

¿Qué es programar?

Autómatas, programas y primitivas

¿Cómo definiríamos qué es programar? ¿Qué tipo de instrucciones podemos darle a una computadora para que resuelva un problema? ¿Qué posibilidades y limitaciones tenemos cuando expresamos soluciones a problemas computacionales? ¿Qué es y cómo se usa Pilas Bloques?

En esta secuencia nos aproximamos a la elaboración de soluciones a problemas con instrucciones simples y precisas para comprender la existencia de restricciones al construir programas para computadoras. Trabajaremos en modalidad "desenchufada" con dramatizaciones y con ejercicios en papel para luego, presentar el entorno de programación Pilas Bloques.

Actividad 1

Las y los estudiantes le dan indicaciones a su docente para que alcance un objetivo como si se tratara de una computadora. De esta manera, comienzan a trabajar con estrategias de solución y se introducen nociones fundamentales de programación, como **problema computacional, autómata, programa, secuencia y primitiva**.

Actividad 2

Las y los estudiantes diseñan un problema para ser resuelto por un autómata, lo que les requiere poner en juego las nociones fundamentales de programación que surgieron en la actividad anterior.

Actividad 3

A partir del desafío en Pilas Bloques **Capó y Guyrá**, las y los estudiantes exploran el entorno del programa, las nociones presentadas en las actividades anteriores, y se familiarizarán con la metodología de trabajo.

Datos curriculares

Nivel: Primaria, segundo ciclo; Secundaria, ciclo básico

Área: Programación

Eje: Soluciones a problemas computacionales

- Diseño de programas: estrategia.

Eje: Lenguajes de programación

- Herramientas de lenguajes de programación: comandos primitivos, secuencias.
- Ejecución secuencial de programas: ejecución, autómata.

Objetivos de aprendizaje

- Conocer el nivel de precisión y especificidad de los comandos primitivos empleados en programación y su relación con cómo deben formularse las soluciones computacionales.
- Aproximarse a las nociones de autómata, programa y primitiva o comando primitivo.
- Conocer el entorno de enseñanza de la programación Pilas Bloques y su metodología de trabajo.

Materiales necesarios

- Pizarra o pizarrón.
- Hojas de papel y útiles para escribir.
- Dispositivos con Pilas Bloques instalado o acceso a su versión online <https://pilasbloques.program.ar/>

Acerca de esta iniciativa

Desde el sitio curriculum.program.ar tenemos por objetivo acompañar a la comunidad docente de habla hispana en el desafío de llevar las Ciencias de la Computación al aula. Para ello, construimos un repositorio que reúne diversos recursos para el aula que desde la Iniciativa Program.AR de la Fundación Sadosky impulsamos desde 2013.

Organizados a partir de los saberes a promover con nuestras y nuestros estudiantes y los conceptos de la disciplina presentados en la [Propuesta curricular para la inclusión de las Ciencias de la Computación \(CC\) en el aula](#), encontrarán en curriculum.program.ar proyectos, secuencias didácticas y actividades desarrollados por una diversidad de autores y docentes en conjunto con instituciones y universidades de América Latina.

Estos materiales, que han sido desarrollados para responder a necesidades de diferentes contextos y países y que son heterogéneos en su formato y extensión, comparten un mismo propósito: integrar las Ciencias de la Computación en la escolaridad obligatoria para promover en el conjunto de las y los estudiantes la construcción de saberes que les permitan comprender, apropiarse y transformar la tecnología digital y computacional y así participar de manera crítica del mundo contemporáneo.

Perspectiva de género

La Fundación Sadosky busca propiciar una experiencia educativa inclusiva y promotora de la equidad de género. Sabemos que existe una fuerte desigualdad de género en el acceso al uso de recursos tecnológicos y a conocimientos de ciencias de la computación. Uno de los motivos de esta brecha tiene que ver con que socialmente es considerada como una disciplina de varones. Por eso es imprescindible que, como docentes, podamos contribuir a desnaturalizar prejuicios y generar estrategias para incentivar especialmente el trabajo de estudiantes mujeres y de identidades de género trans y no binarias.

