SECUENCIA DIDÁCTICA

Programamos en papel cuadriculado

Legibilidad y reutilización

¿Podemos dividir un problema complejo en partes y resolverlas por separado? ¿Cómo hacemos para que otras personas entiendan más fácil nuestros programas? ¿Podemos aprovechar soluciones creadas por otras personas para crear un nuevo programa?

En esta secuencia, a partir de un lenguaje simbólico con comandos para pintar casilleros en una grilla, las y los estudiantes se enfocan en las nociones de legibilidad y reutilización. También se recuperan las nociones de estrategia de solución, procedimientos y repetición simple.

Actividad 1

Las y los estudiantes escriben programas para hacer dibujos sencillos con el objetivo de motivar el uso de procedimientos y repetición simple.

Actividad 2

Las y los estudiantes escriben programas para un dibujo con partes diferenciadas, para recuperar la noción de estrategia y reflexionar acerca de la importancia de la legibilidad, es decir, de crear programas pensando en la facilidad de comunicación y comprensión del programa.

Actividad 3

Las y los estudiantes escriben programas para realizar una serie de dibujos similares entre sí, para introducir la noción de reutilización, asociada a la definición de procedimientos. Además, reutilizan los programas de sus compañeras y compañeros para poner de manifiesto cómo la reutilización y la legibilidad permiten la construcción colectiva de programas.

Actividad 4

A modo de actividad integradora, las y los estudiantes proponen un dibujo para que otro equipo escriba un programa que lo realice. Además de proveer una nueva instancia para aplicar las nociones abordadas, la elaboración de un desafío de programación lleva a reflexiones más profundas entre formas del dibujo y las herramientas de programación necesarias para conseguirlas. También, reforzamos la importancia de considerar cuestiones de legibilidad a la hora de compartir soluciones.

Datos curriculares

Nivel: Primaria, segundo ciclo; Secundaria, ciclo básico

Área: Programación

Eje: Soluciones a problemas computacionales

- Diseño de programas: estrategia, legibilidad.

Eje: Lenguajes de programación

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

Objetivos de aprendizaje

- Proponer soluciones que prioricen la legibilidad de los programas mediante la división en subproblemas y la definición y utilización de procedimientos y repeticiones.
- Incorporar la importancia de diseñar soluciones considerando la reutilización.

Saberes previos de CC

Área: Programación

Eje: Lenguajes de programación

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

Eje: Soluciones a problemas computacionales

- Diseño de programas: estrategia, legibilidad.

Materiales necesarios

- Hojas de papel cuadriculado.
- Lápices o lapiceras.

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: Fernando Cáceres y Javier Castrillo.

Coordinación autoral: Julián Dabbah

Coordinación editorial: Inés Roggi

Edición: Florencia N. Acher Lanzillotta

Diseño: Fabio Viale

Cómo citar este documento

Fundación Sadosky (2024), "Programamos en papel cuadriculado. Legibilidad y reutilización", en *Actividades para aprender a Program.AR*. Disponible en: https://curriculum.program.ar/

Listado de secuencias que componen esta colección

Actividad 1
Repeticiones y procedimientos

En esta actividad presentamos el lenguaje simbólico que las y los estudiantes utilizarán en la secuencia. Este lenguaje permite elaborar programas para dibujar en una superficie cuadriculada, y permite que hagamos énfasis en el uso de la repetición y la definición de procedimientos.

Objetivos >

Se espera que las y los estudiantes:

- Utilicen la repetición y los procedimientos para construir programas más fácilmente interpretables y modificables.

Inicio >

El **propósito de este momento** es introducir el lenguaje simbólico que utilizaremos en la secuencia y generar asociaciones con las nociones de autómata, primitiva y ejecución.

Orientaciones

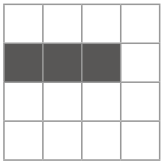
Presentamos, a partir de un ejemplo, un lenguaje para dibujar pintando casillas en una hoja cuadriculada, que emula los lenguajes de programación. Mostramos cuáles son las primitivas, cada una representada por un símbolo, y explicamos que permiten moverse un casillero para cada lado, o bien, pintar el casillero en el que se está situado.

