

CIENCIAS DE LA
COMPUTACIÓN
PARA EL AULA

Fichas para estudiantes

2º CICLO SECUNDARIA

MANUAL 2º CICLO DE SECUNDARIA

AUTORES (por orden alfabético)

Claudia Banchoff
Hernán Czemerinski
Julián Dabbah
Franco Frizzo
Claudia Queiruga
Paula Venosa

COORDINADORA DEL EQUIPO AUTORAL

Claudia Queiruga

COORDINADORA PEDAGÓGICA

Silvina Justianovich

REVISORES DE CONTENIDOS

Julián Dabbah
Herman Schinca
Daniela Villani

EDITORES

Ignacio Miller
Alejandro Palermo

FOTÓGRAFO

Facundo Manini

DISEÑADORES GRÁFICOS

Jaqueline Schaab
Juan Martín Serrovalle

ILUSTRADORA

Klinko

COLABORADORES

Vanessa Aybar Rosales
Néstor Castro
Leandro Ariel Farkas
Fernando López
Sofía Martín
Isabel Miyuki Kimura
Sebastián Pacheco Véliz
Carlos Piazza Orlando

COLECCIÓN CIENCIAS DE LA COMPUTACIÓN PARA EL AULA

EDITORES GENERALES

Hernán Czemerinski
Vanina Klinkovich

SUPERVISOR DISCIPLINAR

Franco Frizzo

FUNDACIÓN DR. MANUEL SADOSKY

COORDINADORES DE LA INICIATIVA PROGRAM.AR

María Belén Bonello
Fernando Schapachnik

DIRECTOR EJECUTIVO

Esteban Feuerstein

PRESIDENTE

Ministro de Ciencia, Tecnología e Innovación, Roberto Salvarezza

Ciencias de la computación para el aula : 2do. ciclo de secundaria / Claudia Banchoff Tzancoff ... [et al.] ; contribuciones de Vanessa Aybar Rosales ... [et al.] ; compilado por Silvina Justianovich ; coordinación general de Vanina Klinkovich ; Hernán Czemerinski ; editado por Ignacio David Miller ; Alejandro Palermo ; fotografías de Facundo Manini ; ilustrado por Jaqueline Schaab ; Juan Martín Serrovalle ; Klinko ; prólogo de María Belén Bonello ; Fernando Pablo Schapachnik. - 1a edición para el profesor - Ciudad Autónoma de Buenos Aires : Fundación Sadosky, 2019. Libro digital, PDF - (Ciencias de la Computación para el aula / Klinkovich, Vanina ; Czemerinski, Hernán ; 4)

Archivo Digital: descarga
ISBN 978-987-27416-8-6

1. Informática. 2. Programación. 3. Educación Secundaria. I. Banchoff Tzancoff, Claudia. II. Aybar Rosales, Vanessa, colab. III. Justianovich, Silvina, comp. IV. Klinkovich, Vanina, coord. V. Czemerinski, Hernán, coord. VI. Miller, Ignacio David, ed. VII. Palermo, Alejandro, ed. VIII. Manini, Facundo, fot. IX. Schaab, Jaqueline, ilus. X. Serrovalle, Juan Martín, ilus. XI. Klinko, ilus. XII. Bonello, María Belén, prolog. XIII. Schapachnik, Fernando Pablo, prolog. CDD 004.0712

DISTRIBUCIÓN LIBRE Y GRATUITA

01

CIUDADANÍA
DIGITAL Y
SEGURIDAD

NOMBRE Y APELLIDO:

CURSO:

FECHA:

¡CUIDADO: RIESGO A LA VISTA!



El desembarco de las redes sociales en nuestras vidas ha modificado el modo de relacionarnos con el mundo. Por ejemplo, hasta hace poco, para mostrarle una foto a un amigo, primero teníamos que imprimirla, luego encontrarnos con él y, finalmente, mirarla juntos. Ahora es mucho más sencillo, ¿no? Este nuevo mundo, más virtual y menos personal, también modificó los riesgos a los que nos exponemos. Frente a nuevas reglas de juego, debemos *aggiornar* las precauciones para no exponernos a situaciones riesgosas.

1. Observá las siguientes publicaciones en redes sociales y respondé las preguntas a continuación.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

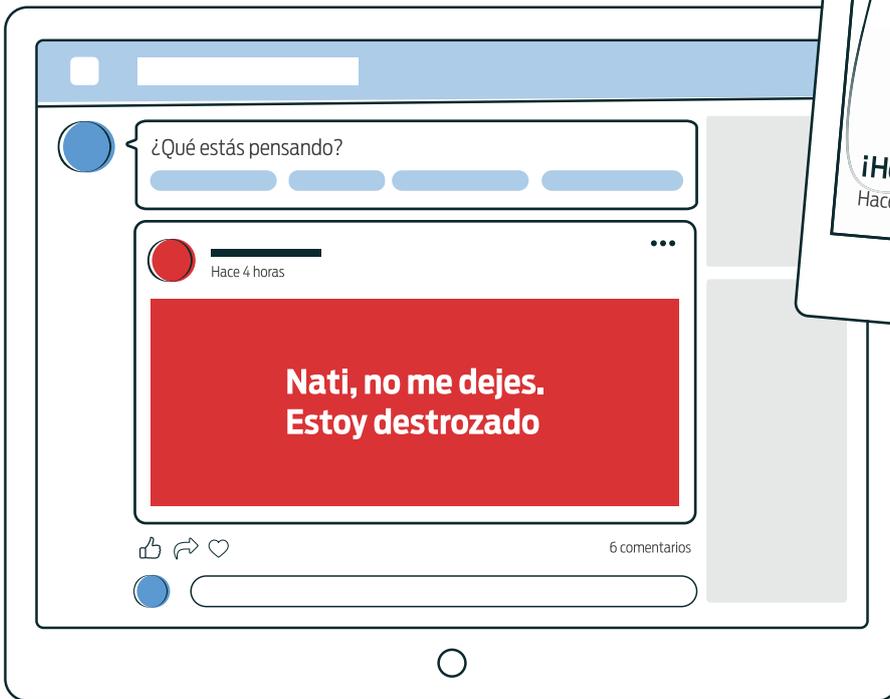


C

D



E



¿Qué tipo de información brinda cada imagen?

NOMBRE Y APELLIDO:

CURSO:

FECHA:

¿Cuáles son los riesgos que podrían originarse a partir de cada una de las situaciones de las imágenes?

En cada caso, ¿la información sensible fue compartida voluntariamente?

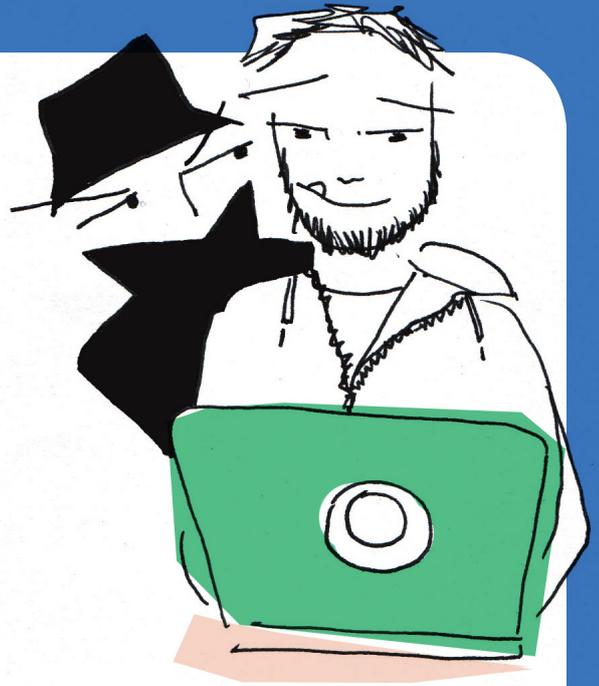
2. ¿Tenemos control sobre la información que publicamos en Internet? ¿Por qué?

NOMBRE Y APELLIDO:

CURSO:

FECHA:

LO QUE PUBLICO, ¿QUIÉNES LO VEN?



A veces es difícil imaginar hasta dónde puede llegar el contenido que publicamos en Internet. Por suerte, las redes sociales poseen configuraciones de privacidad que nos permiten decidir quiénes pueden ver nuestra información y quiénes no.

1. Contestá las siguientes preguntas, pensando en cada una de las redes sociales y aplicaciones que usás a diario. ¿Conocé la respuesta para todas ellas?

¿Quiénes pueden acceder a la información disponible en tu cuenta?

¿A quiénes aceptás como contactos, amigos, seguidores, etc.?

¿Resultás fácil de encontrar y reconocer? ¿Tu nombre y/o foto de perfil es visible para todos?

¿Quiénes pueden escribirte por chat?

NOMBRE Y APELLIDO:

CURSO:

FECHA:

¿Podés bloquear cuentas de desconocidos que intenten contactarte o compartan contenido que te resulta agresivo o desagradable?

¿Cuánta información pueden ver otras personas sobre vos? ¿Todos pueden ver lo mismo?

¿Alguna de las redes sociales que usás se vincula con otras cuentas que uses? ¿Publica automáticamente lo que subís en algún otro sitio? ¿Te pregunta cada vez?

¿Podés ser etiquetado en publicaciones ajenas? ¿Quiénes pueden compartir tus publicaciones?

NOMBRE Y APELLIDO:

CURSO:

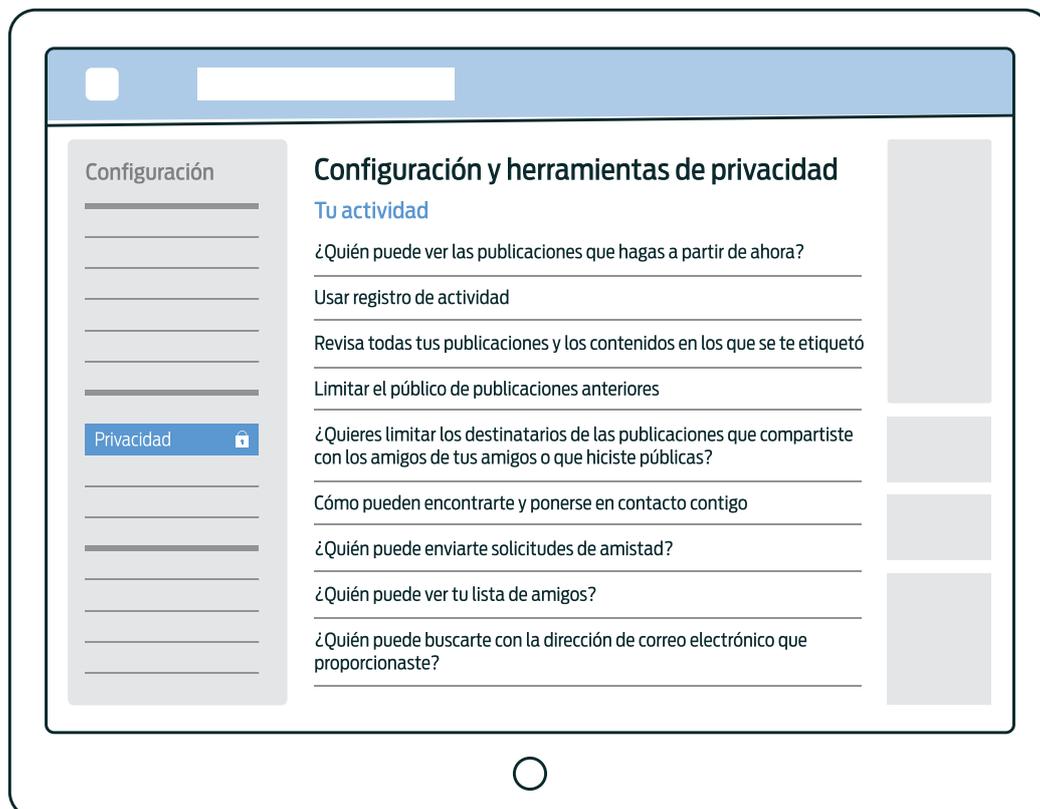
FECHA:

2. Ingresá a las redes sociales con una cuenta que no forme parte de tus contactos y, una vez adentro, buscá tu nombre y apellido. ¿Con qué te encontraste? ¿Sabías que lo que ves está disponible para cualquier persona? Completá la siguiente tabla con tus hallazgos sobre la privacidad de tus datos en las tres redes sociales que utilices con más frecuencia.

RED SOCIAL	¿QUÉ ENCONTRASTE?	¿SABÍAS QUE ESA INFORMACIÓN ERA ACCESIBLE PARA CUALQUIERA?	¿QUERÉS QUE ESA INFORMACIÓN SEA ACCESIBLE PARA DESCONOCIDOS?

DECIDÍ VOS SOBRE LO PRIVADO Y LO PÚBLICO

Es importante que seas consciente de qué información compartís y con quién. Especialmente, a cuál pueden acceder desconocidos. Si bien las redes sociales permiten configurar el nivel de privacidad de lo que publicás, los niveles por defecto no suelen ser los más restrictivos. Revisá la configuración de tus cuentas y asegurate de que el alcance de tu información no vaya más allá de lo que vos querés.

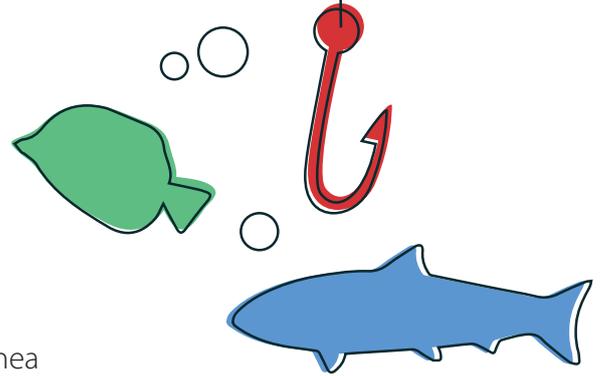


NOMBRE Y APELLIDO:

CURSO:

FECHA:

NO QUIERO SER UN PESCADO



Un pescador pone la carnada en el anzuelo y lanza la línea al agua. Luego, pacientemente esperará hasta que... ¡zaz! ¡El pobre animal mordió el anzuelo! Aunque suene difícil de creer, la pesca virtual existe, y también los pobres pescados.

1. Mirá la siguiente página.



¿Qué es esta página? ¿Para qué se usa?

2. Leé atentamente el siguiente artículo periodístico y respondé las preguntas que se presentan a continuación.¹

7 de agosto de 2017

25 de mayo: el pueblo al que le robaron \$ 3,5 millones con un aviso en Google

Parecía un día más. Un día alegre, en realidad, para este municipio de 40 mil habitantes. Porque en el partido de 25 de mayo, a 220 kilómetros de la Capital, después de mucha insistencia recibían una ambulancia para Norberto de la Riestra, una de las localidades de este partido bonaerense. [...]

La municipalidad

Aquel lunes 21 de noviembre de 2016, Roberto Testa, el tesorero del municipio, entró a trabajar a las 8 de la mañana en 25 de mayo, la ciudad que es cabecera del partido homónimo y que crece junto a la laguna Las Mulitas. A las 10, como todos los días, fue al Banco Provincia. Pidió que le imprimieran un resumen con todos los movimientos de las cuentas bancarias. [...] Ese día detectó, con su ojo prolijo de 32 años de carrera, que algo no andaba bien. "Empiezo y veo en el resumen que teníamos una transferencia por 100 mil pesos, otra por 90 mil. Me fui corriendo a lo del contador a comprobar por qué habíamos hecho esos movimientos", explica. [...]

El home banking

Paolo Salinas, el contador hace 18 años y encargado de emitir los pagos, giró su mirada hacia la puerta de la oficina para recibirlo. Escuchó a Testa y de inmediato se metió en las cuentas de *home banking* con su computadora. Salinas es el único que sabe las claves. Y no las anota en ningún lado, "por seguridad", dice. "Están en mi cabeza", acota. [...]

No recuerda específicamente el proceso de ese día, pero por lo general busca "Banco Provincia BIP" (Banca Internet Provincia) en Google, hace clic en el primer resultado, ingresa las claves y empieza a revisar las cuentas. "En ese momento me decía que había un usuario más adentro al mismo tiempo que yo", recuerda. Le preguntó a Romina Mancha, la subcontadora, si era ella. "No", le contestó Mancha desde el otro lado de la oficina, mientras dejaba su silla y se acercaba a la PC de Salinas. [...]

Mancha dejó el monitoreo y fue corriendo hasta la oficina de Ticera, la secretaria de Hacienda, quien a esa hora estaba reunida. No golpeó la puerta cerrada para entrar. "Me interrumpe y me dice: 'nos están robando, nos están robando' [...]", recuerda. [...] Fueron a buscar al Intendente: "Salimos disparadas". [...]

Tres millones y medio

El intendente, de 32 años [...], no salía de su asombro. Le pidió a Salinas que le mostrara lo que estaba sucediendo. Otra vez. Se dieron cuenta de que habían aparecido más transferencias, todas hechas a proveedores no habituales del municipio. "Cada segundo que pasaba perdíamos más plata", rememora. A esa altura, casi el mediodía de ese lunes fatídico, ya había 3 millones y medio menos de pesos en las arcas municipales [...]. Entonces, recibieron al menos una buena noticia: lograron que les bloquearan las cuentas. Eran poco más de las 12. [...]

¹ El artículo completo puede encontrarse en <http://bit.ly/2krwGH0>.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

Pesca con mediomundo

Juan Ignacio Bidone es fiscal de investigaciones complejas del Departamento Judicial de Mercedes. A esta altura, tiene muy claro lo que pasó. La secuencia del robo de 3 millones y medio pesos al municipio de 25 de mayo, dinero que luego fue extraído de diferentes cuentas, se realizó mediante *phishing*: una forma de engaño informático con la que se logra que un usuario revele información personal. Los ciberdelincuentes crearon un sitio falso similar al de la Banca Internet Provincia, también conocido como BIP por sus iniciales. Era idéntico al verdadero, pero con un detalle: la dirección, que obviamente no puede ser la original. Suplantaron la *a* por la *s*, para hacer más imperceptible el cambio, y montaron un sitio en la dirección bancsprovincia.bancsinternet.com.ar.



El resultado falso que aparecía en Google

Para lograr que alguien visitara ese sitio pensando que estaba entrando al Banco Provincia aplicaron una técnica llamada *black hat SEO*: una estrategia que desafía las reglas para lograr escalar posiciones en los listados de Google. En este caso, contrataron el servicio publicitario de *AdWords*; eso fue determinante, ya que lograron hacer que ante una búsqueda en Google de la frase "Banco Provincia BIP" el sitio falso que crearon apareciera como primer resultado. [...]

Cuando alguien entraba, en el sitio falso se le pedía el nombre de usuario y la clave para ingresar a las cuentas del banco. Capturaba la información y luego redirigía al verdadero sitio BIP para no despertar sospechas. [...] Así —estima la fiscalía— los delincuentes se hicieron de las contraseñas, la única credencial necesaria para realizar la operación. [...]

¿Quién o quiénes fueron las víctimas del ataque descrito en el artículo?

¿Qué información fue robada por el atacante?

NOMBRE Y APELLIDO:

CURSO:

FECHA:

¿Para qué fue usada la información robada? ¿Qué consecuencias tuvo el ataque para las víctimas?

¿Cómo se hizo para robar la información de las víctimas? ¿Qué nombre recibe este tipo de ataque?

¿Qué señales podrían haber alertado al contador de que estaba siendo engañado cuando ingresó al sitio web apócrifo?

Escribí otros ejemplos de información que podría ser robada mediante engaños similares.

¿Qué consejos pueden extraer de este artículo para evitar ser víctimas de un ataque como este?

Volvé a mirar la página web de la consigna 1. ¿Te parece que es auténtica?

¿QUÉ ES PHISHING?

La palabra *phishing* es una modificación del inglés *fishing*, que significa "pesca", pero reemplazando la letra *f* por la escritura *ph*, recurso que es común en el ambiente *hacker*. Se trata de un abuso informático en el que el perpetrador busca obtener información confidencial acerca de la víctima (por ejemplo, la contraseña de algún servicio o los detalles de su tarjeta de crédito) haciéndose pasar por otra persona, empresa, sitio o entidad en quien la víctima confía, a la manera de un cebo que induce al damnificado a morder el anzuelo y proporcionar la información. Posteriormente, la información robada puede usarse para fines espurios, tales como enviar mensajes a nombre de la otra persona o realizar compras por Internet.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

CONTRASEÑAS ¿SEGURAS?

¿Hay contraseñas más seguras que otras? ¿De qué depende su nivel de vulnerabilidad? Acá veremos algunas características de las contraseñas que te van ayudar a saber cuán (in)seguras son las tuyas.

1. A continuación hay una serie de contraseñas que una persona eligió para usar en distintos sitios de Internet. ¿Se te ocurre qué motivos lo llevaron a elegir las? Completá la tabla.



CONTRASEÑA	POSIBLE MOTIVO PARA HABERLA ELEGIDO
Luis2006	
25062006	
Coco	
D4l3B0c4	
Kpo de Temperley	
Ceci te amo	

¿Te parecen seguras estas contraseñas? ¿Por qué?

CONTRASEÑAS VULNERABLES

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 muy inseguras son: **contraseña, 1234, 1111, 123456, 12345678**. Si una es muy usada, es probable que sea insegura. En bit.ly/2jZvdY9 podés encontrar las 25 contraseñas más utilizadas durante 2018.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

2. Es momento de pensar qué factores contribuyen a la seguridad o vulnerabilidad de las contraseñas. Contestá las siguientes preguntas:

¿Cuántas contraseñas distintas de un dígito pueden existir? ¿Y de dos dígitos?

¿Cuántas contraseñas distintas existen si consideramos que pueden tener tanto un dígito como dos?

¿Y cuántas hay si consideramos que son de una letra del alfabeto castellano? ¿Y si son de dos letras?

¿Qué cantidad de posibles contraseñas hay si permitimos tanto dígitos como letras para contraseñas de longitud 1? ¿Y para contraseñas de longitud 2?

En general, si contamos con n símbolos, ¿cuántas combinaciones distintas hay fijando una longitud l ?

Calculá la cantidad de posibilidades que existe si se pueden usar dígitos, letras en mayúscula y letras en minúscula para contraseñas de 6 símbolos. ¿Son muchas? ¿Cuántas?

FUERZA BRUTA

Hay programas que, para descubrir contraseñas, se valen de la **fuerza bruta**: analizan todas las posibles contraseñas hasta llegar a descubrirla. Si las posibles son relativamente pocas, el programa llegará a probar todas las combinaciones en poco tiempo. ¡Ojo: hablar de millones es muy poco para un computadora! Además, ¿notaste que hay muchos sitios que bloquean el acceso a una cuenta cuando un usuario se equivoca en muchos intentos sucesivos al ingresar su contraseña? Así, evitan ser vulnerados por el uso de esta técnica.

¡ÁBRETE, SÉSAMO!

“Alí Babá y los cuarenta ladrones” es un cuento popular incluido en la célebre recopilación de cuentos árabes medievales *Las mil y una noches*. Alí Babá era un honrado leñador que, sin proponérselo, descubre a una banda de ladrones que esconden los tesoros robados en una cueva cuya boca queda sellada pero que, mágicamente, puede abrirse usando la contraseña “Ábrete, sésamo”. Entonces, decide usar la clave a espaldas de los malhechores, para ingresar y llevarse riquezas, de modo que se vuelve también él un saqueador.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

CLAVES COMPARTIDAS



La criptografía es un área de la matemática y la computación que se ocupa de desarrollar técnicas que permitan cifrar y descifrar mensajes de modo que, al enviarlos de un lado a otro, pueda preservarse la confidencialidad. Para garantizar que solo el emisor y el receptor comprendan los mensajes, hay técnicas que requieren que ambos compartan una clave (que nadie más conozca). Las técnicas con esta característica se llaman de **cifrado simétrico**.

Uno de los métodos simétricos más antiguos es el de cifrado por sustitución. Para cifrar un mensaje, hay que reemplazar cada letra por otra, siguiendo un criterio solo conocido por el emisor y el receptor. Luego, para descifrarlo, hay que hacer el reemplazo inverso.

A continuación se muestra un posible esquema de reemplazos:

ORIGINAL	A	B	C	D	E	F	G	H	I	J	K	L	M	N
CIFRADO	F	R	J	B	O	X	V	I	D	Z	K	C	W	Q

ORIGINAL	Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z
CIFRADO	D	S	M	U	A	N	T	G	L	H	Y	E	P

1. ¿Cómo se codifican los siguientes mensajes con el esquema de cifrado propuesto?

CIFRAR MENSAJES → _____

LUCES ESTROBOSCÓPICAS → _____

CUCARACHA → _____

TARTA DE CHOCLO → _____

2. ¡Te llegaron estos mensajes! ¿Qué dicen?

¿UGO MAOTOQBO GNTOB BO WD? → _____

NO CO ONJFMS CF TSATGVF → _____

NOMBRE Y APELLIDO:

CURSO:

FECHA:

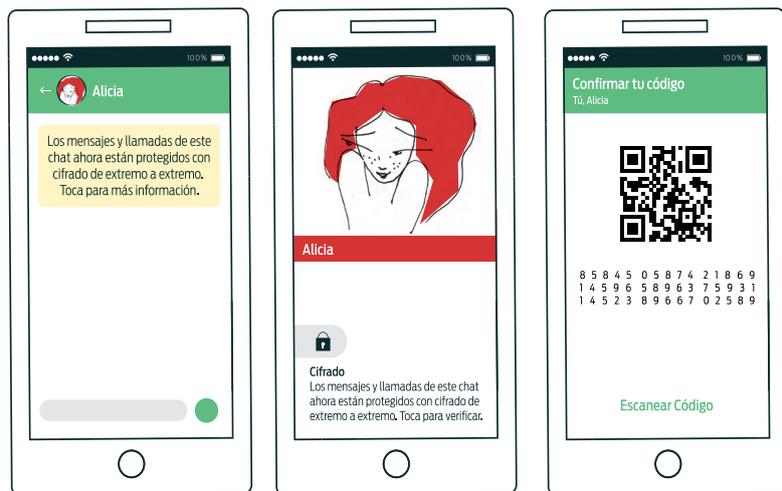
JSQ CFN WFQSN CDWMDFN → _____

CF MOCSTF QS BSRCF → _____

¿SABÍAS QUE...

... las aplicaciones de mensajería instantánea suelen cifrar los mensajes? Por ejemplo, la primera vez que enviamos un mensaje a un contacto, WhatsApp informa que el intercambio viaja cifrado de extremo a extremo, lo que significa que se cifran en el teléfono del emisor y solo pueden ser descifrados en el teléfono del destinatario.

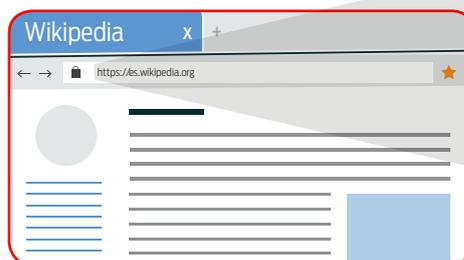
Entrando a la pantalla de opciones de cualquier contacto y seleccionando la opción "Cifrado", se pueden ver dos representaciones de la clave compartida con ese contacto: como código QR y como una serie de números.



SITIOS SEGUROS

Es sumamente importante que, antes de enviar información sensible a un sitio de Internet (contraseñas, mensajes privados, claves bancarias, etc.), verifiquemos que la comunicación con el sitio esté cifrada.

Cuando usamos un navegador de Internet podemos chequear si la comunicación es segura. En ese caso, en la barra de direcciones aparece un candado y la dirección debería comenzar con "https://", a diferencia de las no seguras, que comienzan con "http://".



02

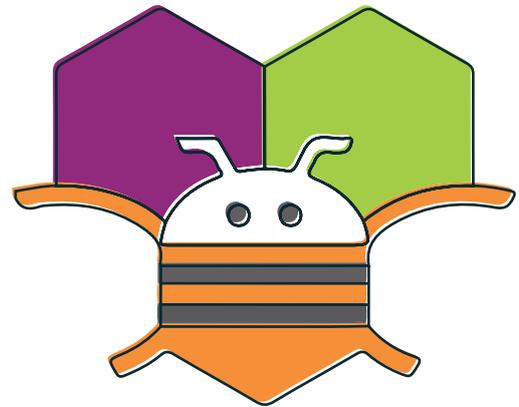
PROGRAMAS Y EVENTOS

NOMBRE Y APELLIDO:

CURSO:

FECHA:

PUESTA A PUNTO DE APP INVENTOR

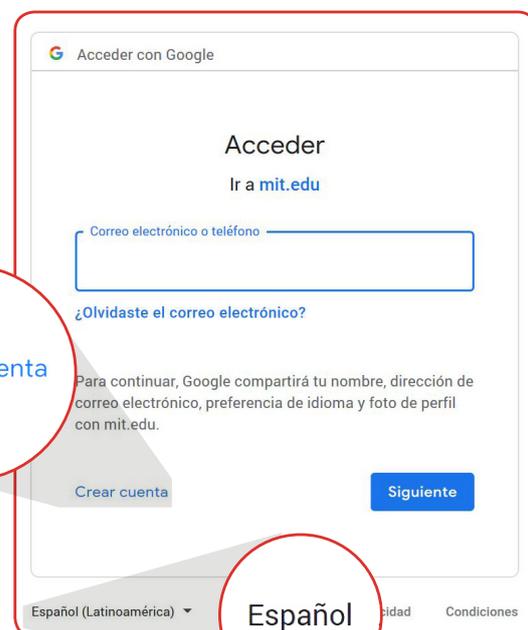


¡Vamos a aprender a usar App Inventor! MIT App Inventor 2 es un entorno de programación en el que podés desarrollar tus aplicaciones para teléfonos y tabletas. Para eso, vas a necesitar abrir una cuenta en el sitio en línea de App Inventor, que te va a permitir crear, guardar y compartir tus proyectos.