En el documento [Enseñar computación desde la mirada de la Educación Sexual Integral \(ESI\)](#) es posible encontrar orientaciones para crear aulas más inclusivas y respetuosas para estudiantes y docentes de todos los niveles educativos.

Cómo utilizar este recurso

Esta secuencia es parte de una colección que se encuentra disponible en el sitio curriculum.program.ar

Se integran actividades “desenchufadas” o en papel, con otras en plataformas especialmente diseñadas para la enseñanza de la programación, como Pilas Bloques o Scratch.

Créditos

La presente propuesta es una adaptación de: Factorovich, P. y Sawady O'Connor, F. (2017), *Actividades para aprender a Program.AR: segundo ciclo de la educación primaria y primero de la secundaria*, segunda edición.

Autores: Javier Castrillo y Fernando Cáceres.

Coordinación autoral: Julián Dabbah

Coordinación editorial: Inés Roggi

Edición: Florencia N. Acher Lanzillotta

Diseño: Fabio Viale

Cómo citar este documento

Fundación Sadosky (2024), “¿Qué es programar? Autómatas, programas y primitivas”, en *Actividades para aprender a Program.AR*. Disponible en: <https://curriculum.program.ar/>



Listado de secuencias que componen esta colección

Primitivas, procedimientos y repetición

1. ¿Qué es programar?
2. Definimos nuestros bloques
3. Programamos en papel cuadriculado
4. Programamos estrategias en Pilas Bloques
5. Creamos desafíos de repetición
6. Seguimos programando estrategias en Pilas Bloques
7. Creamos desafíos de procedimientos

Alternativa condicional

8. ¿Cómo se resuelven problemas cambiantes?
9. Resolvemos recorridos cambiantes
10. Programamos estrategias para problemas cambiantes
11. Creamos desafíos cambiantes

Interactividad y variables

12. ¿Podemos programar otros personajes?
13. Programamos el personaje de un videojuego
14. Guardamos información
15. Programamos nuestro videojuego

Repetición condicional

16. Un videojuego que no sabemos cuándo termina

Actividad 1

Todo al pie de la letra

Las y los estudiantes deben “programar” a su docente: le darán instrucciones para que realice una tarea como si fuera un robot o una computadora. De esta manera, se encontrarán con el desafío que implica formular una tarea con este tipo de instrucciones. Se enmarcará este trabajo con algunas nociones teóricas fundamentales.

Objetivos >

Se espera que las y los estudiantes:

- Expresen tareas en términos de expresiones acotadas, simples y formuladas de manera muy precisa.
- Se aproximen a las nociones de comando primitivo o primitiva, autómatas y programas.



Inicio >

El **propósito de este momento** es introducir la tarea de programación a partir de las concepciones previas de las y los estudiantes sobre el tema.

Orientaciones



¿Qué programas de computadora conocen? ¿Para qué sirven? ¿Qué características tiene un programa? ¿Cómo y quiénes los crean? ¿Les parece que es una tarea fácil o difícil?

Buscamos relevar los saberes previos y los conocimientos intuitivos de las y los estudiantes sobre los programas de computadora y su creación. Esto nos dará herramientas para contextualizar esta secuencia como un primer paso en el aprendizaje de la programación. Sus respuestas también podrán ser insumo para contrastarlas con las reflexiones que surjan en el momento de cierre.

Desarrollo >

El **propósito de este momento** es introducir la noción de programa como una secuencia de instrucciones simples, formuladas de manera muy precisa y sin ambigüedades, que forman parte de un repertorio acotado (las primitivas) que una máquina (el autómatas) puede llevar a cabo (ejecutar) automáticamente.

Orientaciones

Planteamos a las y los estudiantes que los docentes actuemos como **autómatas**, es decir, como una máquina que (como las computadoras) sigue al pie de la letra las instrucciones que se le dan. Proponemos cumplir algún propósito o recorrido en el aula (por ejemplo, llegar a la puerta desde un rincón del aula y salir). El objetivo de las y los estudiantes es dar las instrucciones para cumplir el objetivo. Sin más explicaciones, comenzamos a escuchar instrucciones e iremos construyendo el programa en el pizarrón. Anotamos las instrucciones una debajo de la otra, aprovechando los errores que informa el autómata para descubrir de qué manera formular estas instrucciones.