Primitivas	
←	Mover el lápiz al casillero de la izquierda
→	Mover el lápiz al casillero de la derecha
↑	Mover el lápiz al casillero de arriba
↓	Mover el lápiz al casillero de abajo
■	Pintar el casillero

Primitivas del lenguaje, con su símbolo y su significado.

Hacemos una breve secuencia de comprobación para esclarecer la dinámica de trabajo con estos programas.

Programa: ■ → ■ → ■

Resultado: 

Un programa que pinta tres casilleros hacia la derecha, incluyendo el casillero inicial.

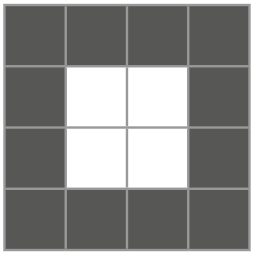
Es importante señalar a las y los estudiantes que, a pesar de que la ejecución la lleve a cabo una persona, dado su rol de autómata, solo podrá interpretar las instrucciones que estén formuladas según las reglas del lenguaje y ejecutar el programa tal cual está escrito (no podrá hacer cambios, aún cuando detecte que hay un error)[1].

Desarrollo >

El **propósito de este momento** es abordar desafíos que motiven el uso de repeticiones y procedimientos como estrategias de legibilidad.

Orientaciones

Dividimos la clase en grupos heterogéneos de dos o tres estudiantes.
El **primer desafío** consiste en escribir usando lápiz y papel un programa para realizar el siguiente dibujo (un cuadrado de cuatro casilleros de lado).
Mostramos a las y los estudiantes el resultado que se espera obtener y recordamos que tengan presentes las restricciones del lenguaje.



Luego de unos minutos de trabajo, les proponemos que compartan sus soluciones a los demás grupos.

■ → ■ → ■ → ■ ↓ ■ ↓ ■ ↓ ■ ← ■ ← ■ ← ■ ↑ ■ ↑ ■ ↑

Un programa posible para dibujar el cuadrado (iniciando en el primer casillero de la primera fila).

[1] Podemos retomar las nociones de instrucciones o comandos primitivos, ejecución y autómata abordadas en la secuencia "¿Qué es programar? Autómatas, programas y primitivas".

¿Hay alguna parte del programa que vean que se repite en sus soluciones? ¿Todos escribieron el programa de igual forma? ¿Qué semejanzas y diferencias hallan?

Con ese intercambio, entre todas y todos, se observará si se respetaron las restricciones y si identificaron o no patrones de repetición. Es posible que haya quienes hayan identificado los patrones sin necesidad de escribir toda la secuencia de instrucciones.

Por ejemplo, en la solución se repiten ■↓, ■←, y ■↑ tres veces consecutivas cada uno. A partir de esta observación, retomamos la noción de repetición y convenimos una forma de expresarla en el lenguaje que estamos usando. Por ejemplo, podemos acordar encerrar entre paréntesis los comandos que se quieren repetir e indicar cuántas veces debe ejecutarse a continuación. Por ejemplo, para expresar la secuencia ■→■→■→ que pinta tres casilleros a la derecha podemos escribir: (■→)3.

A continuación, las y los estudiantes reescriben o completan sus soluciones utilizando repetición.

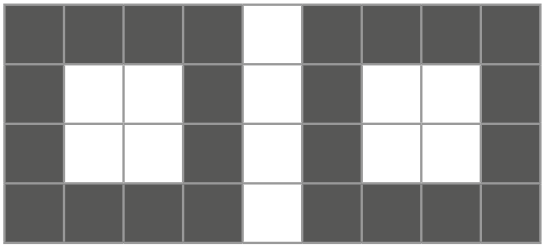
(■→)3
(■↓)3
(■←)3
(■↑)3

Un programa posible para dibujar el cuadrado escrito utilizando repeticiones.

Hacemos una puesta en común en la que les proponemos comparar las soluciones con y sin repetición.

Si no conocieran las dimensiones del cuadrado a dibujar, ¿en cuál de los programas lo podrían averiguar más fácilmente? ¿Por qué? Si tuvieran que cambiar el tamaño del cuadrado, ¿cuál de los programas sería más sencillo modificar?