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

CREAMOS UNA CUENTA

1. Abrí un navegador web y cargá la dirección de MIT App Inventor 2: <http://ai2.appinventor.mit.edu>.
2. ¿La página está en inglés? Abajo a la izquierda hay un menú en el que podés seleccionar el idioma español.
3. Las cuentas de usuario de App Inventor están asociadas a una dirección de correo electrónico de Gmail. Si ya tenés una, podés usarla directamente. Si no, podés crear una haciendo clic en *Crear cuenta*.
4. Cuando inicies sesión por primera vez, vas a ver los términos y condiciones de uso de la aplicación. Al aceptarlos, te comprometés a hacerte cargo de tus producciones, a no generar contenidos inapropiados, a no infringir normas legales y a no hacer publicidad.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

5. Listo, ¡ya estás adentro! Lo que ves es la lista de proyectos creados con tu usuario. Como todavía no hiciste ninguno, la lista está vacía. Si la página está en inglés, podés cambiarla a español.



INSTALAMOS EL EMULADOR

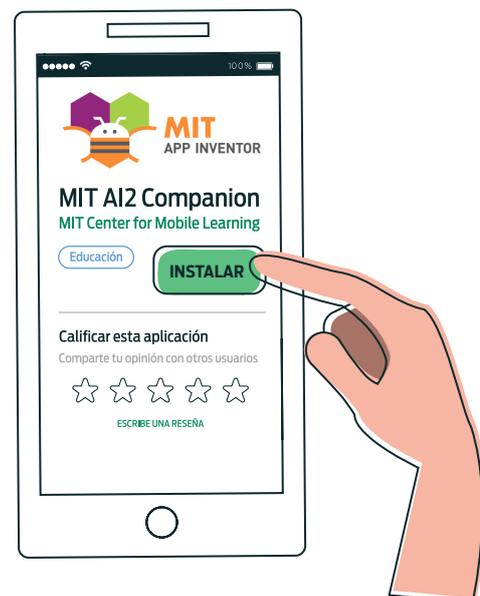
Si bien en principio las aplicaciones creadas con App Inventor son para ejecutarse en dispositivos que corran el sistema operativo Android, también puede instalarse un emulador de teléfonos con Android provisto por los desarrolladores de App Inventor, que permite verlas en funcionamiento directamente en la computadora. El emulador está disponible para Windows, GNU/Linux y Mac OS, y las instrucciones de instalación se encuentran en <http://bit.ly/2J6n0tW>.



INSTALAMOS LA APLICACIÓN PARA EJECUTAR EN TELÉFONOS

Para ejecutar programas creados en APP Inventor en un teléfono o tableta que corra el sistema operativo Android hay que tener instalado en el dispositivo la aplicación MIT AI2 Companion. Al igual que con cualquier otra aplicación, esto puede hacerse entrando en la tienda de aplicaciones Google Play y haciendo clic sobre INSTALAR.

¡Ahora ya tenés todo lo que hace falta para crear tus propias aplicaciones!



NOMBRE Y APELLIDO:

CURSO:

FECHA:

¡HOLA, AMIGO!

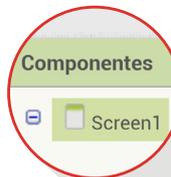
Saludar es, ciertamente, de buena educación. Usando App Inventor, vas a ocuparte de que tu teléfono sea muy pero muy cortés.

1. Creá un proyecto con tu usuario. Al ingresar, vas a encontrar una pantalla como esta.

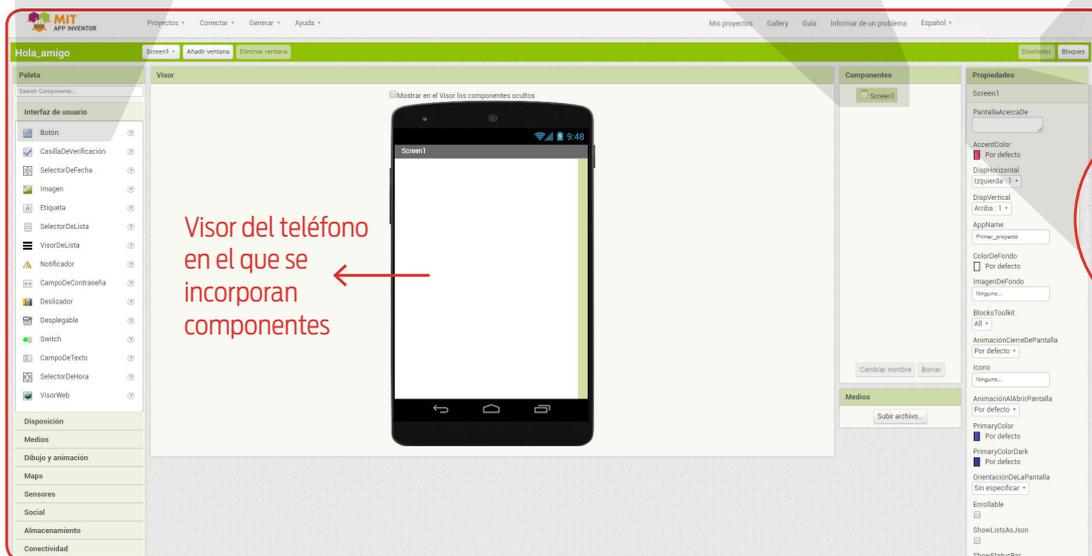


Paleta de componentes que se pueden incorporar a la aplicación

Listado de componentes incorporados a la aplicación



Botón para acceder al editor de bloques



Visor del teléfono en el que se incorporan componentes

Propiedades

Screen1

PantallaAcercaDe

Al seleccionar un componente de la aplicación, muestra sus propiedades (tamaño, color, etc.)

2. Agregá un botón al teléfono y ocupate de que la leyenda diga "Saludar". Después, ejecutá la aplicación en el emulador y hacé clic sobre el botón. ¿Ya saluda el teléfono? ¿Por qué?

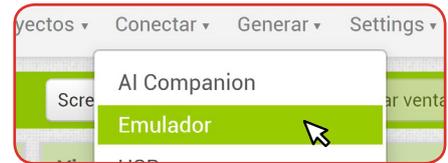
NOMBRE Y APELLIDO:

CURSO:

FECHA:

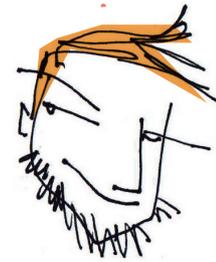
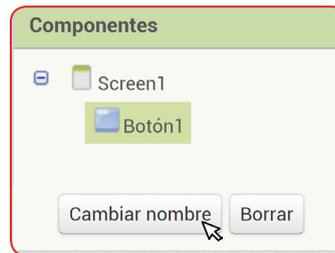
PARA VER LA APLICACIÓN EN LA COMPU

Para ejecutar la aplicación en la computadora tenés que ir al menú *Conectar* y seleccionar la opción *Emulador*.



¡CUIDADO CON LOS NOMBRES!

Hasta ahora agregaste solo un botón al que App Inventor le puso por defecto el nombre *Botón1*. Sin embargo, generalmente las aplicaciones van a tener muchos componentes y va a resultar indispensable poder reconocerlos por su nombre. En este caso podríamos llamarlo *BotónSaludar*. ¡Cambialo!

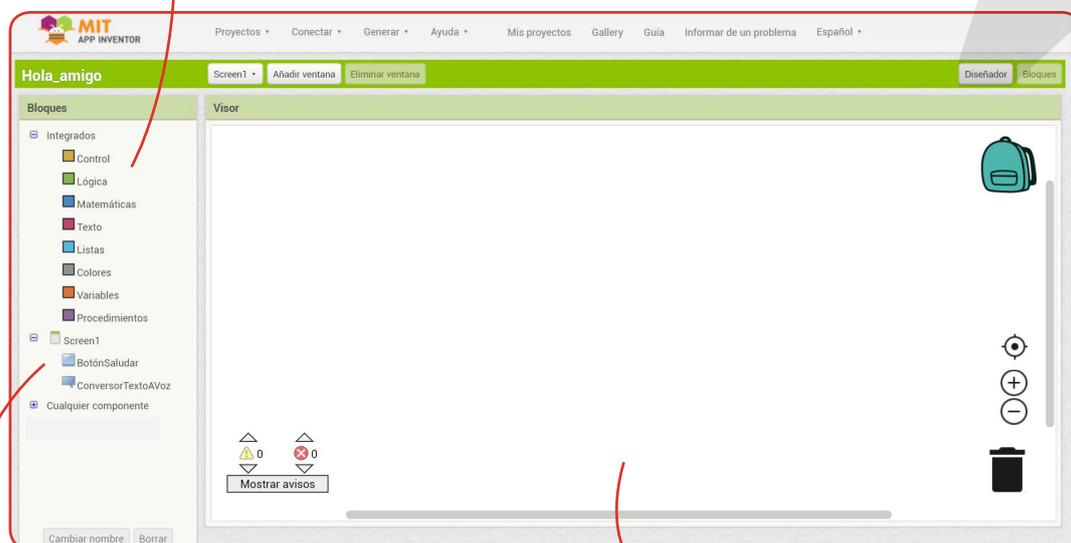


3. Para hacer que tu programa hable, tenés que agregar un componente más a la aplicación. Insepcioná los que están disponibles en *Paleta* y arrastrá el que te sirva sobre el visor del teléfono. ¿Cuál agregaste? ¿Lo ves en el visor? ¿Por qué?

4. Ingresá al editor de bloques. Vas a encontrarte con algo así.

Bloques disponible para todas las aplicaciones

Botón para volver al editor de diseño



Bloques específicos de los componentes de la aplicación

Espacio en el que se arman los programas arrastrando bloques

NOMBRE Y APELLIDO:

CURSO:

FECHA:

Usá los bloques que hagan falta para que la aplicación hable.



EVENTOS Y MANEJO DE EVENTOS

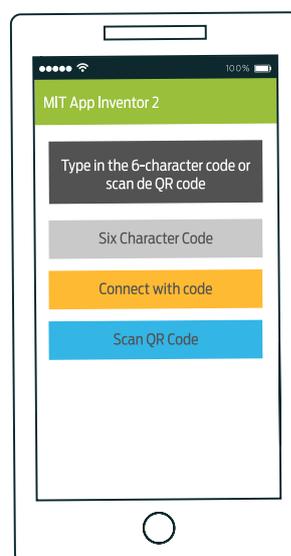
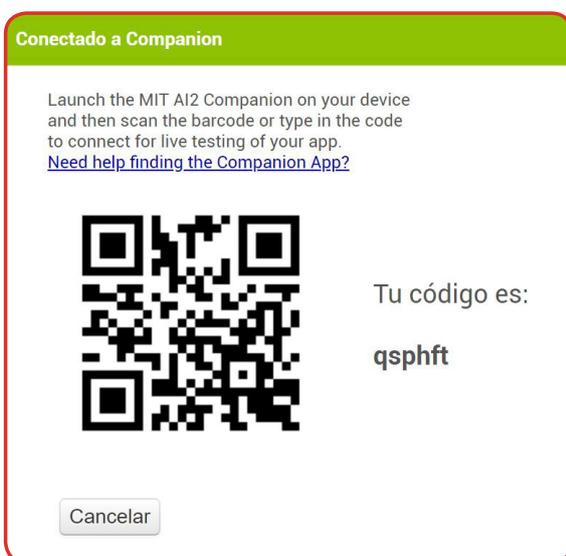
Hay dos elementos importantes que tenés que diferenciar: los **eventos** y el **manejo de eventos**.

Los eventos son sucesos que acontecen y que, *a priori*, no sabemos cuándo ocurrirán –por ejemplo, que el usuario presione el botón–. Por su parte, el manejo de eventos es el conjunto de acciones que se disparará cuando ocurra el evento –en este caso, que el teléfono diga “¡Hola, amigo!”–. Se trata, en definitiva, de la parte del programa que se ejecuta cuando ocurre un evento.

5. Ahora le tenés que agregar una nueva funcionalidad a la aplicación. Cuando alguien agite el teléfono, el programa tiene que decir “¡No te muevas así!”. Investigá los sensores que podés usar para reconocer el zarandeo. ¿Cómo lo resolviste?

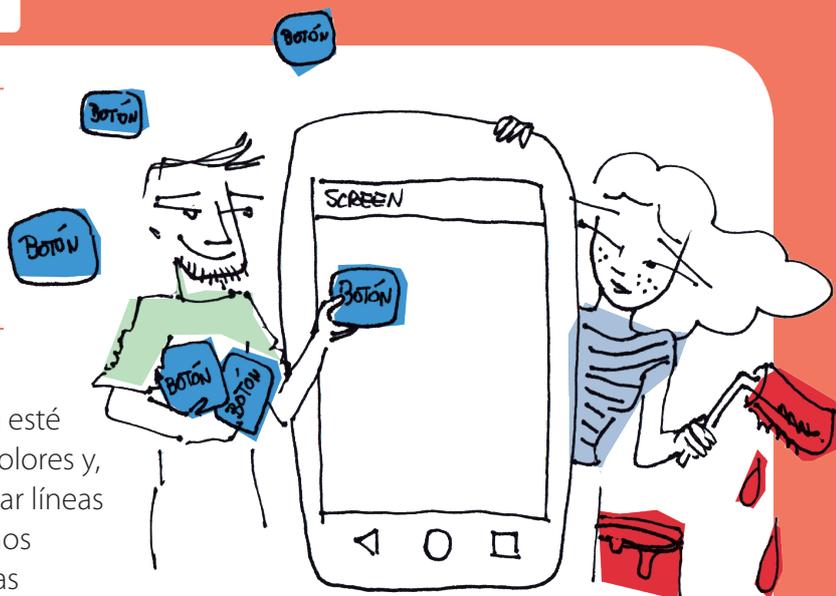
PARA EJECUTAR LA APLICACIÓN EN EL TELÉFONO

Para ejecutar la aplicación en teléfono, en primer lugar tenés que reiniciar la conexión para no continuar ejecutándola en el emulador. A continuación, en el menú *Conectar* seleccioná la opción *AI Companion*. Entonces, verás aparecer dos códigos, uno QR y otro alfanumérico de 6 caracteres. Al abrir *MIT AI2 Companion* en el teléfono, podrás cargar la aplicación usando cualquiera de los dos.



LA INTERFAZ DE DIBUJARTE

¡Vamos a diseñar la interfaz gráfica de una aplicación para dibujar! Cuando la aplicación esté terminada, con nuestros dedos elegiremos colores y, arrastrándolos por la pantalla, podremos trazar líneas y colorear figuras. Por ahora, nos encargaremos de su aspecto. No solo del contenido viven las aplicaciones, ¡también de la forma!

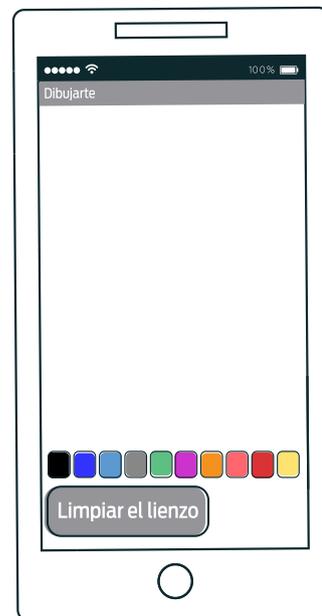


1. Mirá la interfaz de la aplicación e identificá sus componentes. ¿Qué tipo de componente son los cuadraditos de colores?

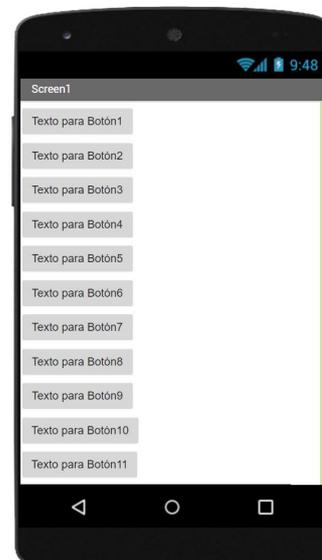


LAS APARIENCIAS ENGAÑAN

Los componentes suelen lucir de un modo pero, en general, ese aspecto es algo que podemos modificar. A los botones, por ejemplo, les podemos quitar el texto, cambiarles el color de fondo, asociarlos a una imagen, etc. ¡Inspeccioná el panel de propiedades y fijate qué posibilidades te da cada tipo de componente!



2. Crea un proyecto llamado *Dibujarte* y agregá al visor todos los componentes que hayas identificado. ¿Te muestra algo parecido a esto? Si es así, ¿por qué? ¿Cómo ubica App Inventor por defecto los componentes que agregamos a nuestras aplicaciones?

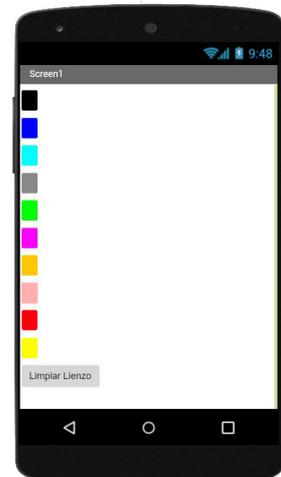


NOMBRE Y APELLIDO:

CURSO:

FECHA:

3. Cambiá el color de fondo de los botones y acomodá los textos. Deberías conseguir ver algo así. ¿Lo lograste? ¿Cómo lo hiciste?

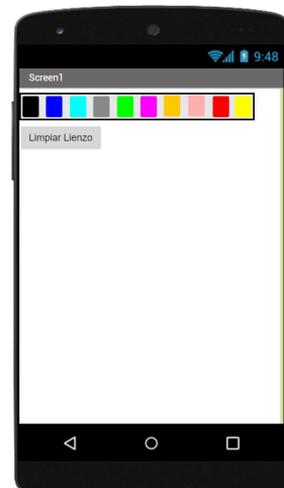


4. Los botones deberían aparecer uno al lado del otro. Por favor, ocupate de corregirlo. Deberías conseguir verlo como se muestra en la figura. ¿Para qué te parece que sirven los componentes que se encuentran en *Paleta > Disposición* listados a continuación?

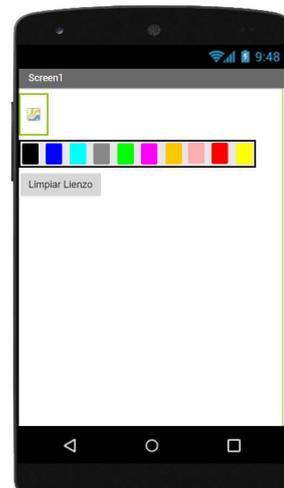
Disposición horizontal: _____

Disposición vertical: _____

Disposición tabular: _____



5. ¡Todavía falta el lienzo en el que vamos a dibujar! Buscalo en *Paleta > Dibujo y animación* y agregalo. ¿Qué cosas se pueden hacer con un lienzo?

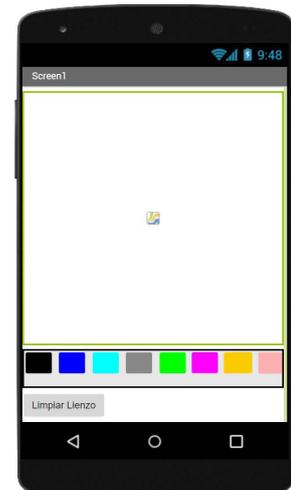


NOMBRE Y APELLIDO:

CURSO:

FECHA:

6. Ahora, ¡a ocuparse de las dimensiones! Fijate en las propiedades de cada componente que agregaste si podés jugar con los tamaños. Para ir viendo cómo queda, ejecutá la aplicación en el emulador. ¿Cómo hiciste para encontrar la medida justa de cada uno?

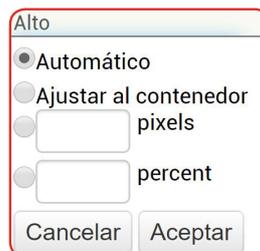


NO TODO LO QUE VES, ES

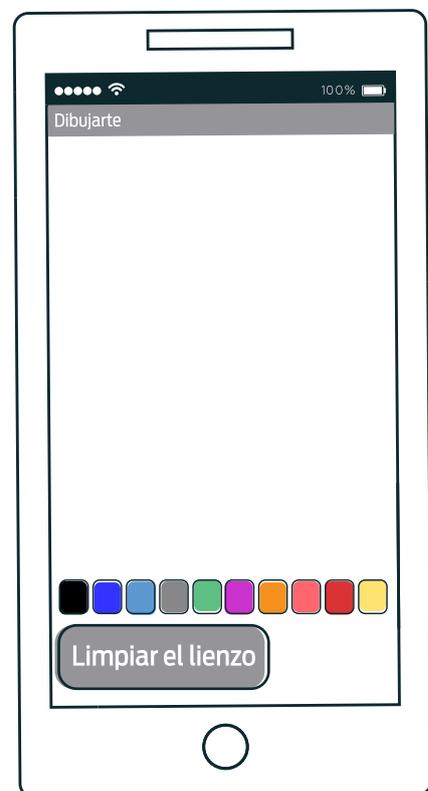
Si desaparecieron algunos botones, no te preocupes. Lo que muestra el visor de diseño no es exactamente lo que se va a ver cuando ejecutes la aplicación.

ALTO Y ANCHO

App Inventor presenta cuatro opciones para establecer el alto y el ancho de un componente visible: **Automático**, en el que la elección de dimensiones se delega en App Inventor; **Ajustar al contenedor**, que produce que el componente ocupe todo el espacio disponible; **[] pixels**, para especificar una cierta cantidad fija de píxeles; y **[] percent**, para indicar un porcentaje de la dimensión de la pantalla.



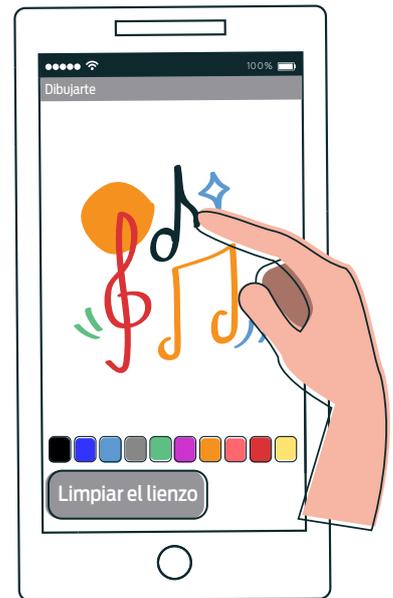
7. Últimos detalles: cambiá el título del componente *Screen1* y hacé que el botón para limpiar el lienzo tenga fondo gris y letras blancas y en negrita. Ahora sí, ejecutá la aplicación y fijate si se ve como esperamos.



DIBUJARTE EN ACCIÓN

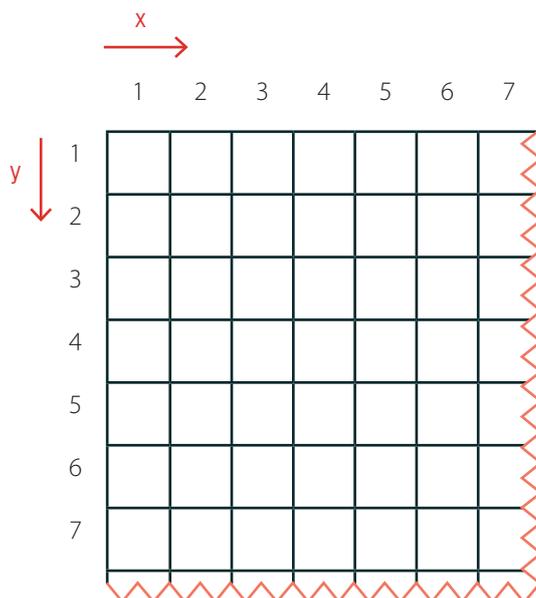
¡A encargarte de que *Dibujarte* nos permita dibujar! Seguí las consignas para poder jugar con puntos, líneas y colores, y volver a empezar.

1. Ocupate de que, cada vez que se presione con el dedo en algún lugar de lienzo, allí se dibuje un punto. Investigá el entorno en busca de cómo manejar los eventos que involucran a este componente. ¿Cómo hiciste para identificar dónde se presiona el lienzo?



LOS PUNTOS DE UN LIENZO

Un lienzo es un panel rectangular bidimensional sensible al tacto en el que se puede, entre otras cosas, hacer un dibujo. Cualquier ubicación en el lienzo puede identificarse con un par de valores (x, y) , donde x es la distancia al borde izquierdo del lienzo y y es la distancia al borde superior del lienzo –en ambos casos, medida en cantidad de píxeles–. Con $(1,1)$ se identifica el vértice superior izquierdo del lienzo.



NOMBRE Y APELLIDO:

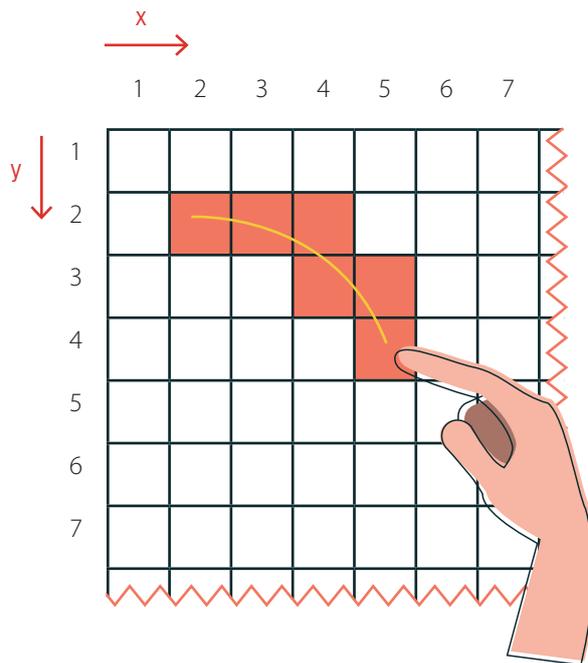
CURSO:

FECHA:

2. Ahora tenés que encargarte de que, cuando presionamos el lienzo con el dedo y lo arrastramos por la pantalla, se dibuje una línea que siga su recorrido hasta que lo levantemos.

ARRASTRAMOS EL DEDO SOBRE EL LIENZO

En App Inventor tenemos un bloque que nos permite manejar el evento que se produce al arrastrar el dedo por un lienzo: `cuando Lienzo.Arrastrado (XInicial) (YInicial) (XPrevio) (YPrevio) (XActual) (YActual) (draggedAnySprite) / ejecutar { }`. Tené en cuenta que la ejecución de los bloques que pongas dentro se disparará cada vez que muevas el dedo de un píxel a otro. Por ejemplo, en el caso del gráfico, se producirá 5 veces.



¿Qué indican los parámetros `(XInicial)` `(YInicial)` `(XPrevio)` `(YPrevio)` `(XActual)` y `(YActual)` ?

PARÁMETRO	XINICIAL	YINICIAL	XPREVIO	YPREVIO	XACTUAL	YACTUAL
INDICA						

3. Llegó el momento de dibujar en colores y poder borrar lo que hicimos si no nos gusta. ¡A ocuparte de los botones! Una pista: las propiedades de los componentes no solo las podés modificar desde el editor de diseño; también podés hacerlo programáticamente. ¿Qué propiedad te sirvió para poder dibujar con distintos colores? ¿De qué componente es esa propiedad?

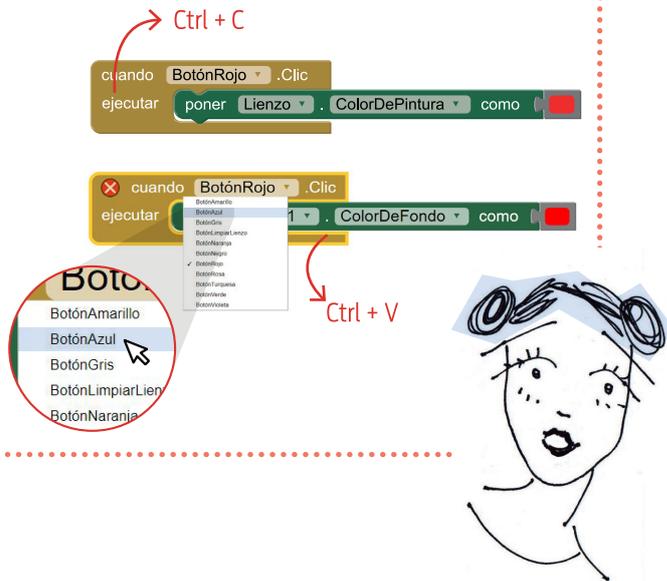
NOMBRE Y APELLIDO:

CURSO:

FECHA:

PARA COPIAR, PEGAR Y DESHACER

Una forma sencilla de copiar y pegar bloques es usar la tecla *CTRL* junto a las letras *C* (para copiar) y *V* (para pegar). Al nuevo conjunto de bloques se lo puede modificar de acuerdo a lo que se necesite. Además, para deshacer lo último que se hizo mientras se arma un programa, se puede presionar *CTRL + Z*.