A medida que las y los estudiantes realizan sus propuestas, pueden suceder las siguientes situaciones:

- **La instrucción podría ser incorrecta**, es decir que tal como está formulada, no forma parte del repertorio de instrucciones factibles o conocidas por el autómata. Esto puede deberse a la **complejidad** de la instrucción (podemos considerar que “ir hasta la puerta” o “avanzar 10 pasos” son instrucciones complejas mientras que “avanzar un paso” o “dar un paso adelante” son instrucciones lo suficientemente simples), a que la instrucción sea **ambigua** (si nos dicen “mover” podemos interpretarlo como moverse en el lugar o avanzar en cualquier dirección) o a que la instrucción esté **expresada de una manera sustancialmente diferente a la esperada** (por ejemplo, si esperamos “avanzar un paso” entonces “mover adelante” o “caminar” serán consideradas incorrectas). En estos casos, podemos responder “No sé hacer eso” para exponer las dificultades que implican resolver una tarea sólo con instrucciones simples y formuladas de manera muy precisa¹.
- **La instrucción puede ser correcta, pero no puede ser ejecutada** (por ejemplo, “abrir la puerta” si estamos lejos de la puerta o esta ya está abierta). En ese caso, indicamos que no podemos ejecutar la instrucción.
- **La instrucción sea correcta y pueda ser realizada, pero nos aleje de resolver el problema** (por ejemplo, caminar en el sentido contrario o avanzar sobre un banco o contra una pared). En ese caso, la ejecutamos, es decir, obedecemos la orden propuesta para mostrar que se aleja de lo buscado. Es probable que las y los estudiantes identifiquen esta situación y quieran corregirnos sobre la marcha o apelen a nues-

¹ Si surgieran comandos más avanzados, como la repetición o la alternativa condicional, los consideraremos como demasiado complejos dado que se trata de una actividad introductoria.

tro sentido común para que interrumpamos la ejecución. Esta es una oportunidad para reforzar que, como autómatas, no somos capaces de evaluar si las instrucciones nos ayudan a resolver el problema o no, solo ejecutamos las instrucciones recibidas en el orden en el que nos fueron dadas.

- **La instrucción puede estar correctamente formulada, puede ser ejecutada y nos permita resolver el problema.** En ese caso, seguimos la instrucción (la ejecutamos) y la registramos en el pizarrón con la precisión con la que debe ser formulada.

Cierre >

El **propósito de este momento** es conceptualizar la noción de *programa*, *comando primitivo* o *primitiva* y *autómata* y que las y los estudiantes reconozcan que el orden en el que se indican las primitivas en un programa es importante para que se cumpla el objetivo.

Orientaciones

Recuperamos con las y los estudiantes la experiencia del primer ejercicio para construir algunas definiciones presentes en cualquier problema de programación. Se puede recurrir a las concepciones previas por las y los estudiantes en el momento de inicio y hacer un refuerzo o contrastación con lo que comenten en el intercambio de esta instancia.

Es importante arribar a las siguientes nociones.

- Un **programa** es una descripción de una **solución** posible a un problema específico que se busca resolver con un **autómata** de forma automática, expresado como una **secuencia de instrucciones**. En esta secuencia, un ejemplo de programa sería lo que quedó escrito en el pizarrón al terminar el primer ejercicio.
- Las **instrucciones** con las que se construyen los programas deben ser lo suficientemente simples y estar formuladas de manera específica para que el autómata pueda reconocerlas y realizarlas, como señalamos cada vez que propusieron una instrucción que no sabíamos ejecutar. A estas instrucciones les diremos **comandos primitivos** o **primitivas**. Por ejemplo, una primitiva podría ser "avanzar un paso".
- Al proceso por el cual un autómata interpreta las instrucciones escritas en un programa y las lleva a cabo, le llamamos **ejecución** del programa.



