumann-LosAlamos.gif>

Computadora Clementina - 01: Biblioteca Digital / Programa de Historia de la Facultad de Ciencias Exactas y Naturales, UBA.

Program.AR, Fundación Sadosky

Av. Córdoba 832, 5º piso.

Ciudad Autónoma de Buenos Aires, Argentina.

Ciencias de la computación para el aula : 2do. ciclo de primaria : libro para docentes / Carlos Areces ... [et al.] ; contribuciones de Franco Frizzo ; Pablo Matías Factorovich ; Alfredo Olivero ; compilado por Luciana Benotti ; María Cecilia Martínez ; coordinación general de Hernán Czemerinski ; Vanina Klinkovich ; editado por Ignacio David Miller ; Alejandro Palermo ; editor literario Luz Luz María Rodríguez ; ilustrado por Luciano Andújar ... [et al.] ; prólogo de María Belén Bonello ; Fernando Pablo Schapachnik. - 1a ed. - Ciudad Autónoma de Buenos Aires : Fundación Sadosky, 2018.

Libro digital, PDF - (Ciencias de la Computación para el aula / Klinkovich, Vanina; Czemerinski, Hernán; 1)

Archivo Digital: descarga
ISBN 978-987-27416-5-5

1. Informática. I. Areces, Carlos II. Frizzo, Franco, colab. III. Factorovich, Pablo Matías, colab. IV. Olivero, Alfredo, colab. V. Benotti, Luciana, comp. VI. Martínez, María Cecilia, comp. VII. Czemerinski, Hernán, coord. VIII. Klinkovich, Vanina, coord. IX. Miller, Ignacio David, ed. X. Palermo, Alejandro, ed. XI. Luz María Rodríguez, Luz, ed. Lit. XII. Andújar, Luciano, ilus. XIII. Bonello, María Belén, prolog. XIV. Schapachnik, Fernando Pablo, prolog.

CDD 005.1

Queda hecho el depósito que dispone la Ley 11.723

Ediciones Colihue.

Primera edición: julio de 2018.



El contenido del manual se distribuye bajo la licencia
Creative Commons Compartir Igual.