4. La línea de dibujo está muy finita. Cambiale el ancho para que tus dibujos sobre el lienzo luzcan mejor aún. ¡Terminantemente prohibido hacerlo desde el editor de diseño! ¿Qué evento manejaiste para que, al cargar la aplicación, se defina el ancho de la línea? En general, ¿para qué puede ser útil manejar este evento?

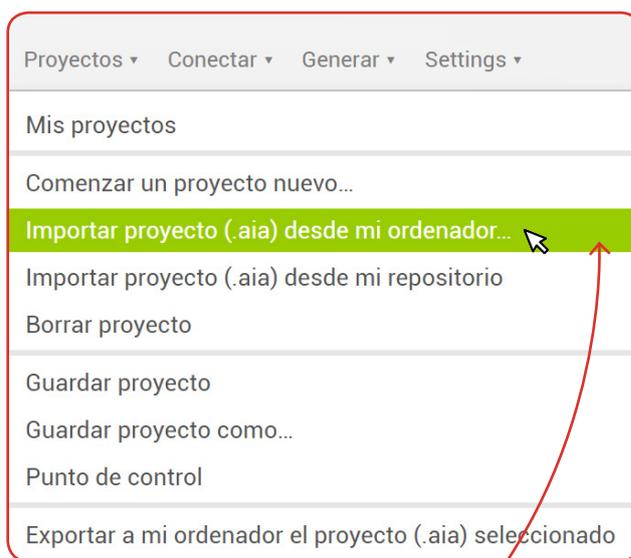
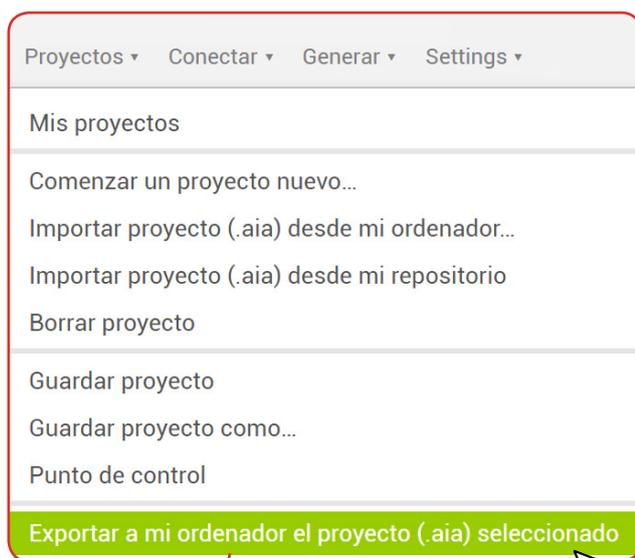
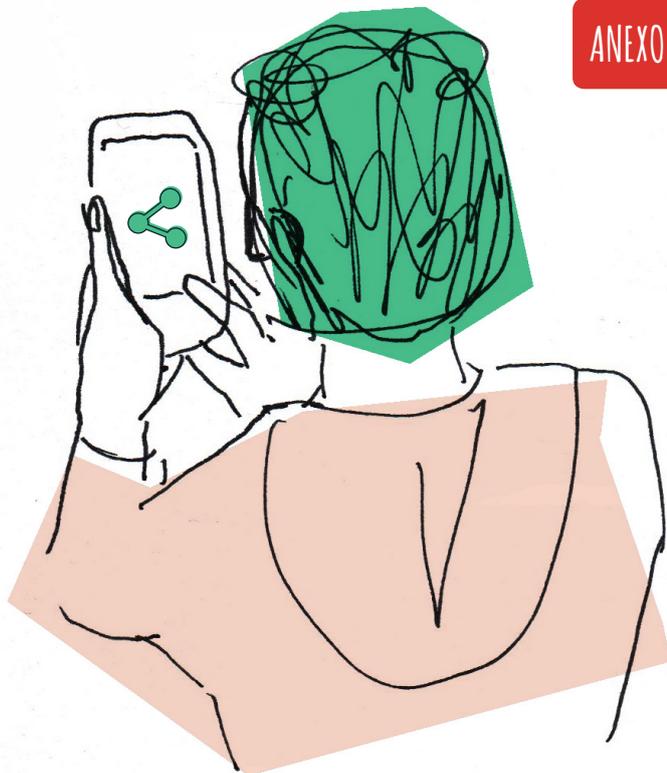
COMPARTIMOS APLICACIONES

ANEXO

Todas las aplicaciones que hagas las podés compartir con amigos, familiares, miembros de la comunidad de App Inventor y público en general. Acá te mostramos algunas formas de hacerlo.

A. Exportar e importar el proyecto

Para compartir el proyecto con otras personas que tengan una cuenta de usuario App Inventor, podés (i) exportar el proyecto, generando un archivo con extensión .aia; (ii) enviárselo a la otra persona (ya sea mediante correo electrónico, dándoselo en un *pendrive*, etc.); y (iii) indicarle a esta persona que lo importe. Una vez hecho esto, lo verá aparecer en la lista de proyectos de su usuario.



Se exporta un proyecto

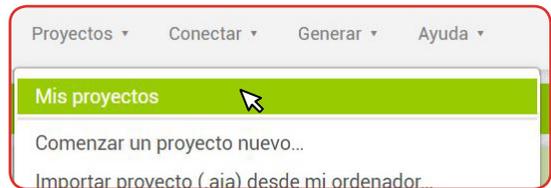


Se importa un proyecto

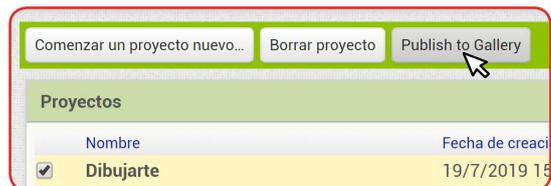
Se envía el archivo .aia

B. Publicar en la galería de App Inventor

La galería de MIT App Inventor es el sitio principal para compartir aplicaciones construidas con App Inventor y acceder a aplicaciones desarrolladas por otros. Todo lo que se publica en la galería es de código abierto: no solo se pueden usar las aplicaciones publicadas sino que también se puede acceder a su interior –es decir, a su diseño y a sus bloques–.



Para publicar una aplicación en la galería, en primer lugar hay que acceder a la lista de nuestros proyectos y, luego, seleccionar el proyecto que queremos compartir y hacer clic en el botón *Publish to Gallery*.



Entonces aparecerá un formulario que permite complementar la publicación con una foto, una descripción, un enlace a un video que muestre la aplicación en funcionamiento, etc. Para completar la publicación, solo hay que hacer clic en *PUBLISH*.

Cargá un ícono para la aplicación

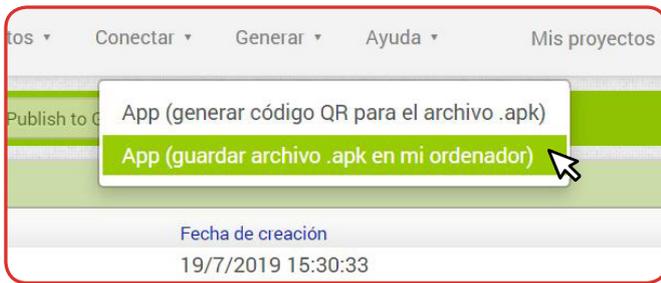
Nombre de la aplicación

Si hiciste un tutorial, poné la URL

Si el proyecto está basado en otros proyectos de App Inventor, referencialos acá

Descripción de la aplicación

A screenshot of the MIT App Inventor 'Publish to Gallery' form. The form includes a section for uploading a project image, a 'PUBLISH' button, and a 'CANCEL' button. Below this is a text area for the app name, which contains 'Dibujarte' and 'programar2020'. There are also fields for 'Created Date' (2019/08/01) and 'Changed Date' (2019/08/01). A text area for a tutorial URL contains the text: 'If this app has a tutorial or video, please enter the URL here.' Another text area for crediting other apps contains: 'Are you remixing code from other apps? Credit them here.' A final text area for the app description contains: 'Please write the description of the app here.' Red arrows point from the annotations to the corresponding fields in the form.



PERMISOS DE INSTALACIÓN

Para instalar una aplicación que no provenga de la tienda oficial *Google Play* en un teléfono o una tableta Android, es necesario habilitar ciertos permisos. En general, esto puede hacerse editando las opciones de seguridad del dispositivo.



C. Empaquetar la aplicación

A diferencia de las otras opciones, si empaquetás la aplicación vas a poder instalarla en cualquier dispositivo que corra el sistema operativo Android.

En primer lugar, hay que generar un archivo con extensión *.apk* y descargarlo a la computadora (los archivos con extensión *.apk* son aquellos que se usan para distribuir e instalar aplicaciones en dispositivos con sistema operativo Android). Esto podés hacerlo desplegando el menú *Generar* y seleccionando la opción *App (guardar archivo .apk en mi ordenador)*. Luego, podés enviar este archivo a otros por medio de correo electrónico, una aplicación de mensajería instantánea, etc. Cuando los receptores lo descarguen y lo abran en sus teléfonos, la aplicación se instalará.



Si quisieses distribuir la aplicación de forma más amplia, también la podés publicar en la tienda de aplicaciones *Google Play*. Sin embargo, esta opción no es gratuita. Para más información sobre cómo hacerlo, consultá la página <https://bit.ly/3IQqcCl>.

03

PROCEDIMIENTOS

NOMBRE Y APELLIDO:

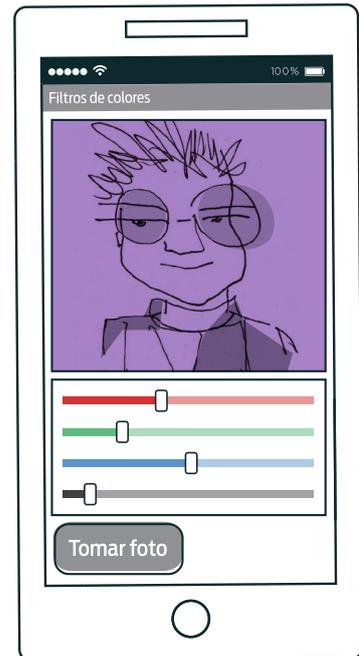
CURSO:

FECHA:

FILTROS DE COLORES

Llegó la hora de que programes una aplicación con la que sacar fotos para aplicarles filtros de colores. Aunque parezca una broma, vas a poder aplicar filtros usando... ¡más de 16 millones de colores!

1. Empezá por la interfaz gráfica. En la imagen que sigue están los componentes y sus dimensiones. En la tabla, las propiedades que tenés que cambiar para que se vea igual a la que te mostramos. Seguí las indicaciones para componerla.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

NOMBRE SUGERIDO	TIPO DE COMPONENTE	PROPIEDAD	VALOR PREVIO	VALOR NUEVO
Screen1 ¹	Pantalla	Título	Screen	Filtros de colores
Lienzo	Lienzo	Alto	Automático	60%
		Ancho	Automático	Ajustar al contenedor
PanelDeslizadores	Disposición horizontal	Disp. horizontal	Izquierda	Centro
		Disp. vertical	Arriba	Centro
		Ancho	Automático	Ajustar al contenedor
DeslizadorRojo	Deslizador	Color izquierda	Por defecto	Rojo
		Color derecha	Por defecto	Rojo
		Ancho	Automático	90%
		Valor máximo	50	255
		Valor mínimo	10	0
		Posición del pulgar	30	0
DeslizadorVerde ²	Deslizador	Color izquierda	Por defecto	Verde
		Color derecha	Por defecto	Verde
DeslizadorAzul ³	Deslizador	Color izquierda	Por defecto	Azul
		Color derecha	Por defecto	Azul
DeslizadorOpacidad ⁴	Deslizador	Color izquierda	Por defecto	Gris oscuro
		Color derecha	Por defecto	Gris claro
		Posición pulgar	30	127
BotónTomarFoto	Botón	Color de fondo	Por defecto	Gris
		Negrita	(ninguno)	✓
		Texto	Texto para botón	Tomar foto
		Color de texto	Por defecto	Blanco

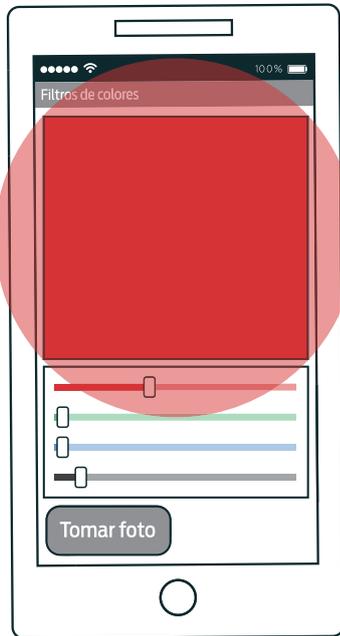
¹ En App Inventor no es posible renombrar el componente que representa la pantalla de la aplicación.

² Los valores de los restantes atributos coinciden con los del deslizador rojo.

³ Los valores de los restantes atributos coinciden con los del deslizador rojo.

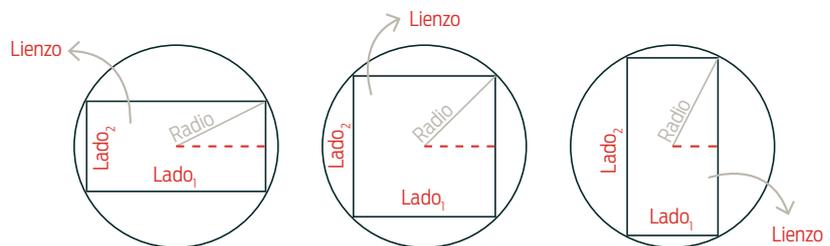
⁴ Los valores de los restantes atributos coinciden con los del deslizador rojo.

2. Ahora tenés que conseguir que cuando se mueva el deslizador rojo se imprima un círculo rojo que cubra todo el lienzo.

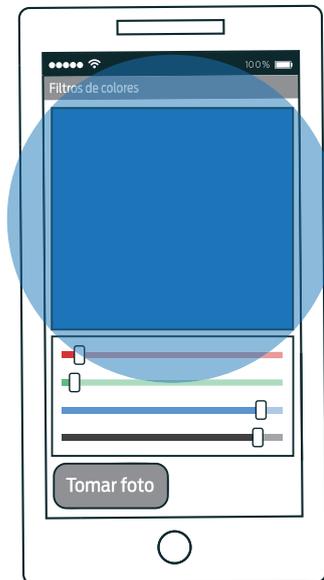
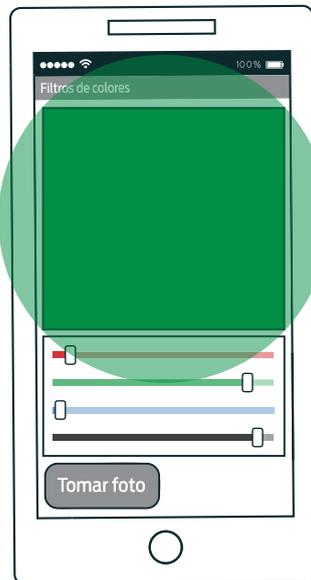
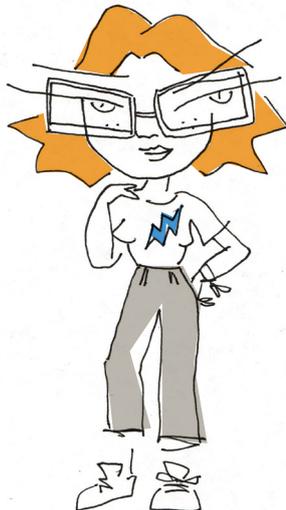


¡PITÁGORAS AL RESCATE!

Un lienzo es un rectángulo y, *a priori*, no conocemos sus dimensiones. ¿Cómo lo cubrimos con un círculo? Por suerte Pitágoras, matemático y filósofo griego, demostró en el siglo VI a.C. un teorema que nos da la respuesta. Para cubrir completamente un rectángulo con un círculo, si situamos el centro del círculo en el centro del rectángulo, su radio debe ser igual a la raíz cuadrada de la suma de los cuadrados de sus lados dividido 2: $radio = \sqrt{\frac{lado_1^2 + lado_2^2}{2}}$.



3. Seguí adelante. Ahora, al mover el deslizador verde, dibujá un círculo verde; y al mover el azul, uno azul.



3. Tres colores es muy poco. No te conformes con menos de ¡16 millones! Encargate de que el color del círculo esté determinado por la posición de las perillas de los deslidores rojo, verde y azul.

PISTA

Investigá el bloque `crear color []`.



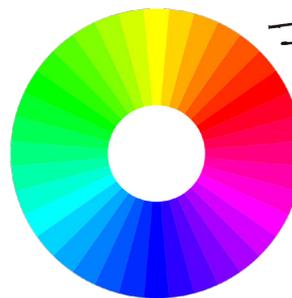
NOMBRE Y APELLIDO:

CURSO:

FECHA:

COLORES EN RGB

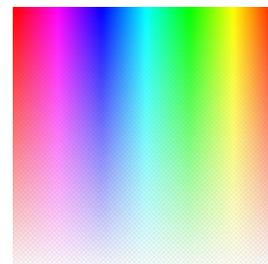
Las imágenes digitales están formadas por una cuadrícula de puntos muy pequeños llamados *píxeles*, cada uno de un color. El color que adquiere un píxel se basa en la mezcla de tres colores. El punto de partida es el negro, que va adquiriendo color a través de tres luces minúsculas: una roja, una verde 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 pequeñas luces se producen distintos colores. Como cada luz admite 256 intensidades diferentes, si hacés las cuentas vas a ver que con este sistema se pueden formar ¡más de 16 millones de colores!



5. Hasta acá solo trabajaste sobre el color del círculo que dibujaste en el lienzo. Sin embargo, el objetivo final es que podamos aplicar filtros sobre fotos para lograr un efecto parecido al que observamos al mirar a través de un papel celofán de color. Llegó el momento de trabajar sobre la transparencia.

COMPOSICIÓN ALPHA

En computación gráfica, hay un proceso llamado **composición alpha** que consiste en combinar una imagen con un fondo para generar la apariencia de transparencia. En tu aplicación, la composición *alpha* te permitirá graduar el nivel de opacidad –o, inversamente, de transparencia– del círculo que termines dibujando sobre la fotografía.



En App Inventor, para crear un color se puede usar un cuarto parámetro. ¿Para vos de qué se trata? Fijate lo que podés hacer presionando la tuerca de arriba a la izquierda del bloque `construye una lista [] [] []`.



¡Atención! Al dibujar un círculo, tenés que borrar el que había antes. ¿Por qué esto es importante?

6. Con lo hecho, ¿qué sucede cuando hacés clic sobre el botón *Tomar foto*? Bueno, ocupate de eso ahora.

NOMBRE Y APELLIDO:

CURSO:

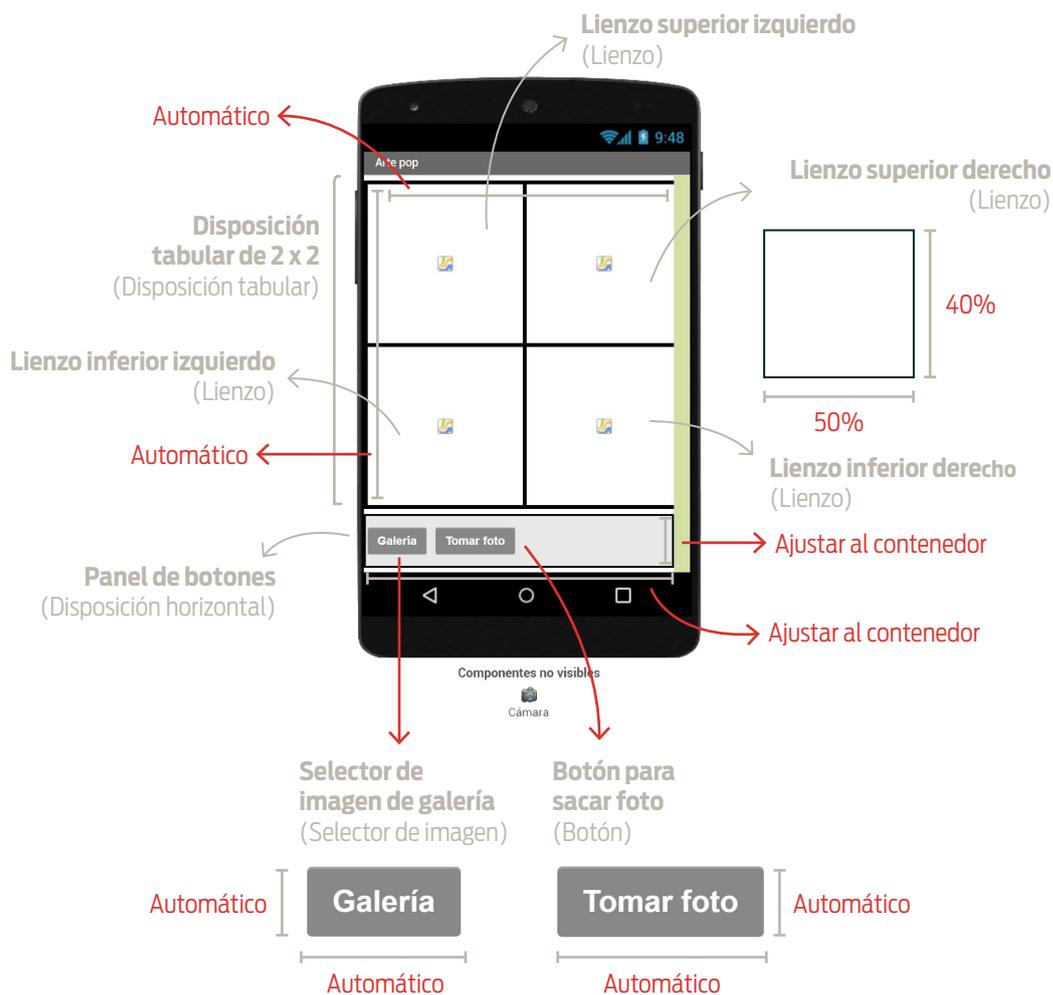
FECHA:

TU ARTE POP

El arte pop fue un movimiento artístico que surgió en el Reino Unido y los Estados Unidos en la década de 1950. Desafiando las tradiciones de las bellas artes, sus obras incorporaban elementos pertenecientes a la cultura popular, como publicidades de latas de tomates e historietas. En esta actividad vas a crear una aplicación para transformar imágenes en ¡obras pop!



1. Arrancá por la interfaz gráfica. En la imagen vas a encontrar los componentes y sus dimensiones. En la tabla, las propiedades que tenés que modificar para que luzca como la imagen.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

SELECTOR DE IMAGEN

App Inventor provee un componente que permite seleccionar una imagen de la galería. Se trata de *SelectorDeImagen*, que se encuentra disponible en *Paleta > Medios*.



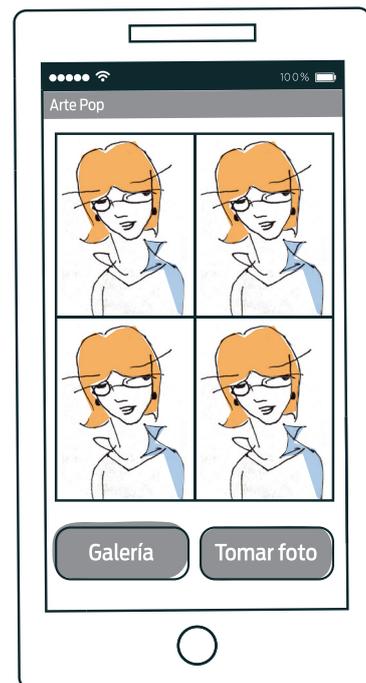
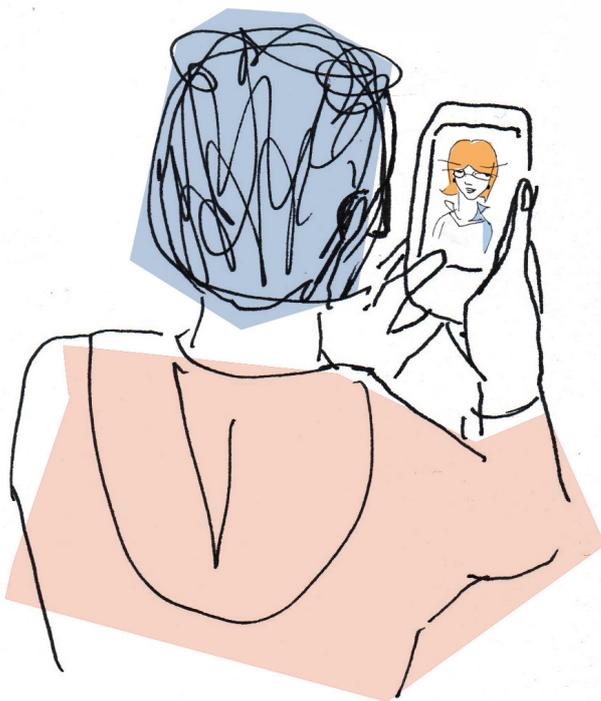
NOMBRE SUGERIDO	TIPO DE COMPONENTE	PROPIEDAD	VALOR PREVIO	VALOR NUEVO
Screen1	Pantalla	Título	Screen	Arte Pop
LienzoSuperior Izquierdo	Lienzo	Alto	Automático	40%
		Ancho	Automático	50%
LienzoSuperior Derecho	Lienzo	Alto	Automático	40%
		Ancho	Automático	50%
LienzoInferior Izquierdo	Lienzo	Alto	Automático	40%
		Ancho	Automático	50%
LienzoInferior Derecho	Lienzo	Alto	Automático	40%
		Ancho	Automático	50%
PanelDeBotones	Disposición horizontal	Disp. vertical	Arriba	Centro
		Alto	Automático	Ajustar al contenedor
		Ancho	Automático	Ajustar al contenedor
SelectorDeImagen DeGalería	Selector de imagen	Color de fondo	Automático	Gris
		Negrita	(ninguno)	✓
		Texto	Texto para botón	Galería
		Color de texto	Por defecto	Blanco
BotónTomarFoto	Botón	Color de fondo	Automático	Gris
		Negrita	(ninguno)	✓
		Texto	Texto para botón	Tomar foto
		Color de texto	Por defecto	Blanco

NOMBRE Y APELLIDO:

CURSO:

FECHA:

2. Incorporá los bloques necesarios para sacar una foto y ubicarla como fondo de los cuatro lienzos. ¿Cómo lo hiciste? ¿Qué eventos manejaste para lograrlo?



EL PADRE DEL ARTE POP

Uno de los exponentes más relevantes de este movimiento fue el artista Andy Warhol. Hay un museo exclusivamente dedicado a este artista (<https://www.warhol.org/>), además de que sus obras forman parte de exposiciones de muchos de los museos más importantes del mundo.