¿Qué forma tiene el programa? ¿Se podrían intercambiar instrucciones de lugar y que el programa siga funcionando? ¿Cómo separarían el programa en partes? ¿Por qué?

Analizamos el programa escrito en el pizarrón para reflexionar sobre dos aspectos.

- Al observar la **forma secuencial del programa**, reflexionamos sobre el **orden** en el que aparecen las instrucciones en la secuencia: en algunos casos, será importante para que se resuelva el problema y en otros no. Podemos promover que las y los estudiantes especulen sobre cómo funcionaría el programa con las instrucciones en otro orden. Podemos señalar ejemplos en los que el orden sea importante, tales como en las situaciones en las que, para avanzar, primero se debe esquivar un obstáculo avanzando hacia un costado y luego avanzando por el camino libre; para el caso contrario, podemos identificar recorridos en terreno libre que involucren más de una dirección (por ejemplo, avanzar hacia el costado y hacia adelante sin obstáculos).
- Al observar el **programa completo**, podemos señalar que, si bien la solución está formulada como una única secuencia de instrucciones muy simples, podemos identificar **partes dentro de esta secuencia donde cada una resuelve una situación particular**. Estas situaciones son más complejas que las que puede resolver cada primitiva por separado pero más simple que el problema original. Estos son **sub-problemas**. Para ponerlo de manifiesto, proponemos que busquen estas partes en el programa y especifiquen qué resuelve cada una. Luego, leyendo la solución desde este enfoque, vemos que existe una **estrategia** para resolver el problema, que refleja la manera en la que elegimos resolverlo (por ejemplo, eligiendo un camino para llegar a la puerta) pero que no suele ser única (probablemente existan otros caminos que podríamos haber tomado). Resaltamos estas ideas para continuar trabajando con ellas en futuras actividades.

Girar 90° a la derecha	} Estrategia de solución
Avanzar	
Avanzar	
Avanzar	
Girar 90° a la izquierda	
Avanzar	
Avanzar	} Ubicarse frente a la puerta
Girar 90° a la izquierda	
Extender el brazo	} Abrir la puerta
Empujar la manija	
Avanzar	
	} Salir

Un ejemplo de programa para salir del aula, en el que se utilizan primitivas de avance y de giros.

Actividad 2

Seamos autómatas

Profundizamos las ideas trabajadas en la actividad anterior cambiando los roles: serán las y los estudiantes quienes formulen los problemas para que sus compañeros y compañeras los resuelvan como si fueran autómatas.

Objetivos >

Se espera que las y los estudiantes:

- Demuestren comprensión de la noción de primitivas.
- Se aproximen a la noción de **problema computacional**.



Inicio >

El **propósito de este momento** es recuperar la noción de programa como solución a un problema y evidenciar los límites de los problemas que pueden ser resueltos por un autómata determinado.

Orientaciones

Dividimos la clase en grupos de dos o tres estudiantes. Podemos utilizar una dinámica lúdica para generar grupos al azar, con el objetivo de formar grupos heterogéneos.

Comenzamos planteando a las y los estudiantes que piensen un problema para ser resuelto por un autómata. Adelantamos que se lo entregarán a otro grupo para que lo resuelva con instrucciones paso a paso, del mismo modo en que lo hicimos en la **Actividad 1**, y, por eso, deberán describir claramente el problema e indicar todo lo que consideren necesario para que el otro grupo pueda construir una solución para que el autómata resuelva el problema.



¿Qué deberían tener en cuenta para definir un problema a ser resuelto por un autómata? ¿Qué información deberán pasarle al otro grupo para que pueda construir una solución al problema?

Estas preguntas pretenden que las y los estudiantes recuperen las nociones presentadas en la actividad anterior, con el objetivo de puntualizar la consigna de la actividad. Considerando esto, se puede orientar el intercambio hacia, por ejemplo, consensuar qué acciones podrán realizar los autómatas. Es importante hacer hincapié en que el relevamiento completo

de un problema requiere informar las primitivas, las condiciones iniciales y de contexto en las que se enmarca.