Con este intercambio buscamos poner en evidencia el aporte de la repetición a la **claridad de un programa** y la **facilidad para modificarlo**.



El **segundo desafío** consiste en dibujar una figura compuesta por dos cuadrados de iguales dimensiones al del primer desafío. Mostramos a las y los estudiantes el resultado que queremos obtener.

El objetivo es recuperar el uso de procedimientos para aprovechar o reutilizar fragmentos de programas. Cuando aborden el desafío, es probable que surja rápidamente la necesidad de reutilizar un fragmento de programa existente. Cuando sea una preocupación compartida por todos los grupos, interrumpimos el trabajo para hacer una puesta en común y asociar esta necesidad con la definición de **procedimientos**[2].

Como hicimos con la repetición, acordamos una manera de definir y utilizar procedimientos dentro de los términos del lenguaje que estamos usando. Es posible que algunos grupos ya hayan definido una forma propia; en ese caso, invitamos a que los grupos compartan sus propuestas para que sean consideradas.

Una forma de definir procedimientos puede ser escribir un nombre y seguido de dos puntos indicar los comandos del cuerpo de la definición. Una vez hecho esto, usaremos el nombre del procedimiento para invocarlo en el programa.

Una vez acordado cómo se expresarán los procedimientos, proponemos a los y las estudiantes reescribir el programa que dibuja los dos cuadrados.

```
Dibujar un cuadrado:
(■ →)3
(■ ↓)3
(■ ←)3
(■ ↑)3

Al ejecutar:
Dibujar un cuadrado
(→ ■)5
Dibujar un cuadrado
```

Una solución posible en la que se invoca dos veces al procedimiento "Dibujar un cuadrado" construido a partir del programa del desafío anterior.

```
Dibujar un cuadrado:
(■ →)3
(■ ↓)3
(■ ←)3
(■ ↑)3

Posicionarse en el segundo
cuadrado:
(→ ■)5

Al ejecutar:
Dibujar un cuadrado
Posicionarse en el segundo
cuadrado
Dibujar un cuadrado
```

Otra solución posible es tomar la repetición (→■) 5 y hacer un nuevo procedimiento que se llame "Posicionarse en el segundo cuadrado".

Al utilizar un procedimiento en distintos lugares del programa, es clave que las y los estudiantes tengan en cuenta en qué casillero de la grilla deberá estar ubicado el autómata para que el procedimiento consiga el dibujo esperado. Para ello, deberán incluir en el programa

[2] Para repasar esta noción, se puede consultar la secuencia "Definimos nuestros bloques. Procedimientos y repetición simple".

el desplazamiento necesario antes de ejecutar por segunda vez el procedimiento.

Comparen las soluciones que utilizan procedimientos con las que no y discutan: si no supieran cuántos cuadrados hay en la figura, ¿en cuál de los programas sería más fácil averiguarlo? Si tuvieran que cambiar esta cantidad, ¿cuál de los dos programas sería más fácil de modificar?

Al igual que con el desafío anterior, comparamos las soluciones para evidenciar y señalar cómo, al utilizar procedimientos, el programa se vuelve más fácil de interpretar y modificar.

Cierre >

El **propósito de este momento** es evidenciar que los programas requieren ser interpretados por las computadoras y las personas, y comenzar a construir la noción de **legibilidad**.

Orientaciones

¿Para qué partes de las soluciones de los desafíos les resultó conveniente utilizar repeticiones? ¿Por qué? ¿Y procedimientos?

Recuperamos el trabajo durante la actividad para identificar que, tanto al indicar repeticiones como al definir procedimientos para usar más de una vez, estamos evitando escribir muchas veces un mismo fragmento de programa.

¿Cuál es la ventaja de evitar fragmentos de programa repetidos?

Nos interesa destacar que estas herramientas, además de la ventaja evidente de ahorrarnos tiempo en escribir instrucciones, hacen que **los programas sean más fáciles de interpretar por las personas**. Esto resulta en que sean más fáciles de corregir (porque es más fácil leerlos para identificar errores), sean más fáciles de modificar (porque es más fácil comprender dónde y qué modificación hay que hacer) y también sean mejores para construir en equipo (porque es más fácil comprender los programas producidos por las otras personas para aumentarlos o mejorarlos).