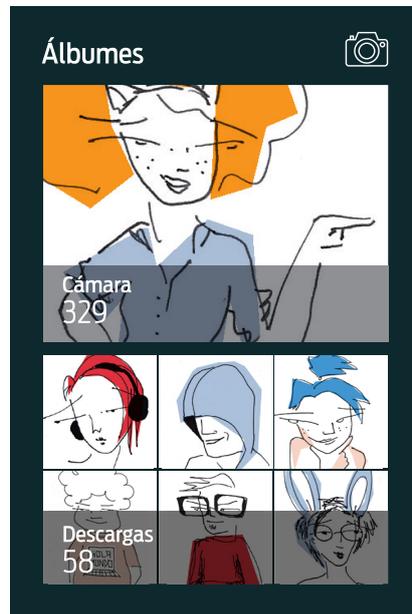


NOMBRE Y APELLIDO:

CURSO:

FECHA:

3. Ahora, ocupate de que la imagen también pueda seleccionarse desde la galería de imágenes de tu teléfono.



¿Qué diferencia encontrarás entre el bloque para manejar el evento que se produce al sacar una foto y el que se produce al seleccionar una imagen de la galería?

cuando SelectorDelImagenDeGalería .DespuésDeSelección
ejecutar

cuando Cámara .DespuésDeTomarFoto
imagen
ejecutar

PARA QUE TENGAS EN CUENTA

La foto que selecciones en la galería está disponible en la propiedad

`SelectorDeImagen.Selección.`

SelectorDelImagenDeGalería Selección

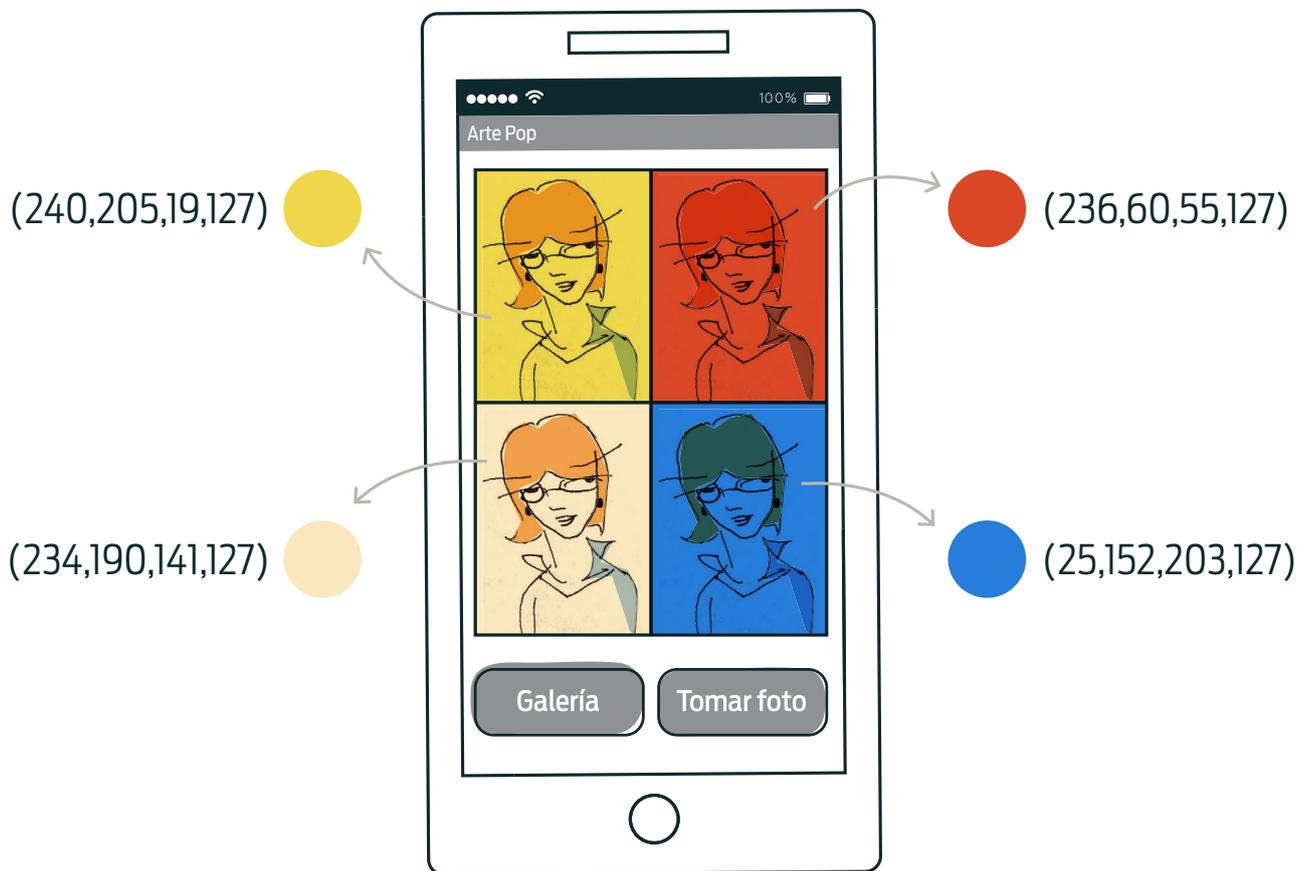


NOMBRE Y APELLIDO:

CURSO:

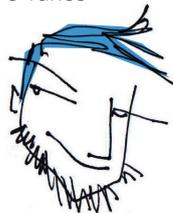
FECHA:

4. Llegó el momento de transformar la foto en una pieza pop. Aplicá filtros de cuatro colores sobre cada uno de los lienzos. Acá te pasamos una opción de colores que remite a algunas obras de este movimiento artístico, pero probá con otras hasta encontrar la combinación que más te guste.



CONSEJOS

- Usá procedimientos para descomponer problemas en unidades más simples.
- Si en tu programa te encontrás con fragmentos muy parecidos entre sí, cuyo propósito es prácticamente el mismo, considerá definir un nuevo procedimiento con uno o varios parámetros.



CUALQUIER COMPONENTE

¡Investigá qué hacen los bloques que se encuentran en la categoría *CualquierComponente*!



04

ALTERNATIVAS,
REPETICIONES Y
VARIABLES

NOMBRE Y APELLIDO:

CURSO:

FECHA:

CARA O CECA

Muchas veces enfrentamos situaciones que se tienen que resolver mediante el azar, como decidir qué materia estudiar cuando falta solo un día para los exámenes de dos materias y no sabemos nada de ninguna; o si será tu amigo o vos quien irá a la pizzería de la otra cuadra a comprar unas empanadas y volver. Ahora vas a hacer una aplicación para poder resolverlas, aun cuando no tengas una moneda en el bolsillo.



1. Creá un proyecto de *App Inventor* e incorporá cuatro archivos multimedia: inicio.png, cara.png, ceca.png y lanzamiento-moneda.mp3.

cara.png



ceca.png



PARA AGREGAR ARCHIVOS A UN PROYECTO

Para agregar archivos a un proyecto, tenés que presionar el botón *Subir archivo* del panel *Medios*. Entonces,

se desplegará una ventana en la que, en primer lugar, tenés que hacer clic en *Seleccionar archivo* (para seleccionar el archivo que querés incorporar); y, en segundo, presionar *Aceptar* para incorporarlo al proyecto. Finalmente, en *Medios* verás aparecer el archivo incorporado.

Medios

Subir archivo...

Subir archivo...

Seleccionar archivo Ningún archivo seleccionado

Cancelar

Aceptar

Subir archivo...

Seleccionar archivo cara.png

Cancelar

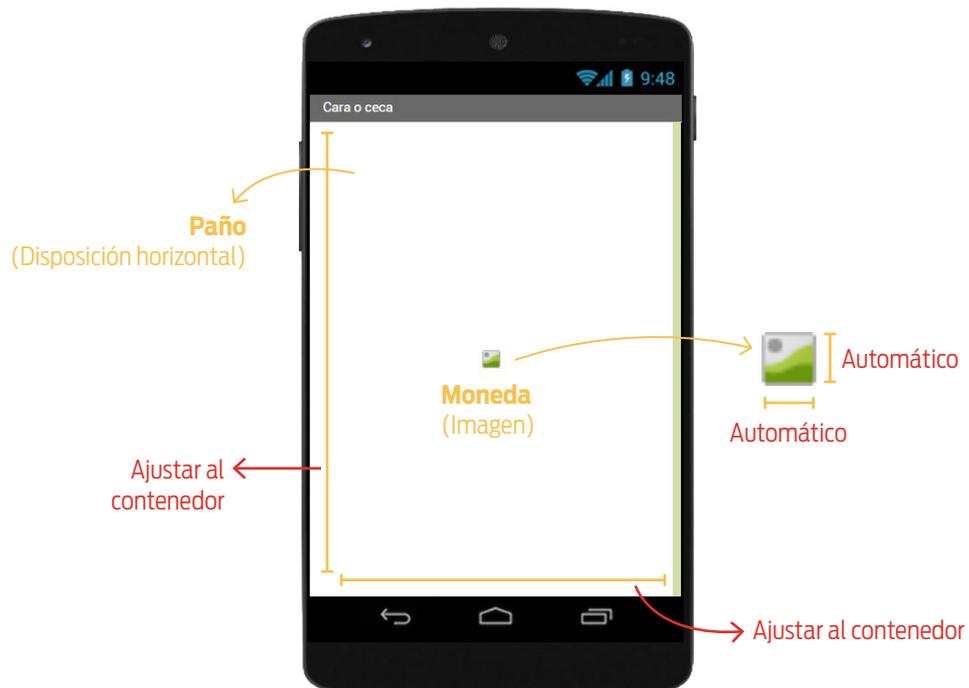
Aceptar

NOMBRE Y APELLIDO:

CURSO:

FECHA:

2. Ahora armá la interfaz gráfica. En la imagen están los componentes y sus dimensiones. En la tabla, las propiedades que tenés que cambiar para que se vea igual a la que te mostramos. Seguí las indicaciones para componerla.



NOMBRE SUGERIDO	TIPO DE COMPONENTE	PROPIEDAD	VALOR PREVIO	VALOR NUEVO
Screen1	Pantalla	Título	Screen1	Cara o ceca
Paño	Disposición horizontal	Disposición horizontal	Izquierda	Centro
		Disposición vertical	Arriba	Centro
		Alto	Automático	Ajustar al contenedor
		Ancho	Automático	Ajustar al contenedor
Moneda	Imagen	Foto	Ninguno	inicio.png
		Escalar foto al tamaño máximo	(ninguno)	✓

Ejecutá la aplicación en tu teléfono. ¿Qué aparece en la pantalla?

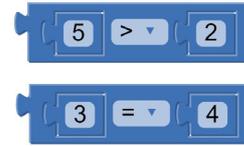
NOMBRE Y APELLIDO:

CURSO:

FECHA:

EXPRESIONES BOOLEANAS

En Ciencias de la Computación, una expresión booleana es una construcción sintáctica que, cuando se evalúa, o bien es verdadera o bien es falsa. Por ejemplo, $5 > 2$ es (verdadera) y $3 = 4$ (falsa).



3. ¡Llegó el momento del azar! Al agitar el teléfono, en la pantalla tiene que aparecer con idéntica probabilidad, o bien el lado cara de la moneda o bien el lado ceca. ¿Cómo lo conseguiste?

AYUDA

Explorá el entorno en busca de algún bloque que te permita generar números aleatorios. Tené en cuenta que vas a necesitar una instrucción que te permita hacer una cosa si una cierta condición es verdadera y, otra, si es falsa.



4. Por último tenés que lograr que, al zarandearlo, el teléfono vibre y emita el sonido que se produce al lanzar una moneda. ¿Agregaste algún componente no visible? ¿Para qué?

¿SABÍAS 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 ejemplo, programas para jugar a la generala o al truco.



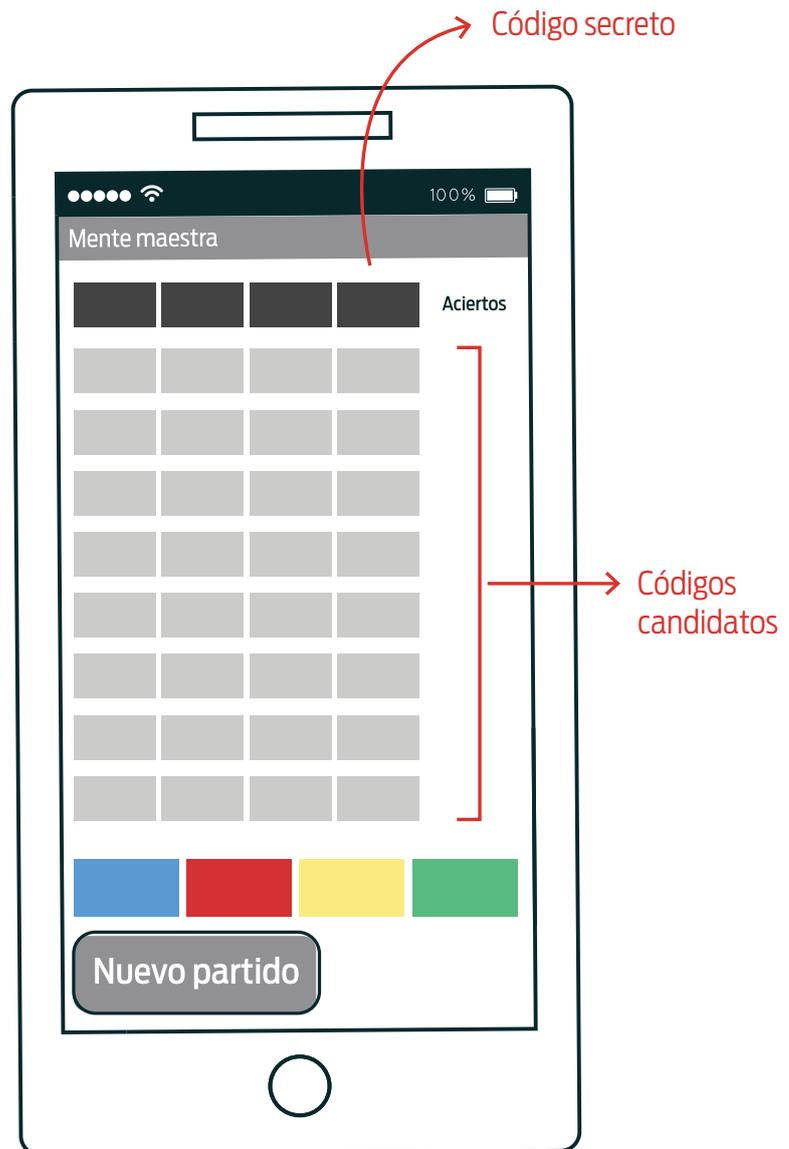
MENTE MAESTRA, LA APLICACIÓN

¡Mirá esta aplicación del juego Mente maestra! En esta versión, el jugador tiene que estar muy atento y concentrado porque, para adivinar el código secreto, cuenta solo con ocho intentos.

ANEXO

¡Comienza el juego!

Esto es lo que se ve en la pantalla apenas la aplicación comienza su ejecución. Los rectángulos en gris oscuro de arriba ocultan el código secreto; y cada una de las ocho líneas siguientes mostrará cada uno de los intentos del jugador por descubrirlo.



Probando, probando, probando...

Para seleccionar los colores de los códigos candidatos, el usuario presionará los botones azul, rojo, amarillo y verde. A medida que lo haga, irán apareciendo los colores elegidos en los rectángulos reservados para los ocho ensayos. Además, cuando completan un intento, a la derecha se mostrará la cantidad de coincidencias entre el código ingresado y el código secreto.

Botones para ingresar códigos candidatos

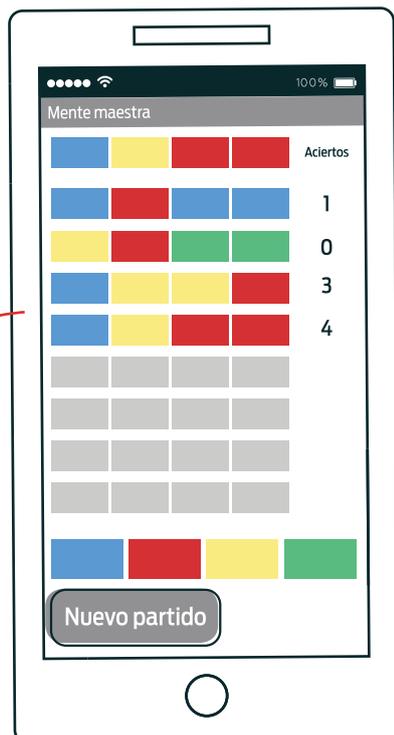


Número de coincidencias entre el código secreto y el código candidato

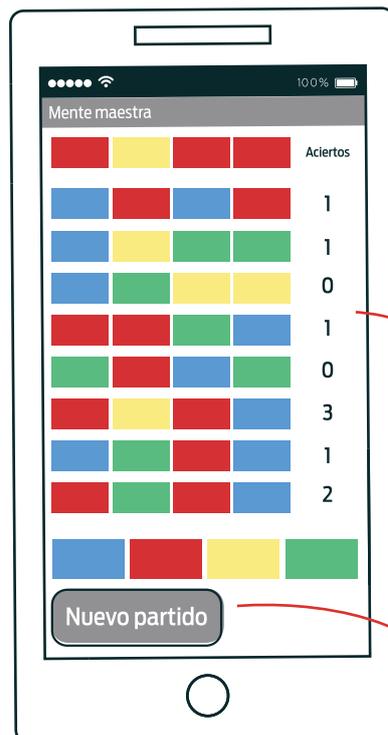
Y en un momento se acaba

El juego finaliza tanto cuando el código secreto es descubierto como cuando se agotan las ocho posibilidades. Por supuesto, en el primer caso el jugador gana y, en el segundo, pierde. Igual, este juego da revancha: presionado el botón *Nuevo partido* todo comienza otra vez.

El jugador gana



El jugador pierde



Para empezar de nuevo

NOMBRE Y APELLIDO:

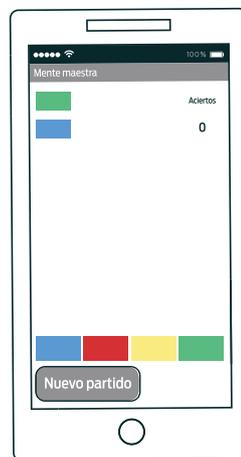
CURSO:

FECHA:

MENTE MAESTRA, PARTE I

¡Primera versión del juego! Acá, el código secreto es solo un color. Además, hay un solo intento. Para ganar, ¡hay que pegarle de una!

1. Comenzá armando la interfaz gráfica de esta primera aproximación al Mente maestra. Acá abajo podés ver los componentes y sus dimensiones. En la tabla, las propiedades que tenés que cambiar para que se vea igual a la que te mostramos. Seguí las indicaciones para componerla.



Código Secreto
(Etiqueta en posición (1,1))
19% x 7%

Panel Código Secreto
(Disposición tabular de 1x5)

Código Candidato
(Etiqueta en posición (1,1))
19% x 7%

Panel Código Candidato
(Disposición tabular de 8x5)

Botón Nuevo Partido
(Botón)
8% x Automático

Panel Botonera
(Disposición Horizontal)
10%
Ajustar al contenedor

7% x 25%
Botón Azul (Botón) **Botón Rojo** **Botón Amarillo** **Botón Verde**

Etiqueta Aciertos
(Etiqueta en posición (1,5))
10% x 19%
7%
Ajustar al contenedor

Etiqueta Coincidencias
(Etiqueta en posición (1,5))
19% x 7%
62%
Ajustar al contenedor

NOMBRE Y APELLIDO:

CURSO:

FECHA:

NOMBRE SUGERIDO	TIPO DE COMPONENTE	PROPIEDAD	VALOR PREVIO	VALOR NUEVO
Screen1	Pantalla	Título	Screen1	Mente maestra
Panel Código Secreto	Disposición tabular	Columnas	2	5
		Alto	Automático	10%
		Ancho	Automático	Ajustar al contenedor
		Registros	2	1
Código Secreto	Etiqueta	Color de fondo	Ninguno	Gris oscuro
		Alto	Automático	7%
		Ancho	Automático	19%
		Texto	Texto para etiqueta	(ninguno)
Etiqueta Aciertos	Etiqueta	Negrita	(ninguno)	✓
		Alto	Automático	7%
		Ancho	Automático	19%
		Texto	Texto para etiqueta	Aciertos
		Posición del texto	Izquierda	Centro
Panel Código Candidato	Disposición tabular	Columnas	2	5
		Alto	Automático	62%
		Ancho	Automático	Ajustar al contenedor
		Registros	2	8
Código Candidato	Etiqueta	Color de fondo	Ninguno	Gris Claro
		Alto	Automático	7%
		Ancho	Automático	19%
		Texto	Texto para etiqueta	(ninguno)
Etiqueta Coincidencias	Etiqueta	Negrita	(ninguno)	✓
		Alto	Automático	7%
		Ancho	Automático	19%
		Texto	Texto para etiqueta	(ninguno)
		Posición del texto	Izquierda	Centro

NOMBRE Y APELLIDO:

CURSO:

FECHA:

NOMBRE SUGERIDO	TIPO DE COMPONENTE	PROPIEDAD	VALOR PREVIO	VALOR NUEVO
Panel Botonera	Disposición horizontal	Disp. horizontal	Izquierda	Centro
		Disp. vertical	Arriba	Abajo
		Alto	Automático	10%
		Ancho	Automático	Ajustar al contenedor
Botón Azul	Botón	Color de fondo	Por defecto	Azul
		Alto	Automático	7%
		Ancho	Automático	25%
		Texto	Texto para botón	(ninguno)
Botón Rojo	Botón	Color de fondo	Por defecto	Rojo
		Alto	Automático	7%
		Ancho	Automático	25%
		Texto	Texto para botón	(ninguno)
Botón Amarillo	Botón	Color de fondo	Por defecto	Amarillo
		Alto	Automático	7%
		Ancho	Automático	25%
		Texto	Texto para botón	(ninguno)
Botón Verde	Botón	Color de fondo	Por defecto	Verde
		Alto	Automático	7%
		Ancho	Automático	25%
		Texto	Texto para botón	(ninguno)
Botón Nuevo Partido	Botón	Color de fondo	Por defecto	Gris
		Negrita	(ninguno)	✓
		Alto	Automático	8%
		Texto	Texto para botón	Nuevo partido
		Color de texto	Por defecto	Blanco

NOMBRE Y APELLIDO:

CURSO:

FECHA:

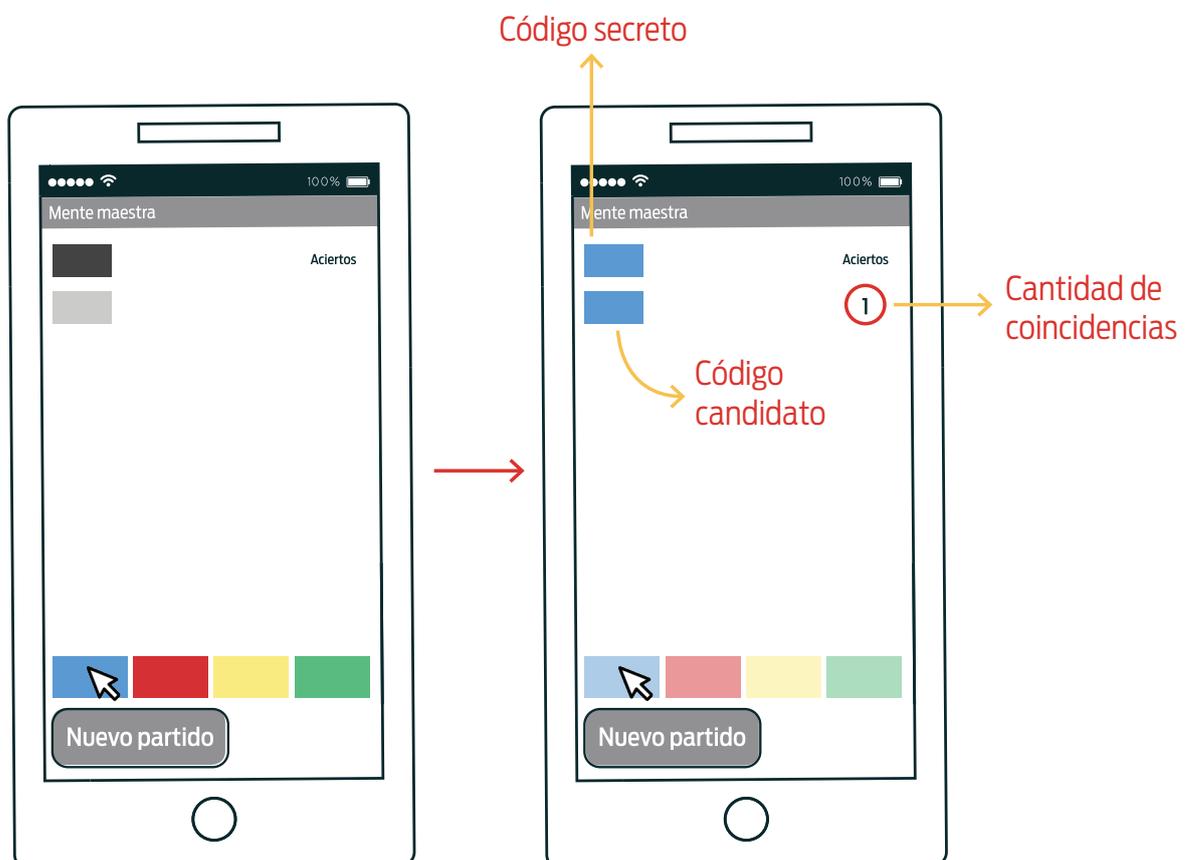
2. Ni bien comienza su ejecución, el juego tiene que generar al azar un código secreto de longitud 1. Se trata, en definitiva, de librar a la suerte la elección de un color. Tené en cuenta que, más tarde, quien juegue buscará adivinarlo. ¿Qué herramienta usaste para resolver el desafío? ¿Para qué?

PARA NO OLVIDAR...

Una variable es un nombre –que elegimos nosotros al programar– que denota un espacio de la memoria de la computadora donde se puede guardar un valor y, luego, recuperarlo o modificarlo. Las variables permiten, por ejemplo, que los juegos registren la cantidad de puntos que alcanza un jugador a medida que el juego avanza.



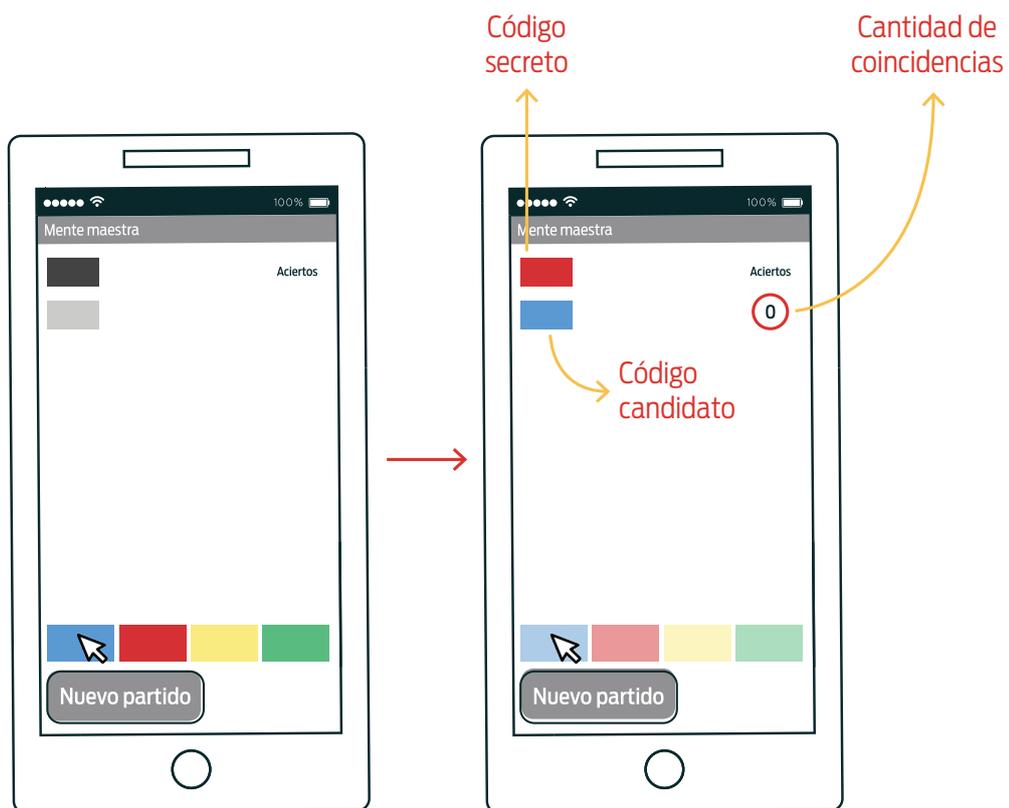
3. Ahora tenés que encargarte de que, una vez que el usuario presione el botón de azul, se muestre si hay o no una coincidencia entre la elección del jugador y el color secreto. Acá podés ver lo que tiene que pasar tanto si hay coincidencia como si no (en la página siguiente).



NOMBRE Y APELLIDO:

CURSO:

FECHA:



4. No solo de apretar el botón azul se trata este juego. También hay que resolver lo que sucede cuando se haga clic sobre el rojo, el amarillo y el verde. Resuélvelo evitando que en tu programa haya redundancia. ¿Qué hay que hacer, en general, si cuando programamos observamos que hay distintas partes del programa que son iguales o muy parecidas?

PROCEDIMIENTOS PARA SIMPLIFICAR LO COMPLEJO

Tené siempre presente que, al resolver un problema, es conveniente comenzar pensando las partes que componen una solución. En tu programa, cada una de estas partes las podés resolver en un procedimiento separado.

5. A los jugadores vamos a darles una sola oportunidad para que adivinen el color secreto. Una vez hecha la elección, los botones tienen que quedar deshabilitados. ¿Cómo lo conseguiste?
-
6. Y ahora, todo otra vez. Con un clic sobre el botón *Nuevo partido* hay que dejar todo como al principio, listo para que el jugador pueda intentar otra vez descubrir un nuevo color secreto.