Podemos tomar nota en el pizarrón de las cuestiones que se acuerdan considerar al escribir el problema y la solución. De esta forma, podemos evidenciar que se trata de decisiones arbitrarias que se acuerdan para establecer un diálogo con el autómata.

Desarrollo >

El **propósito de este momento** es reforzar la comprensión de la noción de primitiva (una instrucción simple de un repertorio arbitrario y acotado, asociadas a un autómata y un problema en particular en un determinado contexto), y, sobre todo, las particularidades que conlleva formular programas a partir de ellas.

Orientaciones

En una primera parte, los grupos definen el problema que quieren que el otro grupo resuelva. Acompañamos el trabajo de los grupos para que logren una versión escrita del problema a resolver, prestando atención a que:

- incluya el autómata y las primitivas disponibles.
- pueda resolverse únicamente con las primitivas planteadas y con una solución no demasiado extensa.

Para que puedan verificar esto, podemos proponer que se coloquen en el lugar de quién recibirá el problema y ensayen una solución modelo.

En una segunda parte, los grupos intercambian sus problemas para resolverlos. Atendemos al intercambio entre los grupos por si no se comprende qué hace una primitiva o alguna pauta del problema a resolver. Asimismo, podemos pedirles que registren las dificultades que encuentren durante la elaboración y la resolución del problema para discutir las en el momento de cierre, especialmente, si las primitivas no eran suficientes, no era claro su funcionamiento o eran muy complicadas de ejecutar.

Para desarrollar la solución, deben escribir claramente las instrucciones en orden secuencial y cumplir con los acuerdos que anotamos en el pizarrón o pizarra en el inicio de esta actividad.

Para probar o presentar la solución, pueden actuar físicamente en el aula (como hicimos en la actividad anterior) o simularla en un esquema en papel.

Cierre >

El **propósito de este momento** es identificar cómo las particularidades del trabajo con primitivas impactan en las maneras de pensar la solución y en el tipo de problemas que podemos resolver, y así comenzar a construir la noción de *problema computacional*.

Orientaciones



¿Qué tan fácil o difícil fue comprender cuál era el problema planteado por el otro grupo? ¿Pudieron resolver el problema con las primitivas disponibles? ¿Por qué? ¿Había alguna primitiva que fuera muy general o que se pudiera haber separado en acciones más chicas? ¿Hubo algún aspecto del problema que no pudiera resolverse?

Recuperar las dificultades encontradas en la solución de los problemas planteados es una oportunidad para revisar y reforzar las nociones abordadas en la actividad anterior (programa, autómatas y primitivas) y sus particularidades (el autómata puede ejecutar un repertorio acotado de primitivas; y estas tienen que estar formuladas de manera muy precisa, porque el autómata ejecuta las instrucciones "sin pensar" ni tomar decisiones sobre ellas).

También nos interesa señalar que, así como las especificidades del trabajo con autómatas y primitivas impacta en la forma de las soluciones que construimos, también influye en el tipo de problemas que podemos resolver, como vieron cuando tuvieron que pensar el problema para las y los compañeros. En este sentido, a los problemas que podemos resolver con un autómata (en particular, con una computadora), les diremos **problemas computacionales**. No nos interesa ensayar una definición cerrada pero sí identificar que al resolver problemas con autómatas y primitivas hay restricciones y tenemos que conocerlas al construir soluciones (que posiblemente sean diferentes a las que conocemos de la vida cotidiana).

Material de referencia para docentes

¿Por qué decidimos no hablar de algoritmos (todavía)?

En el ámbito de las Ciencias de la Computación, un **algoritmo** es una descripción general de un único proceso que permite construir una solución para todas las instancias de un problema, por ejemplo, la descripción de un método que permita transformar una lista cualquiera de números para que quede ordenada de menor a mayor, en el que podemos identificar claramente el modelo de la computación como “entrada, transformación y salida de información”. Asociar a los algoritmos con una secuencia de pasos para resolver tareas cotidianas esconde con su simplicidad aspectos que son inherentes a la noción disciplinar, construyendo una definición errónea de un término clave para la disciplina y poco relevante para las y los estudiantes (dado que, en este nivel, solo se usaría para nombrar un orden de pasos). Abordar esta noción central de la Computación requiere hacerlo en una instancia posterior de la formación de las y los estudiantes en la que puedan razonar en general y en abstracto.