¿Para quién escribimos programas: para las personas o para las computadoras? ¿Entonces cómo debemos escribirlos?

El dilema con el que concluimos este cierre busca resaltar que, si bien la computadora o el autómata tiene que ser capaz de interpretar el programa (y por eso es importante formular las instrucciones

con precisión y respetando los términos del lenguaje), también es fundamental que las personas puedan interpretarlos para estudiarlos, para continuarlos, para mejorarlos, para modificarlos, etc. Para ello, al programar, recurrimos a las herramientas del lenguaje que nos permiten escribir programas más claros o **legibles**. Entendemos por legibilidad a la cualidad de un programa de ser fácilmente interpretable por una persona.

Actividad 2
Legibilidad

Continuando con la programación en papel y haciendo foco en la legibilidad de las soluciones, abordamos nuevos desafíos en los que la definición de procedimientos y su denominación será fundamental.

Objetivos >

Se espera que las y los estudiantes:

- Refuercen la legibilidad como una característica importante de los programas.
- Identifiquen el uso de procedimientos con denominaciones representativas y repeticiones para escribir programas que denoten una estrategia de solución.

Inicio >

El **propósito de este momento** es repasar brevemente las primitivas disponibles en el lenguaje con el que trabajamos en la **Actividad 1**, cómo se construyen los programas y las convenciones que definimos para expresar repeticiones y procedimientos.

Orientaciones

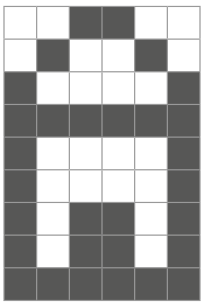
Invitamos a las y los estudiantes a recuperar estas ideas de su experiencia en la actividad anterior. Aprovechamos para conformar grupos heterogéneos pequeños.

Desarrollo >

El **propósito de este momento** es reforzar la importancia de la legibilidad en la construcción de programas más extensos.

Orientaciones

Presentamos una nueva figura que representa una casa para escribir un programa que la dibuje.



Podemos comenzar por invitarlos a que nombren los tramos o partes a dibujar, ya que posiblemente los usen para nombrar los procedimientos, por ejemplo, “techo”, “pared”, “puerta”, “piso”.

Prestamos atención al trabajo de las y los estudiantes, observando (y alentando a) que **definan una estrategia de solución** y que **utilicen procedimientos** y repeticiones para expresar fielmente la estrategia.

A medida que van terminando, intercambian su programa con otro grupo para que lo ejecute para ponerlo a prueba. Quienes lo ejecuten deberán entregar el resultado.

```
Al ejecutar:
(↓)3
Dibujar el techo
Dibujar las paredes y el
piso
(↓)5
(→)2
Dibujar la puerta
```

```
Al ejecutar:
(↓)3
Dibujar pared izquierda
Dibujar piso y puerta
Dibujar pared derecha
Dibujar techo
```

Dos ejemplos (fragmentos) de soluciones posibles para dibujar la casa siguiendo distintas estrategias.

En el caso de que no se realice el dibujo esperado, podrá deberse a errores en el programa o a errores en la ejecución. En ambos casos, deberán registrar cómo pueden mejorar el programa. Por ejemplo, "El programa dibuja correctamente las paredes, pero el techo no está bien ubicado" o "El programa es correcto, pero para ejecutar sin equivocarse la parte que dibuja el techo hay que prestar mucha atención".

Cuando todas y todos hayan finalizado, invitamos a que compartan algunas de las soluciones que elaboraron para compararlas en una puesta en común. También, alentamos a que aquellos y aquellas estudiantes que hayan corregido errores en sus programas compartan cómo los identificaron y de qué manera los corrigieron.

¿En cuáles programas pueden anticipar el dibujo resultante? ¿Cuáles les parece que son más fáciles de ejecutar sin cometer un error? ¿Por qué? ¿En cuáles es más fácil saber en qué orden se dibuja cada parte de la casa? Si tuvieran que elegir uno para darle a alguien que no sabe qué figura hay que dibujar, ¿cuál elegirían?