NOMBRE Y APELLIDO:

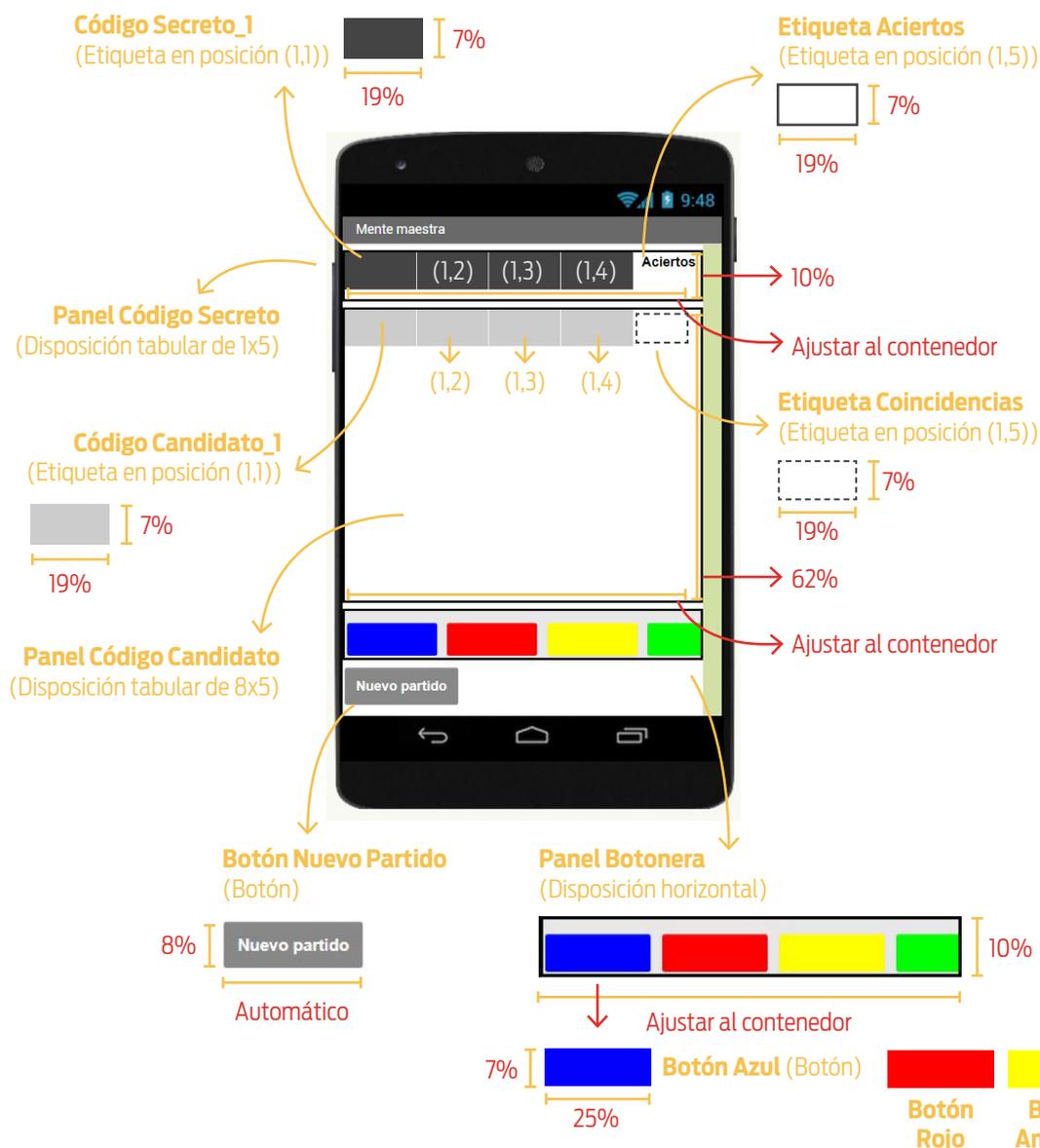
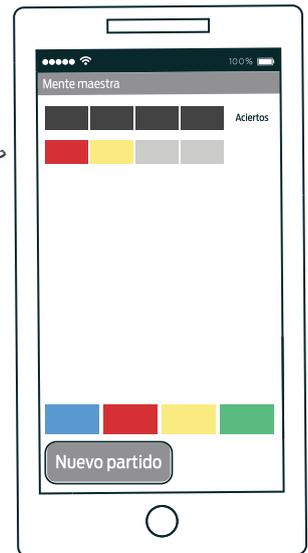
CURSO:

FECHA:

MENTE MAESTRA, PARTE II

Ahora la cosa se pone más complicada. Los jugadores tienen un intento para adivinar un código de cuatro colores. ¡Solo una chance entre 256!

1. Comenzá armando la interfaz gráfica. Acá abajo podés ver los componentes y sus dimensiones. En la tabla, las propiedades que tenés que cambiar para que se vea igual a la que te mostramos. Seguí las indicaciones para componerla.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

NOMBRE SUGERIDO	TIPO DE COMPONENTE	PROPIEDAD	VALOR PREVIO	VALOR NUEVO
Screen1	Pantalla	Título	Screen1	Mente maestra
Panel Código Secreto	Disposición tabular	Columnas	2	5
		Alto	Automático	10%
		Ancho	Automático	Ajustar al contenedor
		Registros	2	1
Código Secreto_1 ¹	Etiqueta	Color de fondo	Ninguno	Gris oscuro
		Alto	Automático	7%
		Ancho	Automático	19%
		Texto	Texto para etiqueta	(ninguno)
Etiqueta Aciertos	Etiqueta	Negrita	(ninguno)	✓
		Alto	Automático	7%
		Ancho	Automático	19%
		Texto	Texto para etiqueta	Aciertos
		Posición del texto	Izquierda	Centro
Panel Código Candidato	Disposición tabular	Columnas	2	5
		Alto	Automático	62%
		Ancho	Automático	Ajustar al contenedor
		Registros	2	8
Código Candidato_1 ²	Etiqueta	Color de fondo	Ninguno	Gris claro
		Alto	Automático	7%
		Ancho	Automático	19%
		Texto	Texto para etiqueta	(ninguno)
Etiqueta Coincidencias	Etiqueta	Negrita	(ninguno)	✓
		Alto	Automático	7%
		Ancho	Automático	19%
		Texto	Texto para etiqueta	(ninguno)
		Posición del texto	Izquierda	Centro

¹Las restantes etiquetas del código secreto requieren los mismos cambios.
²Las restantes etiquetas del código candidato requieren los mismos cambios.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

NOMBRE SUGERIDO	TIPO DE COMPONENTE	PROPIEDAD	VALOR PREVIO	VALOR NUEVO
Panel Botonera	Disposición horizontal	Disp. horizontal	Izquierda	Centro
		Disp. vertical	Arriba	Abajo
		Alto	Automático	10%
		Ancho	Automático	Ajustar al contenedor
Botón Azul	Botón	Color de fondo	Por defecto	Azul
		Alto	Automático	7%
		Ancho	Automático	25%
		Texto	Texto para botón	(ninguno)
Botón Rojo	Botón	Color de fondo	Por defecto	Rojo
		Alto	Automático	7%
		Ancho	Automático	25%
		Texto	Texto para botón	(ninguno)
Botón Amarillo	Botón	Color de fondo	Por defecto	Amarillo
		Alto	Automático	7%
		Ancho	Automático	25%
		Texto	Texto para botón	(ninguno)
Botón Verde	Botón	Color de fondo	Por defecto	Verde
		Alto	Automático	7%
		Ancho	Automático	25%
		Texto	Texto para botón	(ninguno)
Botón Nuevo Partido	Botón	Color de fondo	Por defecto	Gris
		Negrita	(ninguno)	✓
		Alto	Automático	8%
		Texto	Texto para botón	Nuevo partido
		Color de texto	Por defecto	Blanco

NOMBRE Y APELLIDO:

CURSO:

FECHA:

2. Ni bien comienza su ejecución, la aplicación tiene que generar al azar un código secreto de 4 posiciones. Además, a pesar de tener cuatro posiciones con colores, tenés que conservar el código en una única variable.

LISTAS

En computación, una **lista** es un tipo de datos que representa una secuencia de elementos ordenados. Se puede pensar, por ejemplo, en listas de números, listas de colores o listas de palabras, etc. Cada elemento, además, puede aparecer más de una vez en una lista.

1 7 21 3



Lista de cuatro números



Lista de cuatro colores

Hola matungo



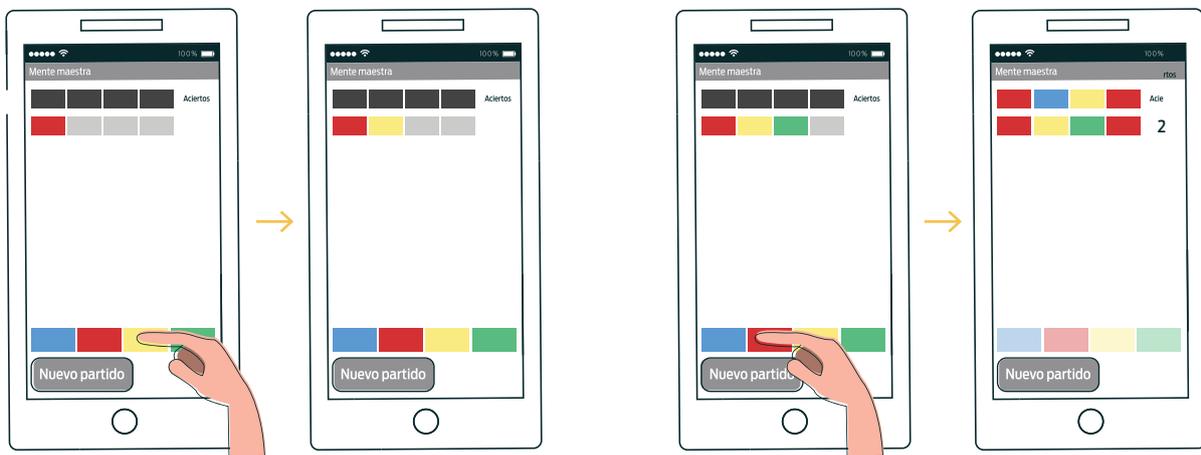
Lista de dos palabras



¿Usaste listas? ¿Por qué?

3. Ahora tenés que programar el manejo de los eventos que se producen cuando se presionan los botones de colores. Hacerlo involucra (i) actualizar en la pantalla el código candidato agregándole el color elegido por el jugador, y (ii) chequear si al agregar el nuevo color se completa el código candidato, caso en el cual hay que develar el código secreto, mostrar la cantidad de coincidencias y deshabilitar los botones de colores.

Acá abajo podés ver dos ejemplos: uno en el que al agregar un color no se completa el código candidato y otro en el que sí.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

¿Cómo hiciste para que se calcule la cantidad de aciertos una vez que el jugador ingrese los cuatro colores del código candidato? ¿Usaste algún bloque que permita repetir secuencias de instrucciones? ¿Cuál?

4. Por último, el botón *Nuevo partido* tiene que dejar todo listo para intentar adivinar el código secreto otra vez. ¿Reusaste alguna parte del programa para conseguir el objetivo? ¿Cuál? ¿Para qué?

NOMBRE Y APELLIDO:

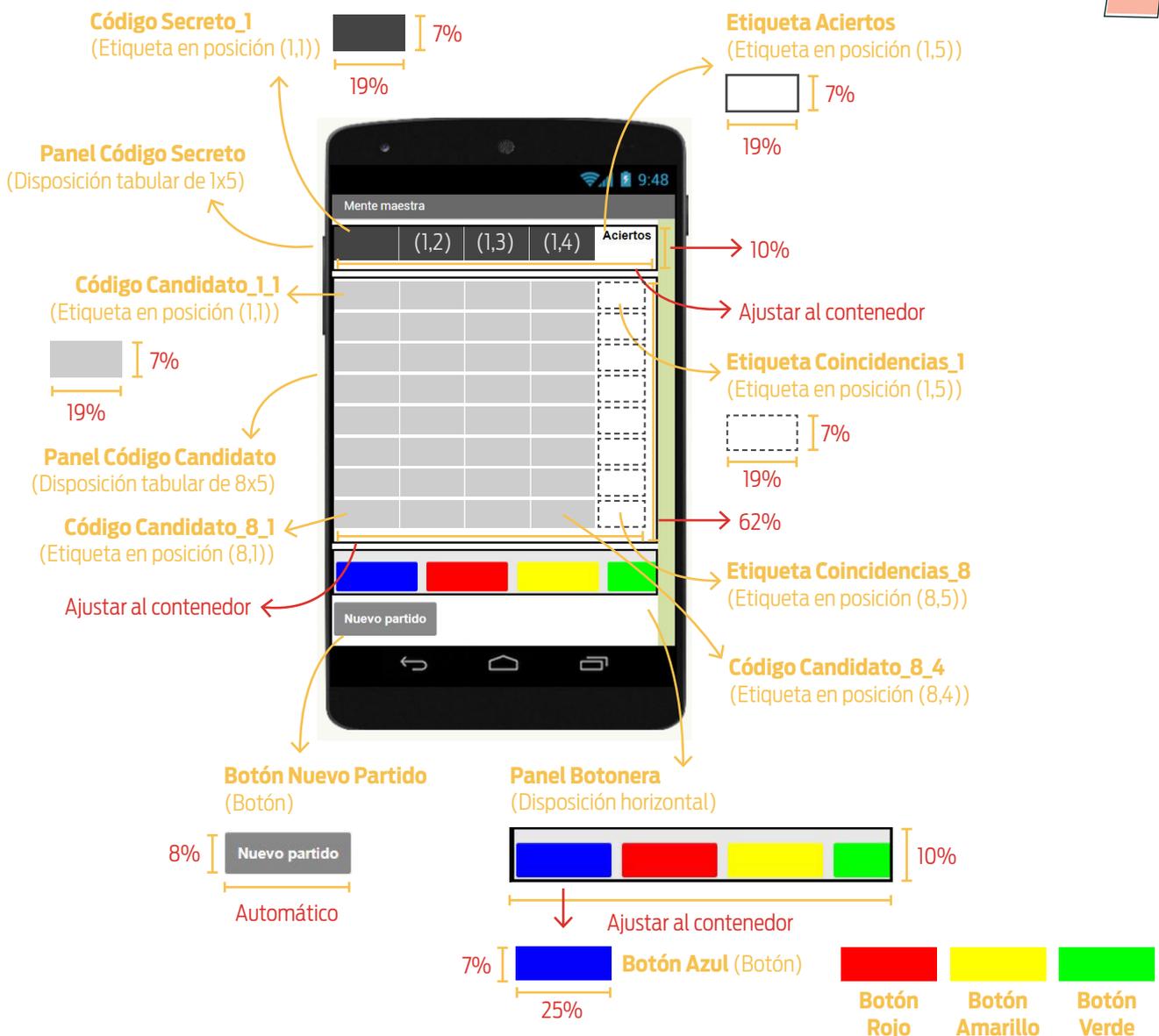
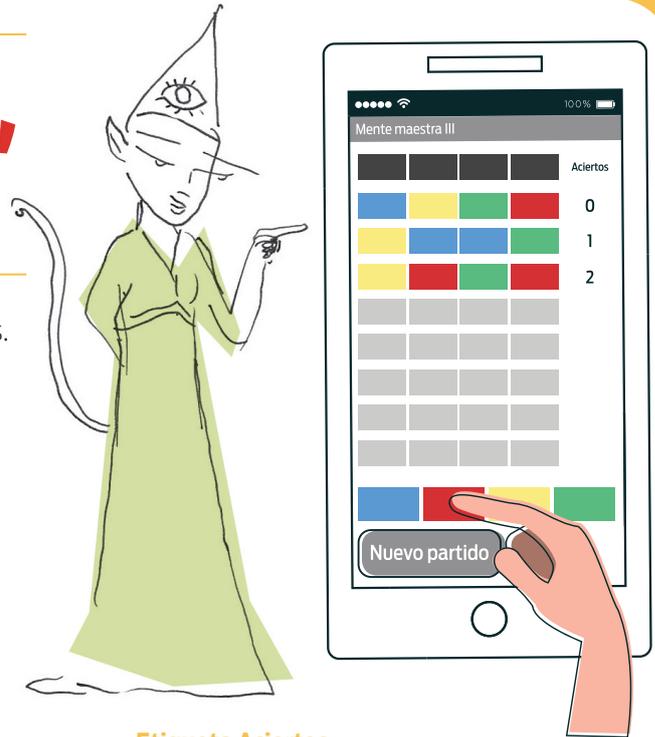
CURSO:

FECHA:

MENTE MAESTRA, PARTE III

Y ahora sí: códigos de cuatro colores y ocho intentos. ¡A armar la versión completa del Mente maestra!

1. Comenzá armando la interfaz gráfica. En la imagen están los componentes y sus dimensiones. En la tabla, las propiedades que tenés que cambiar para que se vea igual a la que te mostramos. Seguí las indicaciones para componerla.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

NOMBRE SUGERIDO	TIPO DE COMPONENTE	PROPIEDAD	VALOR PREVIO	VALOR NUEVO
Screen1	Pantalla	Título	Screen1	Mente maestra
Panel Código Secreto	Disposición tabular	Columnas	2	5
		Alto	Automático	10%
		Ancho	Automático	Ajustar al contenedor
		Registros	2	1
Código Secreto_1 ¹	Etiqueta	Color	Ninguno	Gris oscuro
		Alto	Automático	7%
		Ancho	Automático	19%
		Texto	Texto para etiqueta	(ninguno)
Etiqueta Aciertos	Etiqueta	Negrita	(ninguno)	✓
		Alto	Automático	7%
		Ancho	Automático	19%
		Texto	Texto para etiqueta	Aciertos
		Posición del texto	Izquierda	Centro
Panel Código Candidato	Disposición tabular	Columnas	2	5
		Alto	Automático	62%
		Ancho	Automático	Ajustar al contenedor
		Registros	2	8
Código Candidato_1_1 ²	Etiqueta	Color de fondo	Ninguno	Gris claro
		Alto	Automático	7%
		Ancho	Automático	19%
		Texto	Texto para etiqueta	(ninguno)
Etiqueta Coincidencias_1 ³	Etiqueta	Negrita	(ninguno)	✓
		Alto	Automático	7%
		Ancho	Automático	19%
		Texto	Texto para etiqueta	(ninguno)
		Posición del texto	Izquierda	Centro

¹Las restantes etiquetas del código secreto requieren los mismos cambios.

²Las restantes etiquetas del código candidato requieren los mismos cambios.

³Las restantes etiquetas para mostrar la cantidad de aciertos de un intento requieren los mismos cambios.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

NOMBRE SUGERIDO	TIPO DE COMPONENTE	PROPIEDAD	VALOR PREVIO	VALOR NUEVO
Panel Botonera	Disposición horizontal	Disp. horizontal	Izquierda	Centro
		Disp. vertical	Arriba	Abajo
		Alto	Automático	10%
		Ancho	Automático	Ajustar al contenedor
Botón Azul	Botón	Color de fondo	Por defecto	Azul
		Alto	Automático	7%
		Ancho	Automático	25%
		Texto	Texto para botón	(ninguno)
Botón Rojo	Botón	Color de fondo	Por defecto	Rojo
		Alto	Automático	7%
		Ancho	Automático	25%
		Texto	Texto para botón	(ninguno)
Botón Amarillo	Botón	Color de fondo	Por defecto	Amarillo
		Alto	Automático	7%
		Ancho	Automático	25%
		Texto	Texto para botón	(ninguno)
Botón Verde	Botón	Color de fondo	Por defecto	Verde
		Alto	Automático	7%
		Ancho	Automático	25%
		Texto	Texto para botón	(ninguno)
Botón Nuevo Partido	Botón	Color de fondo	Por defecto	Gris
		Negrita	(ninguno)	✓
		Alto	Automático	8%
		Texto	Texto para botón	Nuevo partido
		Color de texto	Por defecto	Blanco

NOMBRE Y APELLIDO:

CURSO:

FECHA:

- 2.** Ejecutá la aplicación. ¿Con qué te encontraste? ¿Cuántos intentos tuviste para descubrir el código secreto? ¿Por qué?

- 3.** El juego tienen que permitir que un usuario cuente con ocho intentos para develar los cuatro colores ocultos. Usando lenguaje coloquial, delineá una estrategia para conseguirlo.

Programá en App Inventor la estrategia propuesta.

UNA AYUDA

El procedimiento `manejarSelecciónDeColor (colorSeleccionado)` describe la estrategia de la versión anterior, en la que solo había un intento. A lo mejor te sirve como referencia.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

4. No solo cuando se completan los ocho intentos el juego tiene que terminar. ¡Si el jugador descubre el código secreto tampoco tiene sentido seguir! Modificá el programa para que también finalice cuando el jugador triunfa.

PARA QUE TENGAS EN CUENTA

En App Inventor hay dos bloques para armar expresiones booleanas que se construyen a partir de otras más simples: `[] y []` y `[] o []`. La primera corresponde a la operación lógica de **conjunción**, y la segunda, a la de **disyunción**. ¿Te imaginás cómo usar alguna de ellas para resolver la consigna?



5. Por último, ocupate de que cuando se presione el botón *Nuevo partido* se pueda jugar otra vez. ¿Qué cambios tuviste que hacer para conseguirlo?

PROCEDIMIENTOS Y FUNCIONES

¿Sabés cuál es la diferencia entre `como (procedimiento) / ejecutar { }` y `como (procedimiento) / resultado []`? Mientras que el primero se utiliza para realizar una tarea, el segundo se usa para calcular y retornar un valor. En los lenguajes de programación, a estos últimos se los refiere como **funciones**, cuya característica principal es que producen información nueva que es relevante para un programa. ¡Incorporá funciones en tu programa y elegí concienzudamente sus nombres!



05

REPRESENTACIÓN
DE LA
INFORMACIÓN

NOMBRE Y APELLIDO:

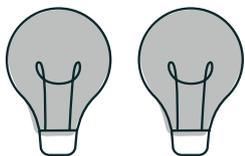
CURSO:

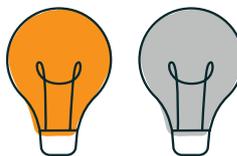
FECHA:

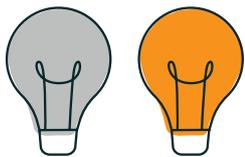
LUCES BINARIAS

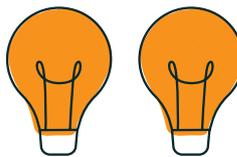
¿Sabías que, usando un elemento que pueda estar en 2 estados diferentes, podemos representar un montón de cosas? ¡Vamos a trabajar con luces para que veas qué poco hace falta para poder decir mucho!

1. Completá las combinaciones posibles de acuerdo a los estados de 2 lamparitas. Usá las palabras *encendida* y *apagada*.









2. Si usáramos cada combinación para representar un número, ¿cuántos números distintos podríamos representar? ¿Y si en lugar de usar 2 lamparitas usáramos 3?

3. ¿Cuántas lamparitas hacen falta para representar 64 números distintos? ¿Y para representar 128?

SISTEMA BINARIO

Todos los datos que usa una computadora se almacenan usando solo dos valores. Internamente, la computadora usa dos niveles de voltaje claramente diferenciables, y para referirnos a ellos solemos usar los dígitos 0 y 1; pero también podríamos usar sí y no, o blanco y negro, o puño y palma, o una lamparita encendida y una apagada. Solo es necesario elegir dos representaciones diferentes para que sea posible distinguirlas sin ninguna duda.

01

NOMBRE Y APELLIDO:

CURSO:

FECHA:

CARTAS Y NÚMEROS BINARIOS

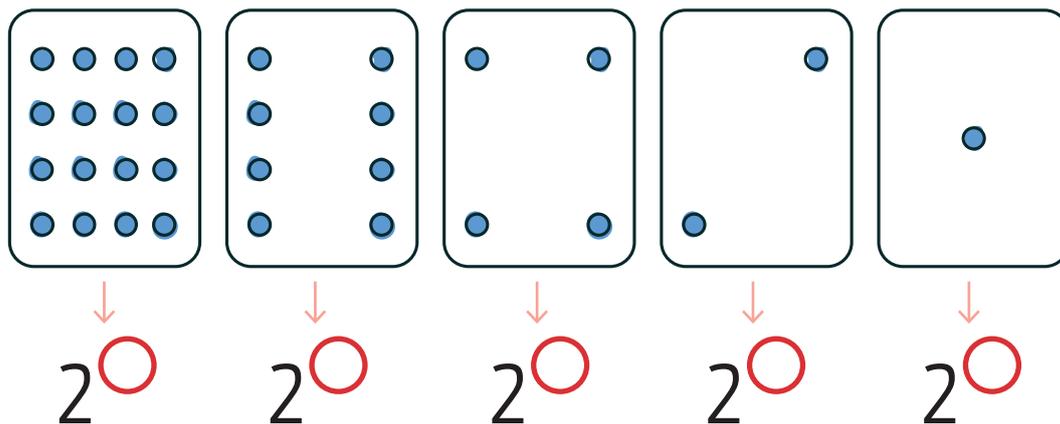


¡Vamos a jugar a las cartas! Para empezar, recortá las cartas de la ficha.

1. ¿Qué cartas hay que dejar visibles y cuáles hay que dar vuelta para representar el número 11? ¿Y para el número 30?

2. Ahora tenés que representar todos los números que puedas, comenzando por el 0, siguiendo por el 1, y continuando en orden hasta el más grande al que puedas llegar. ¿Cuál es el mayor número que alcanzaste?

3. Expresá la cantidad de puntos de cada carta como una potencia de 2. ¿Qué relación hay entre el exponente y la posición que ocupa cada carta en la secuencia, si la leemos de derecha a izquierda?



4. Con la ayuda de las cartas, escribí la representación binaria de los números 15 y 22.

15: _____

22: _____

SISTEMA BINARIO

El uso de los dígitos 0 y 1 es muy común y se los llama *dígitos binarios*. Su importancia es tal que se inventó una palabra para describirlos tomando las dos primeras letras y la última letra de la expresión *dígito binario* en inglés, *binary digit*: bit.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

A large dashed-line box containing seven smaller rounded rectangles. The top-left rectangle contains a 4x4 grid of 16 blue dots. The top-right rectangle contains two vertical columns of 4 blue dots each. The middle-left rectangle contains four blue dots at the corners. The middle-right rectangle contains two blue dots, one at the top and one at the bottom. The bottom-left rectangle contains a single blue dot in the center. A scissors icon is located on the right side of the dashed box.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

EQUIS RAYA

Todos los días usamos textos en nuestros dispositivos digitales. Enviamos mensajes de texto, hacemos comentarios en las redes sociales, leemos páginas web y muchas cosas más. En esta actividad vamos a trabajar sobre los principios de la codificación de texto usados por las computadoras.



1. Mirá la codificación de letras X-RAYA y a continuación representá las palabras *SOY*, *POTUS*, *RUSO*, *ESCURRIDIZO* y *ALBAÑIL*.

X-RAYA					
A: X	F: XX-	K: X-XX	O: X----	T: X-X-X	Y: XX-X-
B: X-	G: XXX	L: XX--	P: X---X	U: X-XX-	Z: XX-XX
C: XX	H: X---	M: XX-X	Q: X--X-	V: X-XXX	
D: X--	I: X--X	N: XXX-	R: X--XX	W: XX---	
E: X-X	J: X-X-	Ñ: XXXX	S: X-X--	X: XX--X	

SOY:

POTUS:

RUSO:

ESCURRIDIZO:

ALBAÑIL:

2. Ahora codificá las palabras *ATA* y *FEA*. ¿Notás algún problema?

ATA:

FEA:

NOMBRE Y APELLIDO:

CURSO:

FECHA:

3. Proponé una sistema de codificación usando los símbolos **X** y **-**, en el que una secuencia de **X** y **-** tenga una y sólo una interpretación posible.

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

4. ¿Cómo se codifican ahora las palabras *ATA* y *FEA*? ¿Resolviste la ambigüedad?

ATA:

FEA:

SISTEMAS DE CODIFICACIÓN DE SÍMBOLOS

A lo largo de la historia, las computadoras utilizaron diferentes maneras para codificar textos en binario. Uno de los sistemas más difundidos es el sistema ASCII, que utiliza 7 dígitos binarios para representar cada símbolo. Esto permite codificar todas las letras del idioma inglés en mayúsculas y en minúsculas, los números del 0 al 9 y una variedad de símbolos especiales. Sin embargo, no permite representar otros alfabetos o letras del alfabeto castellano como la ñ. Para esto se extendió el sistema ASCII a 8 dígitos binarios y se crearon otras normas. Actualmente el sistema más usado es Unicode, que utiliza hasta 32 dígitos binarios.



NOMBRE Y APELLIDO:

CURSO:

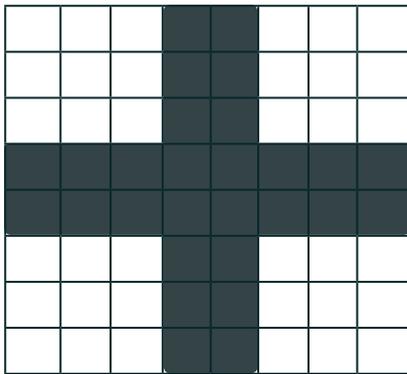
FECHA:

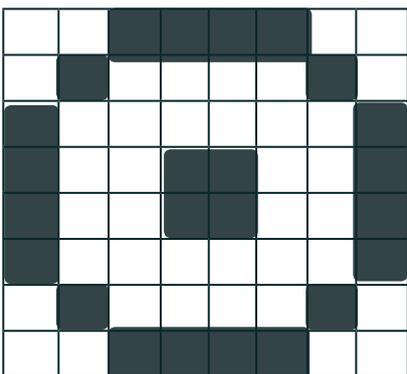
IMÁGENES EN BLANCO Y NEGRO

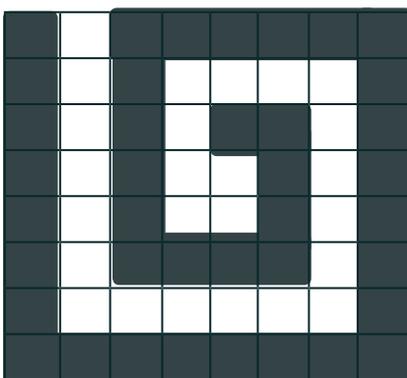
Los mapas de bits son una forma de codificar imágenes en blanco y negro en la que se emplean números binarios. El 0 se utiliza para representar un cuadrado blanco en la imagen y el 1 se utiliza para representar un cuadrado negro. En esta actividad vas a traducir algunas imágenes al lenguaje de las computadoras y vas a interpretar ese lenguaje para poder reconstruir otras imágenes.