Arribamos a esta decisión a partir de discusiones, experiencias e investigaciones en relación con el término y las dificultades que presentó su uso simplificado o banalizado.

Más información al respecto en: Dabbah, J., *et al. Propuesta curricular para la inclusión de las Ciencias de la Computación en la educación obligatoria de Argentina*. Fundación Sadosky, 2024. Disponible en línea. https://curriculum.program.ar/wp-content/uploads/2023/01/Program.ar_Propuesta-Curricular-para-la-inclusion-de-las-Ciencias-de-la-Computacion.pdf

Actividad 3

Capý y Guyrá.

Conocemos Pilas Bloques

Las y los estudiantes abordan el desafío **Capý y Guyrá**, que motiva a explorar el entorno de programación Pilas Bloques aplicando las nociones fundamentales de programación presentadas en las actividades anteriores.

Objetivos >

Se espera que las y los estudiantes:

- Conozcan el entorno de programación Pilas Bloques y su dinámica de trabajo.



Inicio >

El **propósito de este momento** es motivar dinámicas de trabajo que habiliten a las y los estudiantes a explorar Pilas Bloques, prestando atención y reforzando que no haya desigualdades en el uso y el acceso a los dispositivos y en el tipo de tareas que realice cada integrante del grupo.

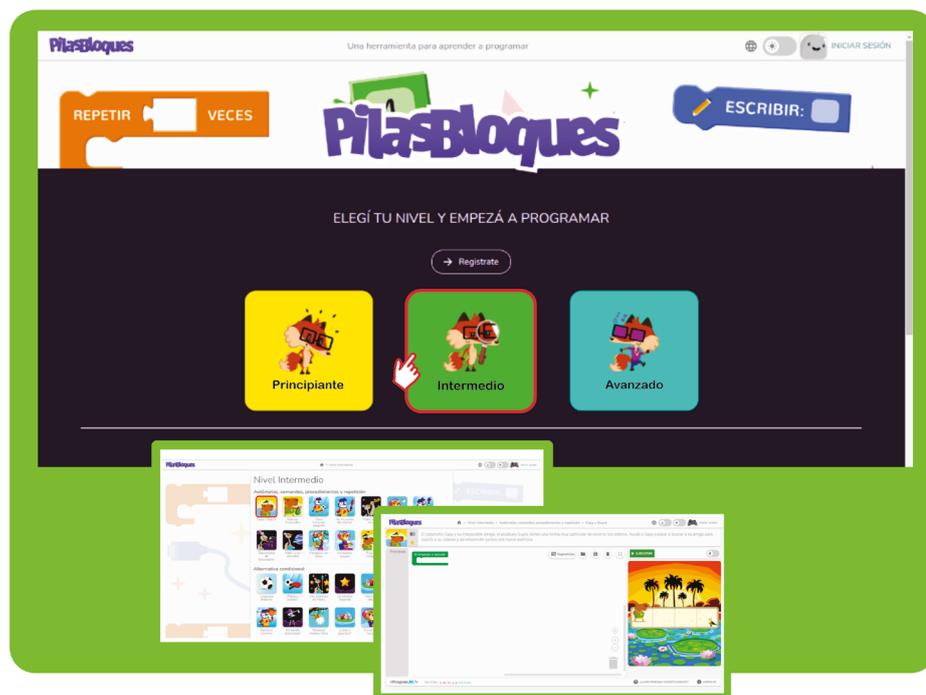
Orientaciones

Esta es una nueva oportunidad para relevar la experiencia de las y los estudiantes en el uso de computadoras, tecnologías digitales, videojuegos y programación. Esta información nos será muy útil para identificar quiénes cuentan con más y con menos experiencia, y procurar aprovechar esa diversidad para beneficio de todos favoreciendo el intercambio en grupos heterogéneos.²

Se propone que las y los estudiantes formen pequeños grupos heterogéneos, idealmente de a dos, e ingresen al desafío **Capý y Guyrá** en Pilas Bloques, del nivel intermedio.