Estas preguntas apuntan a analizar las soluciones compartidas para reforzar la importancia de la legibilidad. Señalamos que cuando podemos leer un programa y comprender sin mucho esfuerzo lo que hace, cómo está estructurado y ejecutarlo sin inconvenientes, podemos afirmar que el programa es **legible**. También señalamos que la legibilidad es importante para reflejar cómo quiso resolver el problema quien lo escribió, es decir, la **estrategia de solución** elegida. Esto se volverá

fundamental cuando trabajemos con programas escritos por otras personas (tanto para estudiarlos como para modificarlos).

Podemos brindar un tiempo para revisar y mejorar las soluciones incorporando estas ideas si hiciera falta.

Cierre >

El **propósito de este momento** es valorar la legibilidad de un programa para poder expresar fielmente la estrategia de solución elegida, facilitar la detección de errores e implementación de modificaciones, y permitir que otras personas puedan comprender con el menor esfuerzo posible su estructura y su propósito.

Orientaciones

Mostramos tres programas para interpretar a golpe de vista y anticipar cuál es el dibujo resultante.

Programa 1

```
Al ejecutar:
(■ ↑)7 ← ←(↓)3 ■ ↓ ■ → ■ ↓ ■ → → ↑
■ ↑ ■ → ■ ↑ ■ ↑ ↑ ← (■ ←)3 (■ ↑)3 (■ →)4 (↓ ■)2
(↑)3 (■ ← ←)2 ■
```

Programa 2

```
Al ejecutar:
Dibujar tallo
(←)2
(↓)3
Dibujar hoja izquierda
(→)2
↑
Dibujar hoja derecha
(↑)2
←
Dibujar pétalos

Dibujar tallo:
(■ ↑)7

Dibujar hoja izquierda:
■ ↓ ■ → ■ ↓ ■

Dibujar hoja derecha:
■ ↑ ■ → ■ ↑ ■

Dibujar pétalos:
(■ ←)3(■ ↑)3(■ →)4
(↓ ■)2
(↑)3 (■ ← ←)2 ■
```

Programa 3

```
Al ejecutar:
Dibujar una parte
(←)2
(↓)3
Dibujar parte izquierda
(→)2
↑
Dibujar parte derecha
(→)2
←
Dibujar el resto

Dibujar una parte:
(■ ↑)7

Dibujar parte izquierda:
■ ↓ ■ → ■ ↓ ■

Dibujar parte derecha:
■ ↑ ■ → ■ ↑ ■

Dibujar el resto:
(■ ←)3(■ ↑)3(■ →)4
(↓ ■)2
(↑)3 (■ ← ←)2 ■
```

Volvemos a mirar los programas para analizarlos en términos de legibilidad.

¿Pudieron averiguar qué dibuja cada programa y la manera en la que resuelve el problema? ¿Por qué?

Recuperamos la experiencia para identificar y valorar aquellas herramientas de programación que facilitaron la interpretación de los programas (procedimientos, repeticiones, división en subproblemas y la utilización de denominaciones representativas).

Para reforzar estas ideas podemos volver a mostrar los programas y observar que al ejecutarlos se llevan a cabo las mismas instrucciones en el mismo orden; la diferencia sustancial radica en la utilización y la denominación de los procedimientos para reflejar la **estrategia de solución** expresando la forma que eligió quien lo elaboró para resolver el problema.

¿Qué programa elegirían como definitivo para realizar este dibujo? ¿Por qué? ¿Y si tuvieran que compartirlo con otras personas? ¿Y si tuvieran que corregir algún error? ¿Y si tuvieran que modificar una parte?

El objetivo de estas preguntas es que, a modo de conclusión, las y los estudiantes recuperen las ventajas de los programas legibles que surgieron en reflexiones anteriores y las justifiquen para la elección del Programa 2. En este programa, por ejemplo, si se hubiera omitido un movimiento será más fácil encontrar el error porque ya sabemos qué debe hacer cada parte; o si quisiéramos modificar alguna de las partes del dibujo, por ejemplo, agregando pétalos o haciendo las hojas más grandes será más claro dónde hacer la modificación. En este sentido, valoramos la legibilidad porque al comprender lo que un código hace, significa que los programas sean mucho más fáciles de estudiar y **compartir** (son más fáciles de interpretar por otras personas), **corregir** (es más fácil encontrar dónde está el error) y **modificar o adaptar** (es más claro identificar qué sección del programa debemos cambiar).