1. Completá los mapas de bits de las siguientes imágenes.







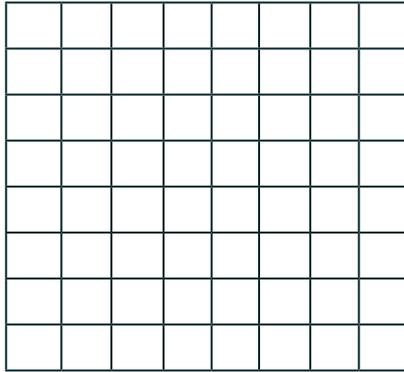
NOMBRE Y APELLIDO:

CURSO:

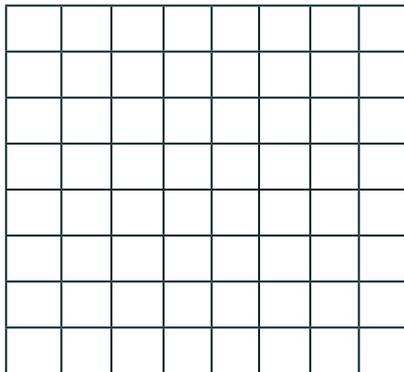
FECHA:

2. Seguí la representación binaria de las imágenes y dibujalas.

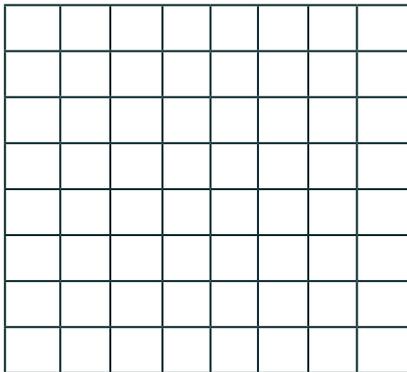
1 0 0 0 0 0 0 1
0 1 0 0 0 0 1 0
0 0 1 0 0 1 0 0
0 0 0 1 1 0 0 0
0 0 0 1 1 0 0 0
0 0 1 0 0 1 0 0
0 1 0 0 0 0 1 0
1 0 0 0 0 0 0 1

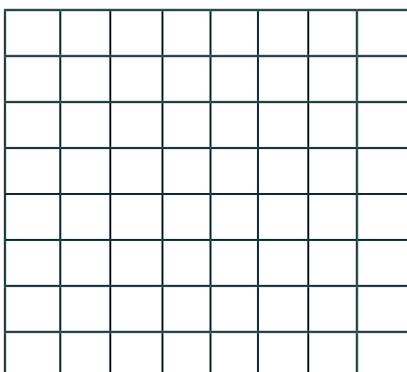


0 0 1 1 1 1 0 0
0 1 1 1 1 1 1 0
0 1 1 1 1 1 1 0
0 1 1 1 1 1 1 0
0 0 1 1 1 1 0 0
0 0 0 1 1 0 0 0
0 0 0 1 1 0 0 0
0 0 0 1 1 0 0 0



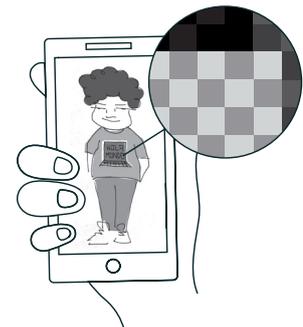
3. Ahora dibujá lo que quieras y escribí su representación como mapa de bits.





IMÁGENES DIGITALES

Cada imagen que ves en las pantallas de los dispositivos digitales está compuesta por puntos que se dibujan uno al lado del otro. En computación, a estos puntos los llamamos **píxeles**. El píxel es la unidad mínima de la composición de las imágenes digitales.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

BANDERAS DE COLORES

¿Alguna vez mezclaste diferentes colores de pintura para hacer otros nuevos? Es común utilizar rojo, amarillo y azul como tres colores **primarios** que se pueden mezclar para producir muchos más colores. Rojo y azul dan violeta, rojo y amarillo dan naranja, por ejemplo.



Las pantallas también se basan en la mezcla de tres colores, pero necesitan un conjunto diferente de colores primarios porque comienzan con una pantalla negra y van añadiendo color a ella a través de luces. Estas mezclas de colores se denominan **aditivas**. Cada píxel en una pantalla se compone típicamente de tres luces minúsculas: una roja, una verde y otra azul. Al aumentar y disminuir la cantidad de luz de cada uno de estas tres, se pueden hacer todos los colores diferentes. Esta mezcla de colores se conoce como RGB por sus siglas de las palabras en inglés: *Red* (rojo), *Green* (verde) y *Blue* (azul). Acá te presentamos la codificación de algunos colores que van a servirte para resolver esta actividad.

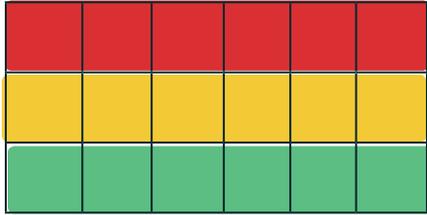
	R	G	B
ROJO	255	0	0
AZUL	0	0	255
AMARILLO	255	255	0
VERDE	0	255	0
CELESTE	116	169	218
BLANCO	255	255	255

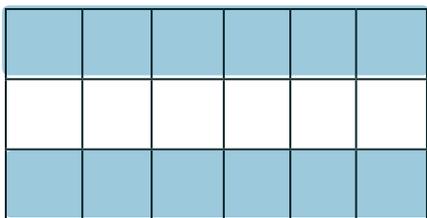
NOMBRE Y APELLIDO:

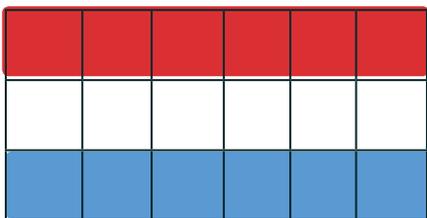
CURSO:

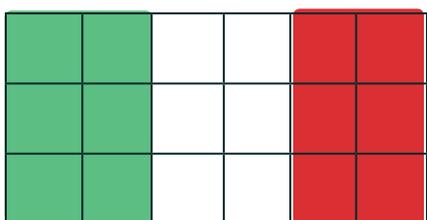
FECHA:

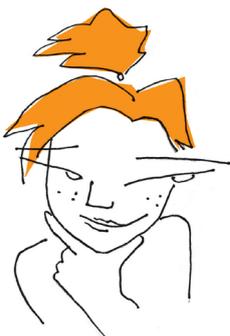
Por ejemplo, rojo se indica (255,0,0) y azul, (0,0,255). Ahora tenés que dar la codificación RGB de las banderas de Bolivia, Argentina, Paraguay y México.











¿SABÍAS QUE...

...la codificación por mapa de bits es la que usan los archivos con extensión **bmp**?

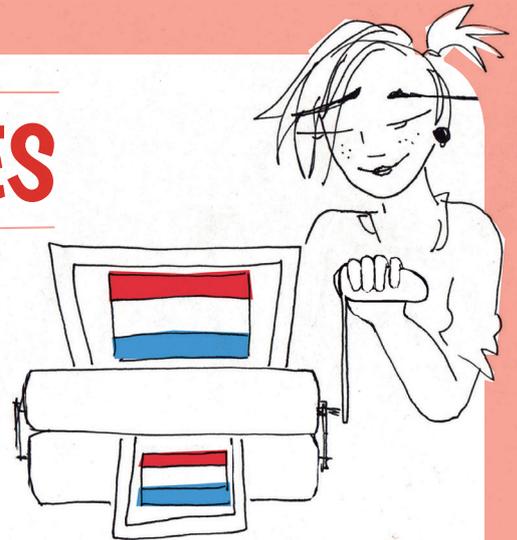
NOMBRE Y APELLIDO:

CURSO:

FECHA:

ENCOGEMOS IMÁGENES

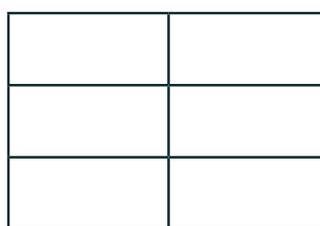
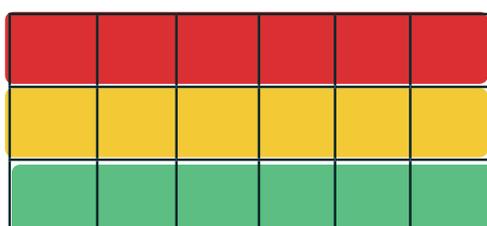
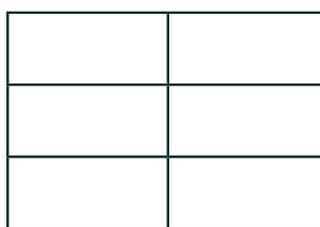
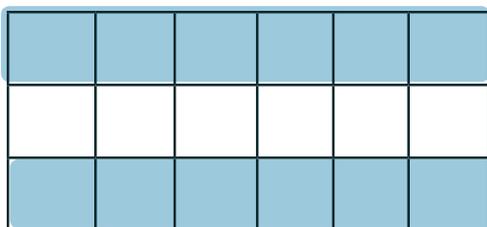
¿Alguna vez te pasó que quisiste mandar un archivo por mail y no pudiste porque era demasiado grande? ¿Sabés cómo hacen las computadoras cuando tienen que almacenar o transmitir enormes volúmenes de información? En esta actividad vas a usar un método que permite reducir el espacio requerido para representar imágenes digitales.



Acá te presentamos los códigos RGB de los colores que vas a necesitar para resolver la actividad. Recordá que, por ejemplo, para indicar el código del rojo se escribe: (255,0,0).

	R	G	B
ROJO	255	0	0
BLANCO	255	255	255
AMARILLO	255	255	0
VERDE	0	255	0
CELESTE	116	169	218
NEGRO	0	0	0

1. ¿Cómo sería la codificación RLE de las banderas de Argentina, Bolivia y México?



NOMBRE Y APELLIDO:

CURSO:

FECHA:

2. ¿Cuántos bytes necesitaste para codificar cada una? ¿Cuántos hubieran hecho falta si no hubieses usado RLE?

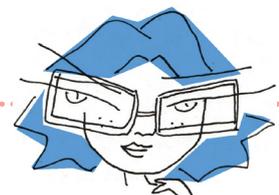
3. ¿Se te ocurre por qué no todas las codificaciones necesitaron la misma cantidad de espacio?

4. ¡Ahora te toca descomprimir una imagen! ¿Qué bandera se formó?

6	(0,0,0)						
6	(255,0,0)						
6	(255, 255,0)						

COMPRESIÓN RLE

¿Sabías que RLE pertenece a una familia de métodos de compresión que no pierden información? Todos ellos permiten recuperar la información original tal cual era antes de comprimirla. Los archivos .zip y .rar también se generan usando algoritmos de esta familia.



06

LA

COMPUTADORA

NOMBRE Y APELLIDO:

CURSO:

FECHA:

¿QUÉ SON LAS COMPUTADORAS?



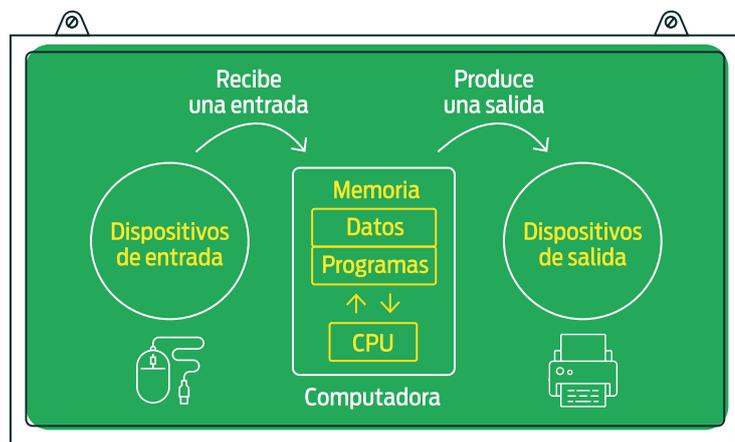
¿Qué caracteriza a las computadoras? ¿Las hay de diferentes formas y tamaños? ¿Qué cosas tienen en común todas ellas?

1. Completá la tabla con cinco computadoras que no sean ni las portátiles ni las de escritorio. Para cada una de ellas identificá, al menos, un dispositivo de entrada y uno de salida.

MÁQUINA QUE ES O QUE CONTIENE UNA COMPUTADORA	DISPOSITIVO DE ENTRADA	DISPOSITIVO DE SALIDA

ARQUITECTURA DE VON NEUMANN

En 1945, el matemático de origen austrohúngaro John von Neumann presentó un modelo conceptual de arquitectura de computadoras, cuyo diseño sigue vigente en las computadoras modernas. En este modelo, conocido como *arquitectura de von Neumann*, una computadora está compuesta por una unidad central de procesamiento –que se encarga de ejecutar las instrucciones de los programas–, una memoria –en la que se almacenan los datos y los programas–, y dispositivos de entrada y salida que permiten el ingreso y egreso de datos.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

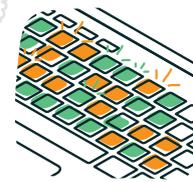
2. Explorá el siguiente anuncio. Relacioná los elementos de la computadora publicitada con los componentes descritos en el modelo de von Neumann.

Bet&RobPro

La nueva computadora *Bet & Rob Pro* cuenta con un microprocesador Intel Core i7-7700HQ que maneja cargas de trabajo pesadas en minutos. Este y la RAM DDR2-667 de 32 Gigabytes le permitirán trabajar con programas complejos y múltiples pestañas, incluso ejecutando varias aplicaciones a la vez.



TOUCHPAD HIPERSENSIBLE



TECLADO LUMINOSO



LECTOR DE HUELLAS

Además, tiene *touchpad* hipersensible, teclado luminoso y lector de huellas dactilares.

Con la pantalla táctil *full HD* podrá comandar la máquina usando los diez dedos de la mano, y la cámara frontal de 8 megapíxeles le permitirá sacarse fotos cuando esté procrastinando.

De regalo, unos parlantes holofónicos y una impresora láser color.

CPU	
MEMORIA	
DISPOSITIVOS DE ENTRADA	
DISPOSITIVOS DE SALIDA	

NOMBRE Y APELLIDO:

CURSO:

FECHA:

UNA COMPUTADORA A CIELO ABIERTO

¿Cómo es una computadora por dentro? Seguí las indicaciones para investigarlo. ¡A arremangarse y meter mano, que así también se aprende!

1. Asegurate de que la computadora no esté conectada al tomacorriente y desenchufale todos los cables, de modo que quede solo el gabinete. Desatornillá todos los tornillos y retirá la tapa lateral. ¿Reconocés algún componente? ¿La memoria? ¿El procesador?

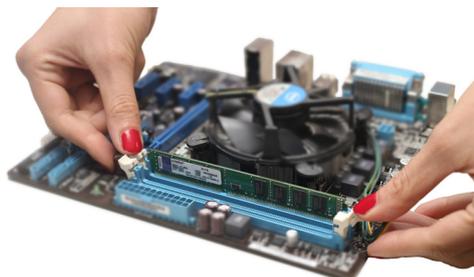


2. Retirá un módulo de memoria. ¿Para qué son los conectores dorados que hay en la parte inferior? ¿Y qué son los bloques negros?



PARA SACAR UN MÓDULO DE MEMORIA

Los módulos de memoria se encuentran en unas ranuras de la placa madre. Para sacar uno, primero se deben empujar hacia afuera los ganchos de retención ubicados en los extremos de las ranuras. Luego, se puede retirar tirando hacia arriba.



NOMBRE Y APELLIDO:

CURSO:

FECHA:

3. ¿Podés ver la unidad central de procesamiento (o procesador)?
¿Por qué? Mirá la imagen de la derecha. ¿Para qué sirven los
contactos dorados?



4. Observá qué unidad de almacenamiento encontrás. ¿De qué
tipo es? ¿Está montada sobre la placa madre?



DISCOS

Existen distintas tecnologías de discos; los más habituales en la actualidad son los **discos rígidos rotacionales HDD** (por la sigla en inglés de *Hard Disk Drive*) y los **discos de estado sólido SSD** (por la sigla en inglés de *Solid State Drive*). Estos últimos son más modernos, más rápidos y más caros. Los rotacionales tienen un cabezal y platos magnéticos giratorios, en los que se guarda información. Por su parte, los de estado sólido no tienen componentes mecánicos; su composición interna es más parecida a los de una memoria USB.



5. ¿Encontraste algún dispositivo de entrada y salida?

6. ¿Para qué sirven los puertos que asoman por las perforaciones que tiene el gabinete?
¿Cuáles encontraste?

7. ¿En qué se diferencian los puertos USB del resto de los puertos que hay en el gabinete?

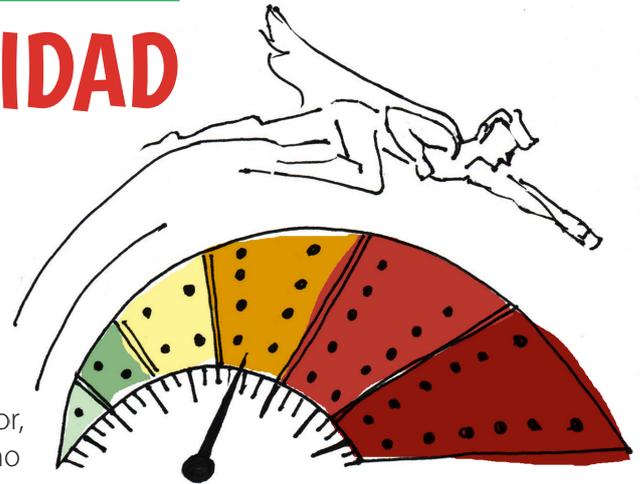
8. Volvé a ensamblar la computadora y dejala como al principio.

NOMBRE Y APELLIDO:

CURSO:

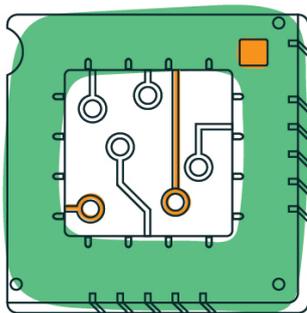
FECHA:

ZOOM IN A LA UNIDAD CENTRAL DE PROCESAMIENTO



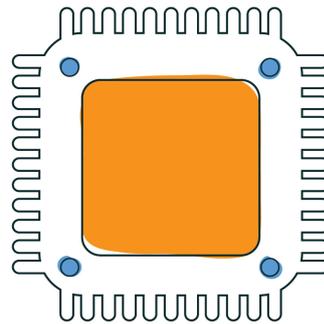
Cuándo miramos el anuncio de venta de un procesador, ¿qué quieren decir los datos que se mencionan? ¿Cómo inciden en el rendimiento de nuestras computadoras?

1. Mirá los anuncios publicitarios y completá las tres primeras filas de la tabla que se encuentra en la siguiente página.



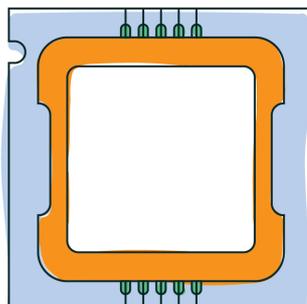
Intel® Core™ i7

- 6 núcleos 4.5 GHz
- Memoria compatible DDR3/DDR4.



AMD Ryzen 7 / 2700

La verdadera inteligencia, compuesta por 8 núcleos, una frecuencia de reloj de 4.1 GHz, compatible con memorias DDR4.



Qualcomm Snapdragon 835

Un 30% más fino y eficiente, para un presente de dobles cámaras y realidad virtual. Los 8 núcleos de 2.45 GHz se integran con memorias LPDDR4, lo que genera máxima eficiencia en dispositivos móviles.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

MARCA	MODELO	CANTIDAD DE NÚCLEOS	FRECUENCIA	COMPATIBILIDAD CON MEMORIA

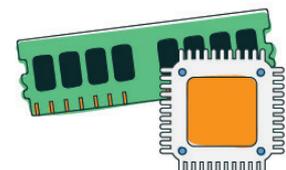
2. Buscá en Internet dos anuncios más y completá las últimas dos filas.

3. ¿Cuál o cuáles de los procesadores de la tabla tiene capacidad de ejecutar más instrucciones en forma simultánea? ¿Por qué, de qué depende?

4. ¿Qué indica la frecuencia de un procesador? ¿Tiene alguna incidencia en la velocidad a la que funciona una computadora?

COMPATIBILIDAD CON LA MEMORIA

Para poder funcionar juntos, el procesador y la memoria deben ser compatibles. Como sucede con casi cualquier componente electrónico, también entre las memorias existen diferentes generaciones tecnológicas. Un procesador, en general, admitirá memorias de una generación. Si querés ampliar la cantidad de memoria de tu computadora, asegurate de comprar una compatible.



¿CUÁNTO CABE EN LA MEMORIA?

ANEXO

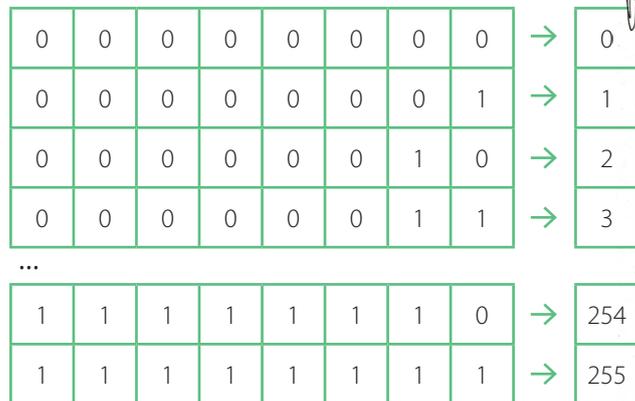


Cuando hablamos de memoria, nos referimos a una serie de componentes físicos –de *hardware*– que tienen la capacidad de representar información. Esto incluye tanto a la memoria RAM, como a los discos rígidos, los dispositivos portátiles de almacenamiento, etc.

¿CÓMO SE ALMACENA INFORMACIÓN EN UN DISPOSITIVO DE MEMORIA?

Una memoria se puede representar como una gran tira de celdas contiguas.

En cada una se puede almacenar uno de dos valores: o bien un cero o bien un uno. Esta es la mínima unidad de información que una computadora puede representar y se llama *bit*. En general, no se piensa en términos de bits, sino que se usan unidades de medida más grandes. Por ejemplo, un *byte* equivale a 8 bits. Como cada bit puede tener un cero o un uno, al considerar todas las posibles combinaciones de ellos agrupados de a 8, podemos ver que en un byte se pueden almacenar 256 valores distintos.



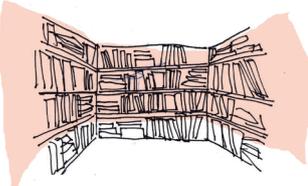
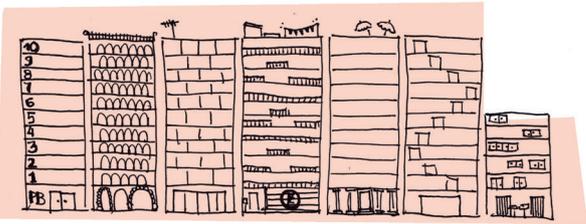
BITS, BYTES Y LO DE MÁS ALLÁ

Así como para medir longitudes hay distintas unidades de medida, como por ejemplo milímetros, centímetros y metros, lo mismo sucede cuando queremos medir cantidades de información. Estas son algunas de ellas.

UNIDAD	Byte	Kilobyte (KB)	Megabyte (MB)	Gigabyte (GB)	Terabyte (TB)
TAMAÑO	8 bits	1024 bytes	1024 KB	1024 MB	1024 GB

El uso del número 1024 como factor de multiplicación entre distintas unidades de medida se debe a que, por ser una potencia de 2 ($2^{10} = 1024$), es de fácil manipulación para una computadora.

Para dimensionar cuánta información cabe en un dispositivo de memoria, mirá la siguiente analogía.

CANTIDAD DE MEMORIA	PUEDE CONTENER...
1 byte	 <p>Una letra de un libro</p>
1 KB	 <p>Una página de un libro</p>
1 MB	 <p>Un libro de 1024 páginas</p>
1 GB	 <p>Una habitación con 1024 libros</p>
1 TB	 <p>Los libros contenidos en 6 edificios de 10 pisos y uno de 5 pisos con cuatro departamentos de cuatro habitaciones por piso</p>



Esto entra en un disco de un Terabyte.

NOBLEZA OBLIGA

El espacio que se utiliza para codificar un carácter depende del sistema de codificación usado. Por ejemplo, algunas versiones de UNICODE llegan a usar cuatro bytes. Pero la codificación ASCII, que permite codificar el alfabeto latino, utiliza un solo byte para cada carácter. O sea que, si todos los libros están en castellano, ¡la analogía es precisa!



NOMBRE Y APELLIDO:

CURSO:

FECHA:

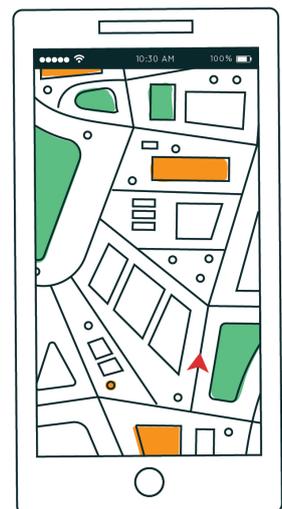
LA CACHÉ

¿Alguna vez tu teléfono inteligente te sugirió "borrar la caché"? Todo muy lindo, pero... ¿qué es la caché?

1. Frecuentemente utilizamos aplicaciones de mapas en los teléfonos inteligentes. Cuando buscamos algún lugar, ¿cómo hace el teléfono para mostrarnos el mapa? ¿De dónde saca la información?



- a. Deshabilita de tu teléfono la conexión wifi y el acceso a la red de datos de telefonía celular. Después abrí la aplicación de mapas que uses habitualmente y buscá la dirección de tu casa. ¿Apareció el mapa en la pantalla? ¿Dónde están guardados esos mapas?



- b. Ahora buscá una ciudad distante, que nunca hayas visitado. ¿Apareció el mapa en la pantalla? ¿Cuál es la diferencia de esta búsqueda con la del punto anterior? ¿Dónde está guardado el mapa de esta ciudad?

CACHING

La idea de *caching* es que, entre el procesador y un dispositivo de memoria, haya otra memoria más pequeña a la que se pueda acceder a mucha mayor velocidad. En ella se conservan los datos más frecuentemente solicitados por los programas y, de esta manera, se reduce significativamente el tiempo que espera el procesador entre que pide esos datos y los obtiene. Algunos ejemplos son un disco para algo almacenado en la nube, la memoria RAM para algo almacenado en el disco, etc.