Dado que es probable de que sea la primera vez que acceden a los desafíos de Pilas Bloques es importante acompañar el acceso al desafío propuesto.

² Para ello, una actividad sencilla, es indicar que levanten la mano, golpeen la mesa o den un aplauso quienes *“tengan computadora en su casa”, “usen internet todos los días”, “jueguen videojuegos”, “alguna vez programaron”,* etc.



Es conveniente explicitar la dinámica de trabajo en grupos, prestando atención a explorar el entorno en conjunto, discutir y consensuar las decisiones que tomarán para resolver el desafío y acceder al dispositivo para asegurar que todas y todos puedan trabajar en Pilas Bloques de forma equitativa en cuanto a tiempo de uso y tipo de tareas.

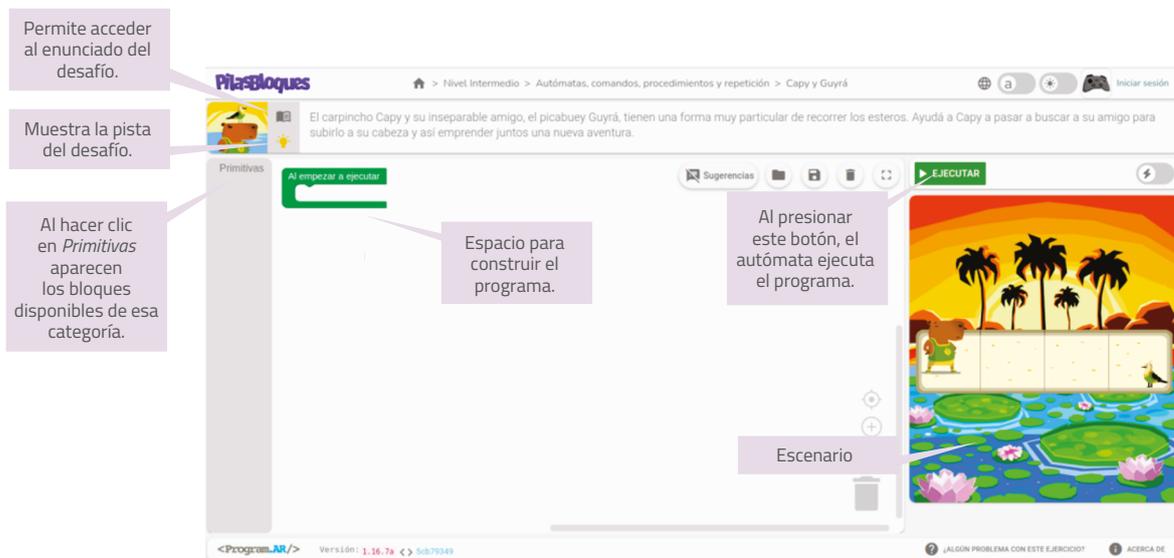
Desarrollo >

El **propósito de este momento** es que las y los estudiantes exploren el entorno de programación Pilas Bloques para comenzar a construir la dinámica de trabajo en la plataforma.

Orientaciones

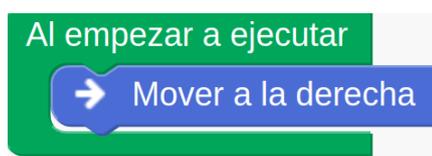
La dinámica de este momento está fuertemente orientada hacia la **exploración de la herramienta por parte de las y los estudiantes**. Como docentes, nuestro rol será alentar esta exploración y resolver inquietudes para sostener este proceso evitando frustraciones.

Mientras que las y los estudiantes resuelven el desafío, y en la medida que sea necesario, se puede orientar la exploración mediante la señalización de elementos presentes en el desafío que representan alguna de las nociones fundamentales de programación que se vienen trabajando desde el inicio de la secuencia. En este sentido, y después del proceso de exploración o incluso después de que algunos grupos hayan resuelto el problema, se propone identificar: la consigna y la pista del desafío, el escenario o contexto (y sus elementos), el espacio para armar el programa y el menú de bloques que se despliega a partir de la categoría *Primitivas*.

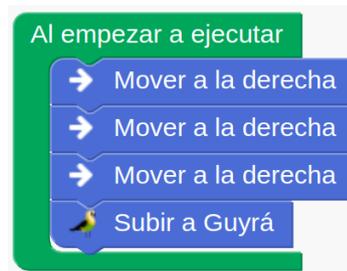


En cuanto al manejo de los dispositivos, es probable que haya quienes necesiten ayuda adicional (de nuestra parte o de sus compañeros y compañeras) en función de su experiencia previa, tal como podemos haberlo identificado al comienzo de la clase. Es importante reforzar el acompañamiento en estos casos para no profundizar desigualdades en el acceso y uso a la tecnología.

Considerando que este es el primer desafío en Pilas Bloques, es probable que requieran asistencia en la elaboración del programa, en particular, en determinar que el programa se construye arrastrando y encastrando bloques según sus formas. En este sentido, se les puede orientar a observar que el bloque **Al empezar a ejecutar** tiene un pico hacia abajo y las primitivas **Mover a la derecha** y **Subir a Guyrá** tienen esa misma forma, que encastraría como si fuera un rompecabezas.



Invitamos a aquellas y aquellos estudiantes que hayan completado el programa, pero no lo hayan ejecutado aún, a que lo hagan para asegurarse de que funciona correctamente. Recordamos que el autómata resuelve el problema **ejecutando** el programa y, para tal fin, tienen disponible el botón **▶ EJECUTAR**. Les indicamos que lo presionen y observen cómo, a medida que se ejecuta el programa, se resalta el bloque que está siendo ejecutado.



Un programa que resuelve el desafío.

Los personajes, en su rol de autómatas, informarán si ocurren errores durante la ejecución del programa (por ejemplo, ejecutar la primitiva **Subir a Guyrá** fuera del casillero en el que se encuentra o intentar desplazar el personaje fuera del tablero).



Ejemplos de errores durante la ejecución del programa.

Si no ocurrieron errores y se cumplió el objetivo del desafío, aparecerá un cartel que indica que pudieron resolver el desafío.



Cierre >

El **propósito de este momento** es comenzar a ensayar una dinámica para el trabajo con Pilas Bloques, relacionándola con las nociones de programación trabajadas en la secuencia.

Orientaciones



¿Cómo se organizaron para explorar el entorno de Pilas Bloques? ¿Lo hicieron en conjunto? ¿Por turnos? ¿Se repartieron las tareas? ¿Pudieron ponerse de acuerdo para decidir las instrucciones para crear el programa? ¿Qué dificultades encontraron en el trabajo en grupo? ¿Cómo las resolvieron?

En este cierre, aprovechamos para retomar los acuerdos que hicimos en el inicio de la actividad sobre la dinámica de trabajo en grupos³.



¿Cómo supieron qué había que hacer, es decir, cuál era el problema a resolver? ¿Cómo supieron qué primitivas podía ejecutar el autómata? ¿Dónde las encontraron? ¿Cómo supieron si el programa funcionaba? ¿Cómo se ejecuta un programa en Pilas Bloques?

En este intercambio nos interesa recuperar de la experiencia:

- cuestiones **instrumentales** fundamentales para trabajar en **Pilas Bloques**, como la ubicación de cada uno de los elementos de un desafío como la consigna, la pista y el escenario, las primitivas y la forma de crear un programa (o solución) y cómo ejecutarlo. Podemos hacer una revisión del entorno para ir identificando dónde aparecen y cómo accedemos a cada uno de ellos.
- la idea de que **en el trabajo en computadora están presentes las mismas nociones** que vimos en las actividades anteriores de la secuencia, para asociar a los personajes con el autómata y a los bloques azules con las primitivas.

³ Podemos aprovechar la ocasión para indagar qué sucedió en torno al uso compartido de la computadora y la división de tareas. Podemos promover la reflexión sobre si se manifestaron ideas estereotipadas acerca de las habilidades asociadas al género u otros prejuicios o preconceptos y movilizar cambios al respecto acompañando al grupo.