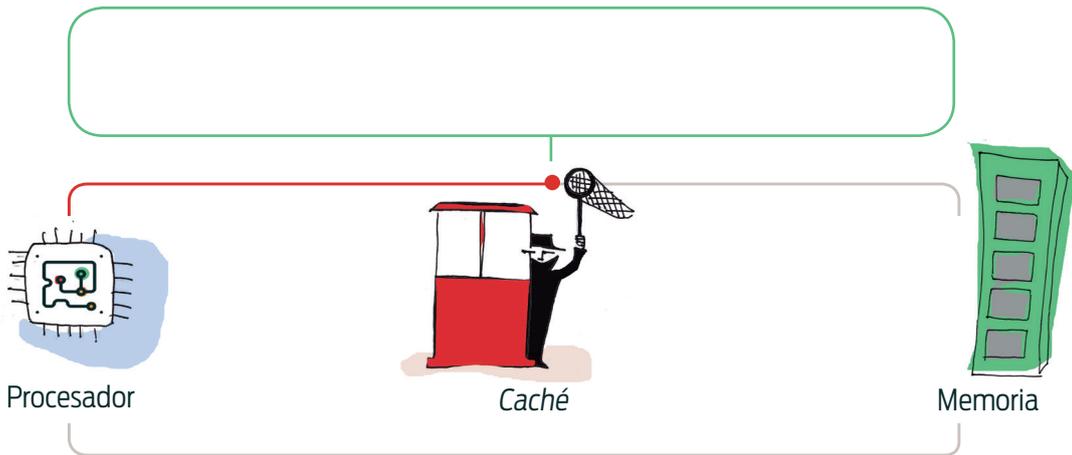
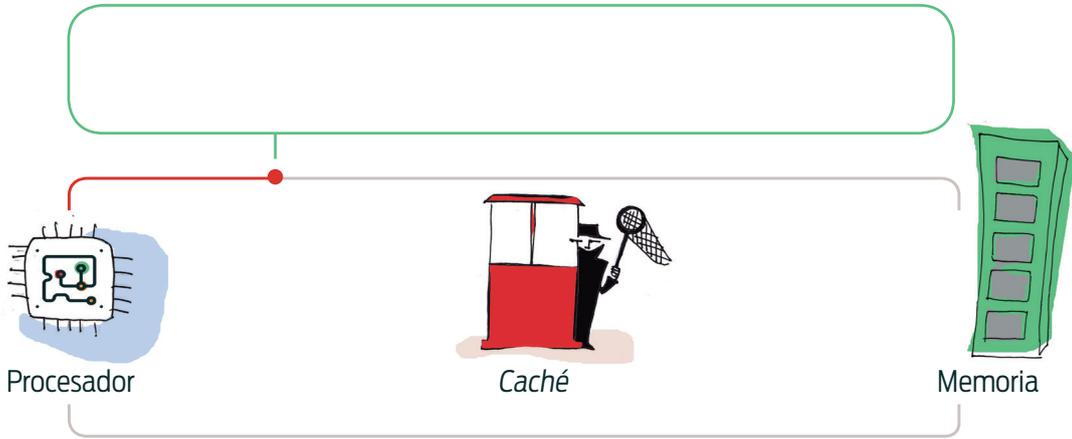
NOMBRE Y APELLIDO:

CURSO:

FECHA:

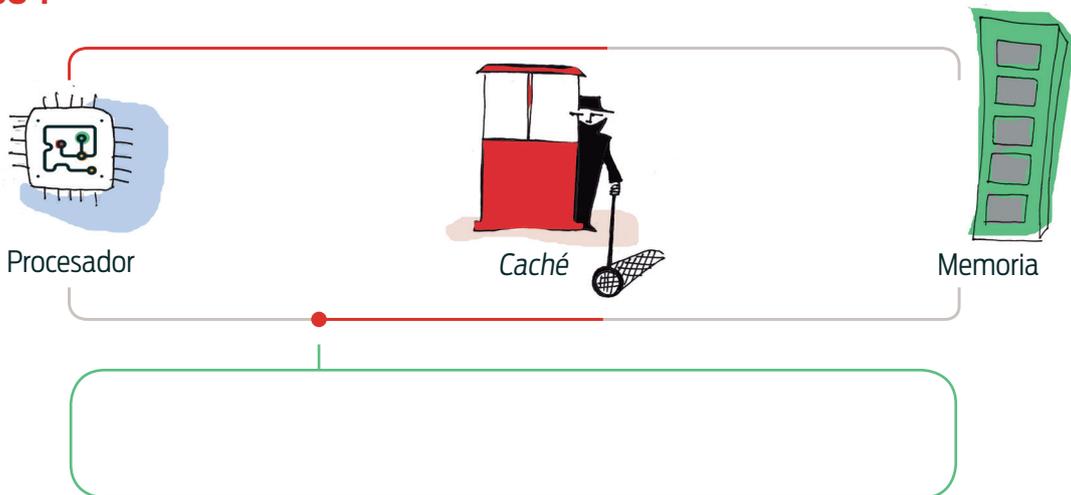
2. Cuando un programa solicita un dato, pueden suceder dos cosas: o bien el dato está en la caché o bien no lo está.

a. Completá en el siguiente esquema lo que va sucediendo desde que el procesador requiere un dato de la memoria hasta que lo obtiene.



En esta instancia pueden suceder dos cosas: o bien el dato está en la caché o bien no lo está.

Caso 1



NOMBRE Y APELLIDO:

CURSO:

FECHA:

Caso 2



NOMBRE Y APELLIDO:

CURSO:

FECHA:

b. Cuando un programa escribe un dato que es interceptado y almacenado en la *caché*, ¿debe también copiarse en la memoria? ¿Por qué?

¿LA PEGA O LA PIFIA?

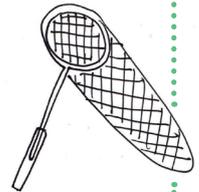
Cuando un dato solicitado se encuentra en la *caché* se dice que la *caché* la *pegó* (o más comúnmente *caché hit*, por su forma en inglés). En cambio, cuando esto no sucede y hay que ir a buscarlo a la memoria, que la *caché* la *pijó* (o *caché miss*).



¿SABÍAS QUÉ...

...los procesadores tienen (muy) pequeños módulos de memoria internos llamados *registros* a los que puede acceder a altísima velocidad?

En la actualidad, un procesador no suele tener más de 32, cada uno de 32 bytes. ¡En ellos se "cachea" información de la memoria RAM!



BAJAMOS HASTA EL LENGUAJE DE MÁQUINA

ANEXO

Cada modelo de computadora tiene componentes electrónicos que son capaces de llevar a cabo operaciones muy simples. Por ejemplo, operaciones aritméticas (como sumar y multiplicar), operaciones lógicas, leer un dato en la memoria, escribir un dato en la memoria, etc. Al programar, ¡por suerte no nos hace falta pensar en esto! En su lugar, usamos lenguajes que son mucho más expresivos, llamados **lenguajes de alto nivel**, que nos permiten razonar en términos del problema que queremos resolver.

Ahora bien, ¿cómo se ejecutan estos programas si la computadora hace tan pocas y rudimentarias operaciones?

```
void saludarEstudiantes() {
    printf("¡Hola, estudiantes!");
}
```



COMPILADOR



COMPILADOR

Un **compilador** es un programa que toma como entrada un programa escrito en un lenguaje de programación de alto nivel y genera como salida otro programa en lenguaje ensamblador que es semánticamente equivalente; es decir, que hace exactamente lo mismo.

ENSAMBLADOR

Un programa en lenguaje ensamblador sigue siendo un archivo con texto. Para ejecutar un programa, lo primero que hay que hacer es cargarlo en la memoria, de donde el procesador irá leyendo sus instrucciones una por una. Sin embargo, en la memoria solo se guardan bits. El **ensamblador** es un programa que toma como entrada un programa en lenguaje ensamblador y genera como salida otro equivalente en **lenguaje de máquina**.



ENSAMBLADOR



```
movl $0xFF001122, %eax
addl %ecx, %edx
xorl %esi, %esi
pushl %ebx
movl 4(%esp), %ebx
leal (%eax, %ecx, 2), %esi
cmpl %eax, %ebx
jnae foo
retl
```

1	0	1	1	1
0	0	0	1	0
1	1	1	1	1
1	1	0	0	1



LENGUAJE DE ALTO NIVEL

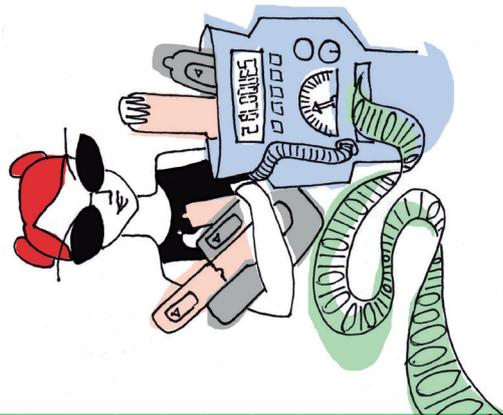
Habitualmente, al construir un programa se utiliza un **lenguaje de alto nivel**. Estos lenguajes permiten que un programador razone sobre el problema que quiere resolver, abstraendo por completo el funcionamiento interno de los componentes de *hardware*.

LENGUAJE ENSAMBLADOR

Cada computadora tiene componentes que realizan algunas operaciones simples: aritméticas, lógicas, de lectura y escritura en la memoria, y algunas más. El **lenguaje ensamblador** de una computadora tiene instrucciones que se corresponden con lo que sus componentes de *hardware* son capaces de llevar a cabo.

LENGUAJE DE MÁQUINA

El **lenguaje de máquina** está formado por una serie de instrucciones codificadas con números binarios que pueden tanto cargarse en la memoria como ser interpretadas por el *hardware* de la computadora.



BIEN ADENTRO DE LA COMPUTADORA

¿Cómo hace una computadora para ejecutar cada instrucción de un programa? ¿Qué componentes internos intervienen? Metámonos bien adentro para ver qué es lo que pasa por allá abajo.



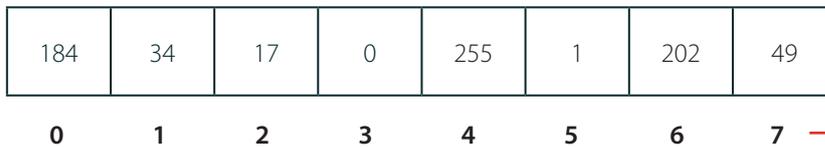
RECORDATORIO

Para que un programa pueda ejecutarse, en primer lugar se compila y se obtiene un programa semánticamente equivalente en lenguaje ensamblador, es decir, con instrucciones que se corresponden con las operaciones que los circuitos del *hardware* de la máquina pueden realizar; en segundo lugar, el programa se ensambla para obtener código de máquina –instrucciones codificadas en un sistema binario que pueden almacenarse en la memoria de la computadora–.



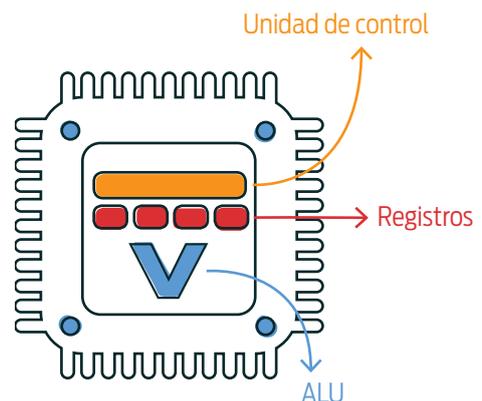
DIRECCIONES DE MEMORIA

En la memoria, los bits se agrupan de a 8 y, por lo tanto, se la concibe como una secuencia de bytes. Cada byte de la memoria puede identificarse con un número, que se corresponde con la ubicación en la que se encuentra, comenzando a contar desde 0. A este número se lo conoce como **dirección de memoria**.



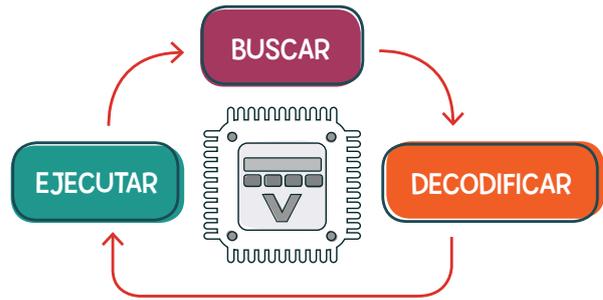
LA CPU POR DENTRO

La unidad central de procesamiento, internamente, tiene: (i) una unidad aritmética lógica (ALU), que cuenta con circuitos para realizar operaciones aritméticas y lógicas simples; (ii) registros, que son pequeñas unidades de memoria, en los que almacena datos; y (iii) una unidad de control, que dirige el proceso de ejecución de las instrucciones de un programa.



CICLO DE INSTRUCCIÓN

Un ciclo de instrucción comprende las tres etapas que lleva a cabo la unidad central de procesamiento para ejecutar una instrucción del lenguaje de máquina. La unidad de control se encarga de (i) **buscar** la instrucción en la memoria; (ii) **decodificar** la instrucción; y (iii) enviar las señales necesarias para que se pueda **ejecutar** la instrucción (dependiendo de cuál sea la instrucción, esto puede ser activar la ALU, cargar datos en registros, enviar señales a los circuitos que comunican la CPU con la memoria, etc.).



La unidad central de procesamiento repite este ciclo para cada instrucción que tiene que ejecutarse, hasta que el programa finalice su ejecución.

¿CÓMO SABE LA CPU CUÁL ES “LA PRÓXIMA INSTRUCCIÓN”?

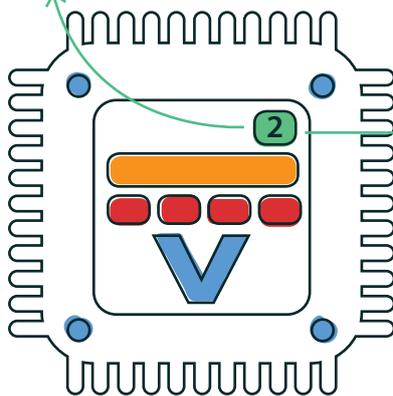
Hay un registro interno de la unidad central de procesamiento, que se llama **puntero de instrucción**, que siempre tiene almacenada la dirección de memoria a la que hay que ir a buscar la próxima instrucción que debe ejecutarse. Antes de comenzar un nuevo ciclo de instrucción, la unidad de control actualiza su valor de modo que pase a contener la dirección de la próxima instrucción del programa que hay que ir a buscar a la memoria. En general, este valor será la siguiente posición de la memoria, aunque esto no siempre es así; por ejemplo, cuando se está en presencia de una alternativa condicional o una repetición.

184	34	17	0	255	1	202	49
-----	----	----	---	-----	---	-----	----

Secuencia de bytes

0 1 2 3 4 5 6 7

Direcciones de memoria



Puntero de instrucción



NOMBRE Y APELLIDO:

CURSO:

FECHA:

LA COMPUTADORA EN ACCIÓN

Cada vez que interactuamos con una computadora hacemos que se ejecuten programas: usamos una aplicación de mensajería instantánea, una que muestra fotos, un reproductor de música, etc. ¿Cómo hace la computadora para ejecutarlos? En esta actividad iremos a fondo para ver cómo se ejecutan los programas en lenguaje de máquina. ¡Salió la ultra fina *Bet & Rob air!* Mirá las especificaciones de *hardware*:



Memoria

Tiene una memoria de 18 bytes; los primeros 9 están reservados para programas y los restantes se usan para almacenar datos.

Registros de la unidad central de procesamiento

Posee 2 registros internos, **REG₁** y **REG₂**, que los programas usan para escribir y leer datos. Además, como cualquier otra computadora, también tiene un registro que actúa de puntero de instrucción: **PI**. Este, sin embargo, no puede ser mencionado en los programas.

CPU		MEMORIA	
			
REG₁	PI	ESPACIO DE PROGRAMA	ESPACIO DE DATOS
<input type="text"/>	→	0	9
REG₂		1	10
<input type="text"/>		2	11
		3	12
		4	13
		5	14
		6	15
		7	16
		8	17

NOMBRE Y APELLIDO:

CURSO:

FECHA:

Lenguaje ensamblador

A continuación está el conjunto de instrucciones de su lenguaje ensamblador:

ASIGNAR [r] [n]: escribe en el registro r el número n .

INCREMENTAR [r]: incrementa en 1 el valor contenido en el registro r .

DECREMENTAR [r]: decrementa en 1 el valor contenido en el registro r .

MOVER_DE_MEMORIA_A_REGISTRO [r] [n]: escribe en el registro r el contenido de la celda de memoria cuya dirección es n .

MOVER_DE_REGISTRO_A_MEMORIA [r] [n]: lee el contenido del registro r y lo escribe en la posición n de la memoria.

SALTO_CONDICIONAL [r] [n] [i]: lee el contenido del registro r , lo compara con el número n y, si son iguales, establece que la siguiente instrucción a ejecutar es la i -ésima (es decir, la que está en la posición i en el programa); si no, el flujo de ejecución continúa con la instrucción que se encuentra inmediatamente a continuación.

SALTO_INCONDICIONAL [i]: establece que la siguiente instrucción a ejecutar es la i -ésima (es decir, la que está en la posición i en el programa).

DETENER: detiene la ejecución del programa.

1. Seguí paso a paso cómo evoluciona el estado del sistema a medida que se ejecuta el siguiente programa. Los valores iniciales de la memoria y los registros se observan a continuación.

CPU		MEMORIA	
REG ₁	PI	ESPACIO DE PROGRAMA	ESPACIO DE DATOS
0	→	0 MOVER_DE_MEMORIA_A_REGISTRO REG ₁ 9	9 2
		1 ASIGNAR REG ₂ 0	10 0
5		2 SALTO_CONDICIONAL REG ₁ 0 7	11 0
		3 INCREMENTAR REG ₂	12 0
		4 INCREMENTAR REG ₂	13 0
		5 DECREMENTAR REG ₁	14 0
		6 SALTO_INCONDICIONAL 2	15 0
		7 MOVER_DE_REGISTRO_A_MEMORIA REG ₂ 10	16 0
		8 DETENER	17 0

NOMBRE Y APELLIDO:

CURSO:

FECHA:

¿Qué hubiese sucedido si, inicialmente, en la posición 9 de la memoria hubiese habido un 3? ¿Y si hubiera un 10?

¿Qué hace el programa?

2. Realizó el seguimiento de la ejecución de este otro programa.

CPU		MEMORIA			
REG ₁	PI	ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
16	→	0	MOVER_DE_MEMORIA_A_REGISTRO REG ₁ 9	9	2
		1	MOVER_DE_MEMORIA_A_REGISTRO REG ₂ 10	10	4
12		2	MOVER_DE_REGISTRO_A_MEMORIA REG ₁ 10	11	0
		3	MOVER_DE_REGISTRO_A_MEMORIA REG ₂ 9	12	0
		4	DETENER	13	0
		5		14	0
		6		15	0
		7		16	0
		8		17	0

NOMBRE Y APELLIDO:

CURSO:

FECHA:

¿Qué hace el programa?

¿Tiene alguna importancia el valor inicial de los registros? ¿Qué pasaría si al iniciar la ejecución del programa, el contenido de las direcciones de memoria 9 y 10 fuese otro?

3. Ahora fijate qué pasa con este.

CPU		MEMORIA			
REG ₁	PI	ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	→	0	ASIGNAR REG ₁ 9	9	2
		1	SALTO_CONDICIONAL REG ₁ 18 5	10	4
5		2	MOVER_DE_REGISTRO_A_MEMORIA REG ₁ REG ₁	11	0
		3	INCREMENTAR REG ₁	12	0
		4	SALTO_INCONDICIONAL 1	13	0
		5	DETENER	14	0
		6		15	0
		7		16	0
		8		17	0

NOMBRE Y APELLIDO:

CURSO:

FECHA:

¿Cómo queda la memoria al finalizar la ejecución del programa?

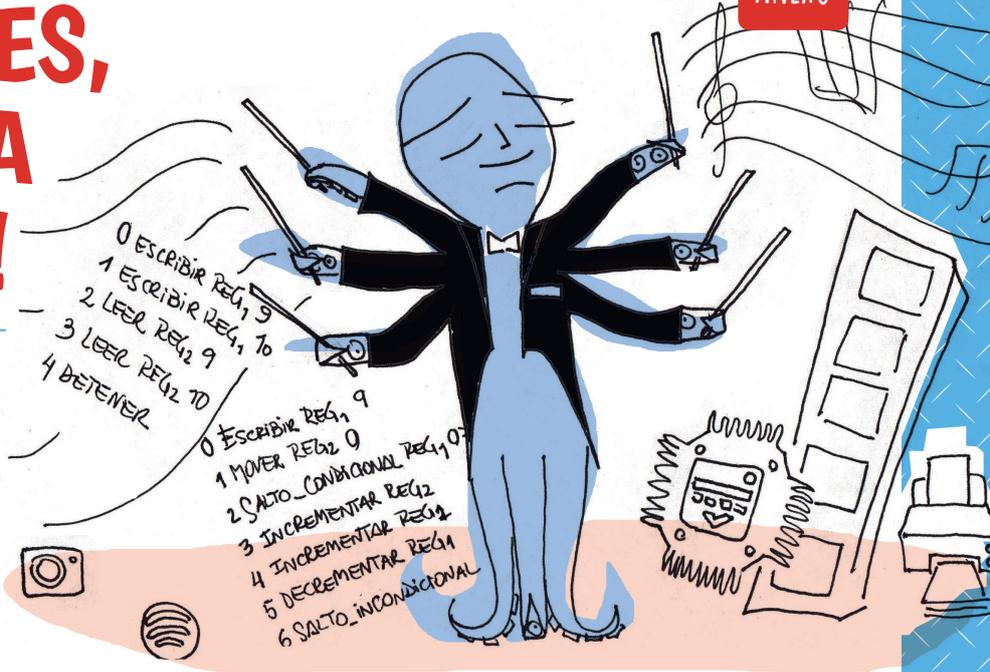
¿Tiene alguna incidencia el contenido inicial de la memoria? ¿Y el valor de los registros?

07

SISTEMAS OPERATIVOS

CON USTEDES, ¡EL SISTEMA OPERATIVO!

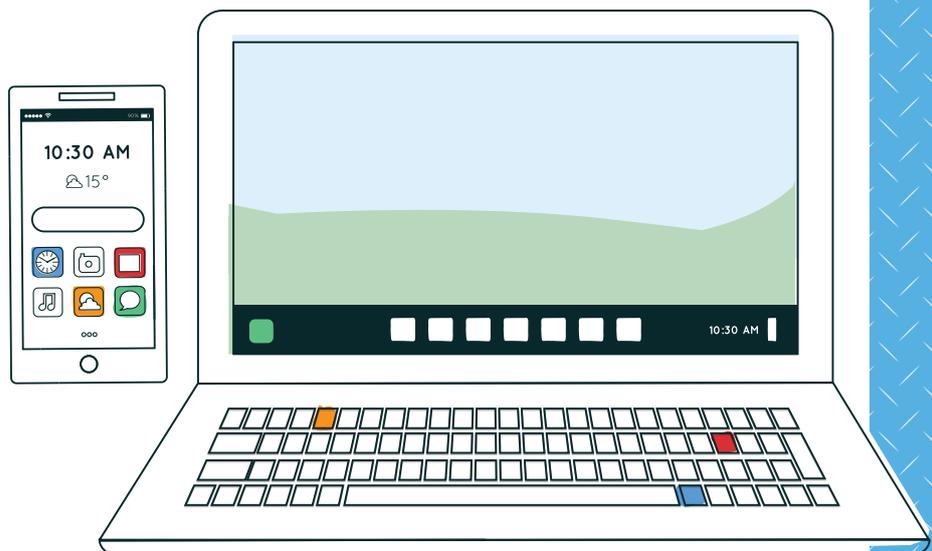
¿Por qué, cuando encendemos una computadora, aparece una interfaz que nos permite operarla? ¿Quién se ocupa de permitirnos organizar información en archivos y carpetas? ¿Cómo se lleva a cabo la interacción entre los programas y los dispositivos de hardware? ¿Por qué podemos ejecutar muchos programas al mismo



tiempo? Aunque solemos dar por sentadas estas cosas sin cuestionarlas, hay piezas de software que se ocupan de que todo esto sea posible: en su conjunto se las conoce como sistema operativo. El sistema operativo es, de algún modo, ¡el director de orquesta que hace que en una computadora todo acontezca!

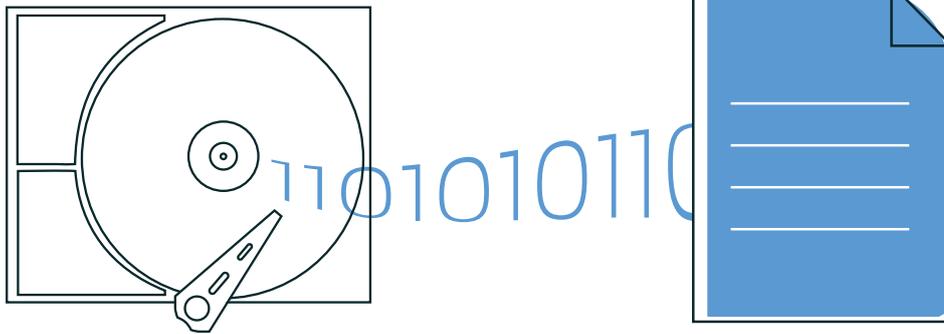
ENTORNO DE TRABAJO

Cuando prendemos nuestros dispositivos nos encontramos con muchos íconos, botones e información –entre otras cosas–, que nos permiten manejarlos. Ya se trate de portátiles, de tabletas o teléfonos inteligentes, tenemos la posibilidad de ejecutar programas, configurar opciones y leer información. ¡Esto no sucede porque sí! Hay un conjunto de programas llamado *sistema operativo* –que, dicho sea de paso, fue programado por personas–, que comienza a correr no bien encendemos nuestras computadoras y que a nosotros, como usuarios, nos brinda un entorno agradable para poder operarlas.



¿QUÉ ES UN ARCHIVO?

Con mucha frecuencia, usamos una computadora para ver fotos, escribir textos, escuchar música, ver videos, etc., que están guardados en archivos en algún medio de almacenamiento, como por ejemplo un disco rígido. Pero los discos, usando un cabezal, solo saben leer, escribir y borrar bits en los platos que hay en su interior. ¡No tienen la menor idea de qué es un archivo! Entonces, ¿qué son los archivos? Los **archivos** no son más que **abstracciones** que nos provee el sistema operativo para que nosotros (y los programas) podamos agrupar lógicamente información que nos resulte de interés y manipularla. Ni más ni menos.



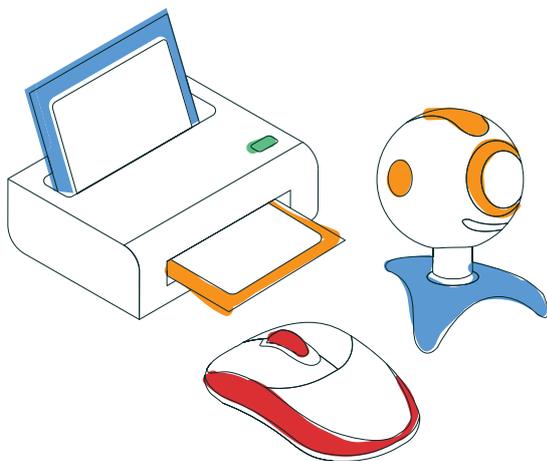
SISTEMA DE ARCHIVOS

Una función clave de un sistema operativo es proveer una interfaz limpia y clara para que podamos crear, leer y modificar archivos, además de organizarlos en carpetas. Al conjunto de programas que nos permite hacerlo se lo conoce como *sistema de archivos*. El sistema operativo tiene registro sobre en qué lugar del disco está cada archivo, cómo es la organización en carpetas, etc.

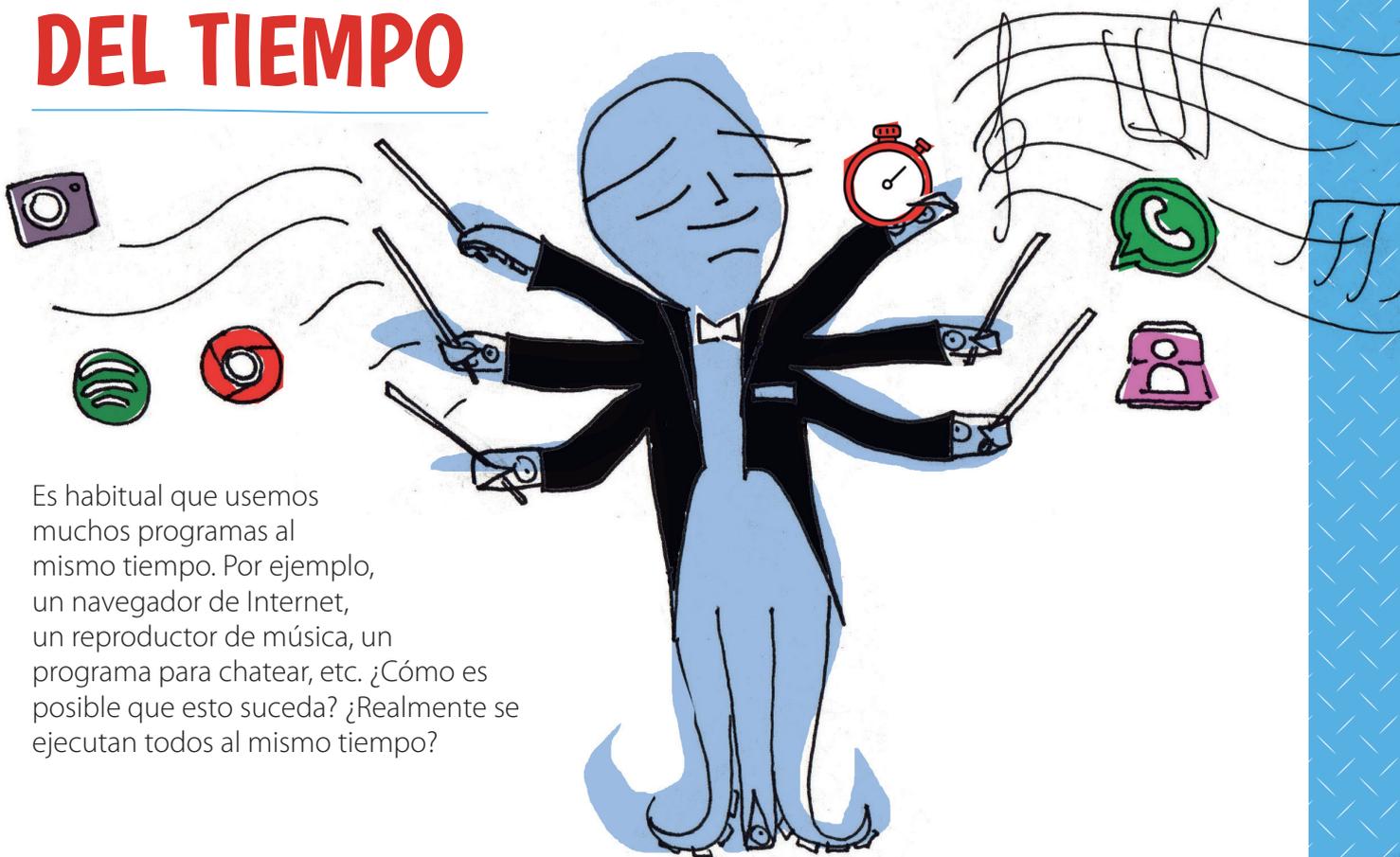


CONTROLADORES DE HARDWARE

No es extraño que un manual que describa cómo interactuar con un dispositivo de *hardware* –como un disco, una impresora, etc.– tenga más 500 páginas. ¡Ningún programador en su sano juicio querría leerlo para hacer algo tan básico como crear un archivo o mandar a imprimir un documento! Por suerte, hay piezas de *software* llamadas *controladores* –o *drivers*, en inglés–, que se ocupan de la interacción con el *hardware*. Los controladores proporcionan una interfaz sencilla para comunicarnos con los componentes físicos de una computadora, sin necesidad de entrar en detalles. Los sistemas operativos contienen varios de ellos y, además, cuando incorporamos nuevos dispositivos a nuestras computadoras –como, monitores, escáneres, etc.– también es posible agregar controladores para estos. De este modo, al programar, ¡podemos interactuar con el *hardware* de un modo claro, simple y lindo, lo que, de otro modo, sería oscuro, difícil y monstruoso!



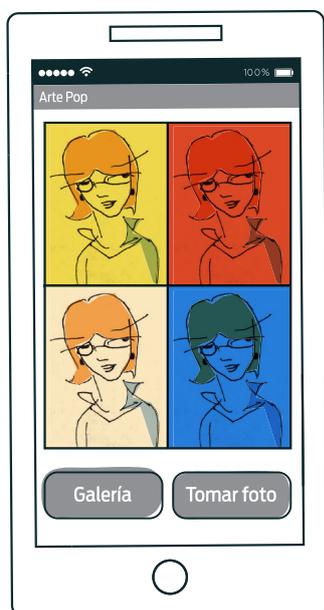
ADMINISTRACIÓN DEL TIEMPO



Es habitual que usemos muchos programas al mismo tiempo. Por ejemplo, un navegador de Internet, un reproductor de música, un programa para chatear, etc. ¿Cómo es posible que esto suceda? ¿Realmente se ejecutan todos al mismo tiempo?

PROGRAMAS Y PROCESOS

Los programas, escritos en algún lenguaje de programación, describen el comportamiento esperado por parte de una computadora. A su vez, cuando se ejecutan, se siguen paso a paso las instrucciones del programa. A los **programas** en ejecución se los llama **procesos**. Los procesos son semejantes a entidades “vivas”: un proceso “nace” –comienza a ejecutarse–, “se desarrolla” –avanza en su ejecución– y “muere” –termina de ejecutarse–. Por su parte, los programas son solo descriptivos; son entidades estáticas.



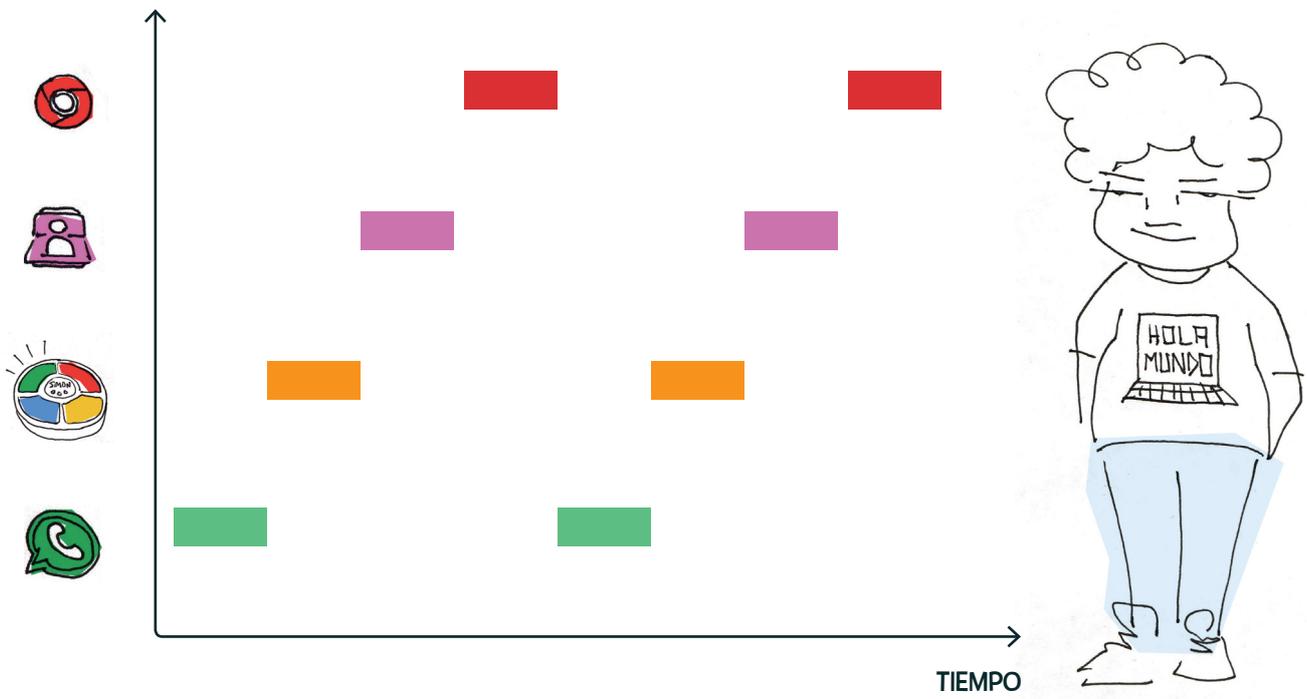
Proceso (programa en ejecución)

Programa (código fuente)

```
cuando Cámara .DespuésDeTomarFoto
imagen
ejecutar
poner LienzoSuperiorIzquierdo . ImagenDeFondo como tomar imagen
poner LienzoSuperiorDerecho . ImagenDeFondo como tomar imagen
poner LienzoInferiorIzquierdo . ImagenDeFondo como tomar imagen
poner LienzoInferiorDerecho . ImagenDeFondo como tomar imagen
```

¿MUCHOS PROCESOS AL MISMO TIEMPO?

Cuando corremos muchos programas al mismo tiempo, las instrucciones de cada uno de ellos se ejecutan intercaladamente, por turnos, de modo que cada proceso puede avanzar un poco en un período de tiempo relativamente corto. La muy alta velocidad a la que funcionan los procesadores es la que genera la ilusión de que el avance de la ejecución de los programas se produce en forma simultánea.



SCHEDULER

En los sistemas operativos, hay un programa llamado **scheduler**, que es el que se encarga de dividir el tiempo de uso del procesador y determinar qué proceso se ejecuta en cada turno. Hay muchas estrategias de *scheduling* distintas que priorizan diferentes variables: el tiempo de espera promedio de los procesos para disponer del procesador, la equidad del tiempo disponible del procesador para cada proceso, la ejecución de aquellos procesos que sean más críticos, etc.

¿Y SI TENEMOS MUCHOS NÚCLEOS?

Las computadoras actuales suelen tener más de un núcleo, por lo que sí pueden ejecutar distintas instrucciones simultáneamente. Sin embargo, cuando se ejecutan más programas que la cantidad de núcleos que tiene la computadora, no pueden ejecutarse todos al mismo tiempo. Por ejemplo, si ejecutamos 8 programas en un dispositivo que posee dos núcleos, resulta indispensable que se vayan turnando para que todos puedan avanzar. También en este caso es el **scheduler** el que se encarga de dividir el tiempo de uso de los núcleos e ir administrando los turnos entre los distintos procesos.



ADMINISTRACIÓN DEL ESPACIO



Esta es la vieja y querida *Bet & Rob compuimpresora*, comúnmente llamada *La 2x1*. Se trató de uno de los primeros modelos de computadora con impresora integrada. El lenguaje para programarla incluía solo dos instrucciones: una para escribir un carácter en una posición de la memoria, y otra para imprimir todo el contenido de un cierto rango de direcciones de memoria.

Lenguaje de la *Bet & Rob compuimpresora*

ESCRIBIR [c, n]: escribe el carácter c en la posición n de la memoria, donde c y n se reemplazan, respectivamente, por un carácter y un número cada vez que la instrucción es invocada.

IMPRIMIR [n_1, n_2]: imprime los caracteres que se encuentran entre las posiciones n_1 y n_2 de la memoria, donde n_1 y n_2 se reemplazan por un número cada vez que la instrucción es invocada.

A modo de ejemplo, mirá lo que sucede al ejecutar el programa que te mostramos a continuación.

MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["Ñ", 16]	16	
1	ESCRIBIR ["U", 17]	17	
2	IMPRIMIR [16, 17]	18	
3		19	
4		20	



MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["Ñ", 16]	16	Ñ
1	ESCRIBIR ["U", 17]	17	
2	IMPRIMIR [16, 17]	18	
3		19	
4		20	

MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["Ñ", 16]	16	Ñ
1	ESCRIBIR ["U", 17]	17	U
2	IMPRIMIR [16, 17]	18	
3		19	
4		20	



NOMBRE Y APELLIDO:

CURSO:

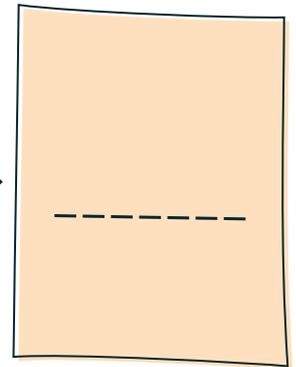
FECHA:

1. Acá hay dos programas para la compuimpresora: *Cotorra volá* y *Lara lara laringe*. Fijate qué hace cada uno y completá tanto el espacio de datos de la memoria al finalizar el programa como la palabra que imprime.

Cotorra volá

MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C", 16]	16	
1	ESCRIBIR ["O", 17]	17	
2	ESCRIBIR ["T", 18]	18	
3	ESCRIBIR ["O", 19]	19	
4	ESCRIBIR ["R", 20]	20	
5	ESCRIBIR ["R", 21]	21	
6	ESCRIBIR ["A", 22]	22	
7	IMPRIMIR [16, 22]	23	
8		24	

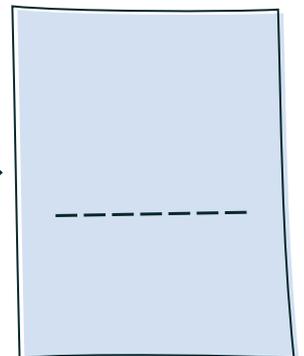
MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C", 16]	16	
1	ESCRIBIR ["O", 17]	17	
2	ESCRIBIR ["T", 18]	18	
3	ESCRIBIR ["O", 19]	19	
4	ESCRIBIR ["R", 20]	20	
5	ESCRIBIR ["R", 21]	21	
6	ESCRIBIR ["A", 22]	22	
7	IMPRIMIR [16, 22]	23	
8		24	



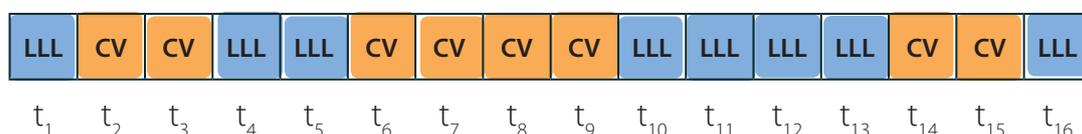
Lará lará laringe

MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["L", 16]	16	
1	ESCRIBIR ["A", 17]	17	
2	ESCRIBIR ["R", 18]	18	
3	ESCRIBIR ["I", 19]	19	
4	ESCRIBIR ["N", 20]	20	
5	ESCRIBIR ["G", 21]	21	
6	ESCRIBIR ["E", 22]	22	
7	IMPRIMIR [16, 22]	23	
8		24	

MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["L", 16]	16	
1	ESCRIBIR ["A", 17]	17	
2	ESCRIBIR ["R", 18]	18	
3	ESCRIBIR ["I", 19]	19	
4	ESCRIBIR ["N", 20]	20	
5	ESCRIBIR ["G", 21]	21	
6	ESCRIBIR ["E", 22]	22	
7	IMPRIMIR [16, 22]	23	
8		24	



2. ¡La *Bet & Rob compuimpresora* viene con un sistema operativo multitarea! Ahora están ambos programas en la memoria para ser ejecutados. El *scheduler* dividió el uso del procesador en turnos y los asignó a los procesos de la siguiente manera:¹



¹ LLL se refiere al programa *Lará lará laringe* y CV a *Cotorra volá*.

NOMBRE Y APELLIDO:

CURSO:

FECHA:

La primera instrucción que se ejecutará es de *Lará lará laringe*, luego dos de *Cotorra volá*, y así siguiendo. Completá la evolución de la ejecución siguiendo el orden definido por el scheduler. Para que veas cómo hacerlo, te mostramos a continuación el efecto de la ejecución de las tres primeras instrucciones.

MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C",16]	16	
1	ESCRIBIR ["O",17]	17	
2	ESCRIBIR ["T",18]	18	
3	ESCRIBIR ["O",19]	19	
4	ESCRIBIR ["R",20]	20	
5	ESCRIBIR ["R",21]	21	
6	ESCRIBIR ["A",22]	22	
7	IMPRIMIR [16,22]	23	
8	ESCRIBIR ["L",16]	24	
9	ESCRIBIR ["A",17]	25	
10	ESCRIBIR ["R",18]	26	
11	ESCRIBIR ["I",19]	27	
12	ESCRIBIR ["N",20]	28	
13	ESCRIBIR ["G",21]	29	
14	ESCRIBIR ["E",22]	30	
15	IMPRIMIR [16,22]	31	

MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C",16]	16	L
1	ESCRIBIR ["O",17]	17	
2	ESCRIBIR ["T",18]	18	
3	ESCRIBIR ["O",19]	19	
4	ESCRIBIR ["R",20]	20	
5	ESCRIBIR ["R",21]	21	
6	ESCRIBIR ["A",22]	22	
7	IMPRIMIR [16,22]	23	
8	ESCRIBIR ["L",16]	24	
9	ESCRIBIR ["A",17]	25	
10	ESCRIBIR ["R",18]	26	
11	ESCRIBIR ["I",19]	27	
12	ESCRIBIR ["N",20]	28	
13	ESCRIBIR ["G",21]	29	
14	ESCRIBIR ["E",22]	30	
15	IMPRIMIR [16,22]	31	

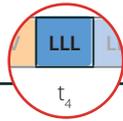
MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C",16]	16	C
1	ESCRIBIR ["O",17]	17	
2	ESCRIBIR ["T",18]	18	
3	ESCRIBIR ["O",19]	19	
4	ESCRIBIR ["R",20]	20	
5	ESCRIBIR ["R",21]	21	
6	ESCRIBIR ["A",22]	22	
7	IMPRIMIR [16,22]	23	
8	ESCRIBIR ["L",16]	24	
9	ESCRIBIR ["A",17]	25	
10	ESCRIBIR ["R",18]	26	
11	ESCRIBIR ["I",19]	27	
12	ESCRIBIR ["N",20]	28	
13	ESCRIBIR ["G",21]	29	
14	ESCRIBIR ["E",22]	30	
15	IMPRIMIR [16,22]	31	

MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C",16]	16	C
1	ESCRIBIR ["O",17]	17	O
2	ESCRIBIR ["T",18]	18	
3	ESCRIBIR ["O",19]	19	
4	ESCRIBIR ["R",20]	20	
5	ESCRIBIR ["R",21]	21	
6	ESCRIBIR ["A",22]	22	
7	IMPRIMIR [16,22]	23	
8	ESCRIBIR ["L",16]	24	
9	ESCRIBIR ["A",17]	25	
10	ESCRIBIR ["R",18]	26	
11	ESCRIBIR ["I",19]	27	
12	ESCRIBIR ["N",20]	28	
13	ESCRIBIR ["G",21]	29	
14	ESCRIBIR ["E",22]	30	
15	IMPRIMIR [16,22]	31	

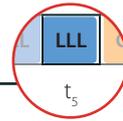
NOMBRE Y APELLIDO:

CURSO:

FECHA:

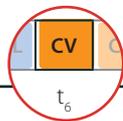


MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C", 16]	16	
1	ESCRIBIR ["O", 17]	17	
2	ESCRIBIR ["T", 18]	18	
3	ESCRIBIR ["O", 19]	19	
4	ESCRIBIR ["R", 20]	20	
5	ESCRIBIR ["R", 21]	21	
6	ESCRIBIR ["A", 22]	22	
7	IMPRIMIR [16, 22]	23	
8	ESCRIBIR ["L", 16]	24	
9	ESCRIBIR ["A", 17]	25	
10	ESCRIBIR ["R", 18]	26	
11	ESCRIBIR ["I", 19]	27	
12	ESCRIBIR ["N", 20]	28	
13	ESCRIBIR ["G", 21]	29	
14	ESCRIBIR ["E", 22]	30	
15	IMPRIMIR [16, 22]	31	

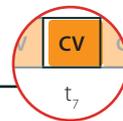


MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C", 16]	16	
1	ESCRIBIR ["O", 17]	17	
2	ESCRIBIR ["T", 18]	18	
3	ESCRIBIR ["O", 19]	19	
4	ESCRIBIR ["R", 20]	20	
5	ESCRIBIR ["R", 21]	21	
6	ESCRIBIR ["A", 22]	22	
7	IMPRIMIR [16, 22]	23	
8	ESCRIBIR ["L", 16]	24	
9	ESCRIBIR ["A", 17]	25	
10	ESCRIBIR ["R", 18]	26	
11	ESCRIBIR ["I", 19]	27	
12	ESCRIBIR ["N", 20]	28	
13	ESCRIBIR ["G", 21]	29	
14	ESCRIBIR ["E", 22]	30	
15	IMPRIMIR [16, 22]	31	

→



MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C", 16]	16	
1	ESCRIBIR ["O", 17]	17	
2	ESCRIBIR ["T", 18]	18	
3	ESCRIBIR ["O", 19]	19	
4	ESCRIBIR ["R", 20]	20	
5	ESCRIBIR ["R", 21]	21	
6	ESCRIBIR ["A", 22]	22	
7	IMPRIMIR [16, 22]	23	
8	ESCRIBIR ["L", 16]	24	
9	ESCRIBIR ["A", 17]	25	
10	ESCRIBIR ["R", 18]	26	
11	ESCRIBIR ["I", 19]	27	
12	ESCRIBIR ["N", 20]	28	
13	ESCRIBIR ["G", 21]	29	
14	ESCRIBIR ["E", 22]	30	
15	IMPRIMIR [16, 22]	31	



MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C", 16]	16	
1	ESCRIBIR ["O", 17]	17	
2	ESCRIBIR ["T", 18]	18	
3	ESCRIBIR ["O", 19]	19	
4	ESCRIBIR ["R", 20]	20	
5	ESCRIBIR ["R", 21]	21	
6	ESCRIBIR ["A", 22]	22	
7	IMPRIMIR [16, 22]	23	
8	ESCRIBIR ["L", 16]	24	
9	ESCRIBIR ["A", 17]	25	
10	ESCRIBIR ["R", 18]	26	
11	ESCRIBIR ["I", 19]	27	
12	ESCRIBIR ["N", 20]	28	
13	ESCRIBIR ["G", 21]	29	
14	ESCRIBIR ["E", 22]	30	
15	IMPRIMIR [16, 22]	31	

→

NOMBRE Y APELLIDO:

CURSO:

FECHA:

CV
 t_8

MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C",16]	16	
1	ESCRIBIR ["O",17]	17	
2	ESCRIBIR ["T",18]	18	
3	ESCRIBIR ["O",19]	19	
4	ESCRIBIR ["R",20]	20	
5	ESCRIBIR ["R",21]	21	
6	ESCRIBIR ["A",22]	22	
7	IMPRIMIR [16,22]	23	
8	ESCRIBIR ["L",16]	24	
9	ESCRIBIR ["A",17]	25	
10	ESCRIBIR ["R",18]	26	
11	ESCRIBIR ["I",19]	27	
12	ESCRIBIR ["N",20]	28	
13	ESCRIBIR ["G",21]	29	
14	ESCRIBIR ["E",22]	30	
15	IMPRIMIR [16,22]	31	

→

CV
 t_9

MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C",16]	16	
1	ESCRIBIR ["O",17]	17	
2	ESCRIBIR ["T",18]	18	
3	ESCRIBIR ["O",19]	19	
4	ESCRIBIR ["R",20]	20	
5	ESCRIBIR ["R",21]	21	
6	ESCRIBIR ["A",22]	22	
7	IMPRIMIR [16,22]	23	
8	ESCRIBIR ["L",16]	24	
9	ESCRIBIR ["A",17]	25	
10	ESCRIBIR ["R",18]	26	
11	ESCRIBIR ["I",19]	27	
12	ESCRIBIR ["N",20]	28	
13	ESCRIBIR ["G",21]	29	
14	ESCRIBIR ["E",22]	30	
15	IMPRIMIR [16,22]	31	

LLL
 t_{10}

MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C",16]	16	
1	ESCRIBIR ["O",17]	17	
2	ESCRIBIR ["T",18]	18	
3	ESCRIBIR ["O",19]	19	
4	ESCRIBIR ["R",20]	20	
5	ESCRIBIR ["R",21]	21	
6	ESCRIBIR ["A",22]	22	
7	IMPRIMIR [16,22]	23	
8	ESCRIBIR ["L",16]	24	
9	ESCRIBIR ["A",17]	25	
10	ESCRIBIR ["R",18]	26	
11	ESCRIBIR ["I",19]	27	
12	ESCRIBIR ["N",20]	28	
13	ESCRIBIR ["G",21]	29	
14	ESCRIBIR ["E",22]	30	
15	IMPRIMIR [16,22]	31	

→

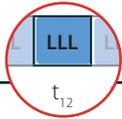
LLL
 t_{11}

MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C",16]	16	
1	ESCRIBIR ["O",17]	17	
2	ESCRIBIR ["T",18]	18	
3	ESCRIBIR ["O",19]	19	
4	ESCRIBIR ["R",20]	20	
5	ESCRIBIR ["R",21]	21	
6	ESCRIBIR ["A",22]	22	
7	IMPRIMIR [16,22]	23	
8	ESCRIBIR ["L",16]	24	
9	ESCRIBIR ["A",17]	25	
10	ESCRIBIR ["R",18]	26	
11	ESCRIBIR ["I",19]	27	
12	ESCRIBIR ["N",20]	28	
13	ESCRIBIR ["G",21]	29	
14	ESCRIBIR ["E",22]	30	
15	IMPRIMIR [16,22]	31	

NOMBRE Y APELLIDO:

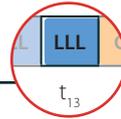
CURSO:

FECHA:

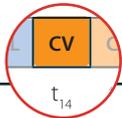


MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C",16]	16	
1	ESCRIBIR ["O",17]	17	
2	ESCRIBIR ["T",18]	18	
3	ESCRIBIR ["O",19]	19	
4	ESCRIBIR ["R",20]	20	
5	ESCRIBIR ["R",21]	21	
6	ESCRIBIR ["A",22]	22	
7	IMPRIMIR [16,22]	23	
8	ESCRIBIR ["L",16]	24	
9	ESCRIBIR ["A",17]	25	
10	ESCRIBIR ["R",18]	26	
11	ESCRIBIR ["I",19]	27	
12	ESCRIBIR ["N",20]	28	
13	ESCRIBIR ["G",21]	29	
14	ESCRIBIR ["E",22]	30	
15	IMPRIMIR [16,22]	31	

→

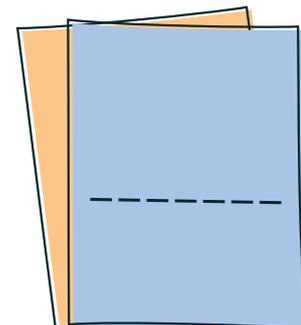
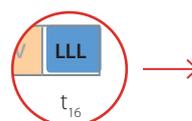
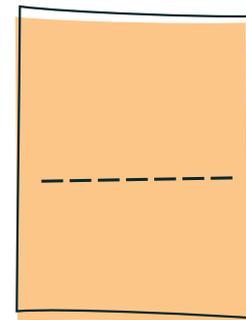
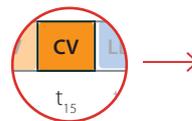


MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C",16]	16	
1	ESCRIBIR ["O",17]	17	
2	ESCRIBIR ["T",18]	18	
3	ESCRIBIR ["O",19]	19	
4	ESCRIBIR ["R",20]	20	
5	ESCRIBIR ["R",21]	21	
6	ESCRIBIR ["A",22]	22	
7	IMPRIMIR [16,22]	23	
8	ESCRIBIR ["L",16]	24	
9	ESCRIBIR ["A",17]	25	
10	ESCRIBIR ["R",18]	26	
11	ESCRIBIR ["I",19]	27	
12	ESCRIBIR ["N",20]	28	
13	ESCRIBIR ["G",21]	29	
14	ESCRIBIR ["E",22]	30	
15	IMPRIMIR [16,22]	31	



MEMORIA			
ESPACIO DE PROGRAMA		ESPACIO DE DATOS	
0	ESCRIBIR ["C",16]	16	
1	ESCRIBIR ["O",17]	17	
2	ESCRIBIR ["T",18]	18	
3	ESCRIBIR ["O",19]	19	
4	ESCRIBIR ["R",20]	20	
5	ESCRIBIR ["R",21]	21	
6	ESCRIBIR ["A",22]	22	
7	IMPRIMIR [16,22]	23	
8	ESCRIBIR ["L",16]	24	
9	ESCRIBIR ["A",17]	25	
10	ESCRIBIR ["R",18]	26	
11	ESCRIBIR ["I",19]	27	
12	ESCRIBIR ["N",20]	28	
13	ESCRIBIR ["G",21]	29	
14	ESCRIBIR ["E",22]	30	
15	IMPRIMIR [16,22]	31	

→



NOMBRE Y APELLIDO:

CURSO:

FECHA:

¿Qué imprimió la compuimpresora? ¿Por qué?

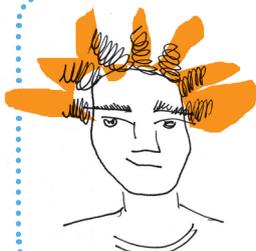
3. Describí una solución para que los distintos procesos no interfieran entre sí.



PARA NO PISARSE

Al correr varios programas a la vez, cada uno tiene que tener su propio espacio en la memoria. De este modo, no interferirá el uno con el otro.

MEMORIA			
COTORRA VOLÁ		LARÁ LARÁ LARINGE	
Programa	Datos	Programa	Datos



ESPACIO DE DIRECCIONES

La porción de la memoria que se le asigna a un proceso se conoce como **espacio de direcciones** y delimita el rango de direcciones que está disponible para el proceso que comienza a ejecutarse. De este modo, todas las direcciones de memoria que sean referenciadas desde el programa serán relativas a este espacio. Por ejemplo, la dirección 16 de un proceso no será la misma que la 16 de otro proceso, y así se evita la interferencia.

FOTOS E ILUSTRACIONES TÉCNICAS

USB-C: https://upload.wikimedia.org/wikipedia/commons/thumb/6/63/LeTV_X600_USB_Type_C_port.jpg/800px-LeTV_X600_USB_Type_C_port.jpg / Disco externo: Freepik.com / Disco SDD: https://upload.wikimedia.org/wikipedia/commons/thumb/5/5e/Vertex_2_Solid_State_Drive_by_OCZ-top_oblique_PNr%C2%B00307.jpg/1920px-Vertex_2_Solid_State_Drive_by_OCZ-top_oblique_PNr%C2%B00307.jpg / Placa de video: <https://upload.wikimedia.org/wikipedia/commons/thumb/6/62/AGP-Video-Card.jpg/1024px-AGP-Video-Card.jpg> / Freepik.com

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 secundaria / Claudia Banchoff Tzancoff ... [et al.] ; contribuciones de Vanessa Aybar Rosales ... [et al.] ; compilado por Silvina Justianovich ; coordinación general de Vanina Klinkovich ; Hernán Czemerinski ; editado por Ignacio David Miller ; Alejandro Palermo ; fotografías de Facundo Manini ; ilustrado por Jaqueline Schaab ; Juan Martín Serrovalle ; Klinko ; prólogo de María Belén Bonello ; Fernando Pablo Schapachnik. - 1a edición para el profesor - Ciudad Autónoma de Buenos Aires : Fundación Sadosky, 2019. Libro digital, PDF - (Ciencias de la Computación para el aula / Klinkovich, Vanina; Czemerinski, Hernán; 4)

Archivo Digital: descarga
ISBN 978-987-27416-8-6

1. Informática. 2. Programación. 3. Educación Secundaria. I. Banchoff Tzancoff, Claudia. II. Aybar Rosales, Vanessa, colab. III. Justianovich, Silvina, comp. IV. Klinkovich, Vanina, coord. V. Czemerinski, Hernán, coord. VI. Miller, Ignacio David, ed. VII. Palermo, Alejandro, ed. VIII. Manini, Facundo, fot. IX. Schaab, Jaqueline, ilus. X. Serrovalle, Juan Martín, ilus. XI. Klinko, ilus. XII. Bonello, María Belén, prolog. XIII. Schapachnik, Fernando Pablo, prolog.
CDD 004.0712

Queda hecho el depósito que dispone la Ley 11.723
Ediciones Colihue.

 Primera edición: Diciembre de 2019.

El contenido del manual se distribuye bajo la licencia Creative Commons Compartir Igual.