Actividad 3
Reutilización

Las y los estudiantes resuelven desafíos en los que se vuelve clave la definición de procedimientos para reutilizar fragmentos de programa.

Objetivos >

Se espera que las y los estudiantes:

- Reconozcan en problemas distintos un subproblema repetido y definan un procedimiento que lo resuelva para reutilizarlo en las soluciones a los otros problemas.
- Asocien la definición de procedimientos y su reutilización con la construcción colectiva de programas.

Inicio >

El **propósito de este momento** es recuperar, si hiciera falta, el lenguaje definido en las actividades anteriores y la modalidad de trabajo con los programas para dibujar, así como también la importancia de construir programas legibles.

Orientaciones

Invitamos a las y los estudiantes a que recuperen sus experiencias previas para recuperar estos acuerdos.

Desarrollo >

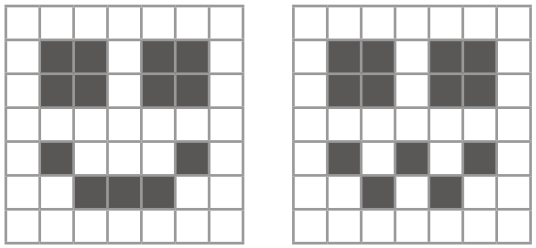
El **propósito de este momento** es resolver problemas reutilizando código propio y de otras personas, y reconocer la importancia de la definición de procedimientos para lograrlo.

Orientaciones

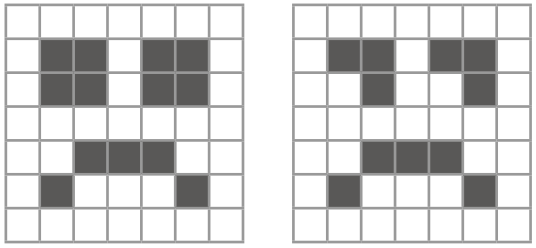
Proponemos la organización de la clase en pequeños grupos heterogéneos. Presentamos el nuevo desafío en el que cada grupo deberá dibujar dos "caritas". Podemos mostrar las opciones y que los grupos elijan, prestando atención a que estén todos los pares cubiertos. Aclaramos que cada grupo deberá hacer dos programas (uno para cada carita) y que pue-

den invocar en un programa cualquiera de los procedimientos que hayan definido en el otro.

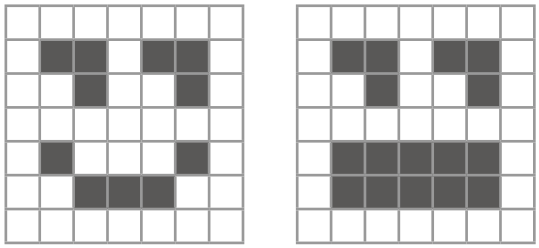
Par 1



Par 2



Par 3



Luego de un tiempo suficiente para la programación, se ponen en común los resultados y comienza el momento de reflexión.

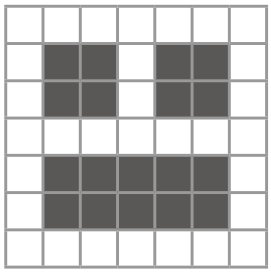
¿Definieron procedimientos? ¿Por qué? ¿Para resolver qué partes? ¿En qué otro/s desafíos de programación les pasó algo parecido? ¿Cuál es la diferencia con este problema?

Como primera aproximación a la idea de **reutilización**, identificamos que es conveniente definir procedimientos para las partes o subproblemas que se repiten. Podemos retomar experiencias previas con otros desafíos (por ejemplo, con el desafío ***Nuevos comandos*** de la secuencia "Definimos nuestros bloques. Procedimientos y repetición simple"), en las que definimos un procedimiento para un mismo subproblema se repitía dentro del desafío. En este deasfío, podría ser para dibujar cada uno de los ojos dos veces. Además, en esta actividad, aprovechamos **los procedimientos definidos para un problema** (la carita de la izquierda) **para resolver un problema nuevo** (la carita de la derecha).

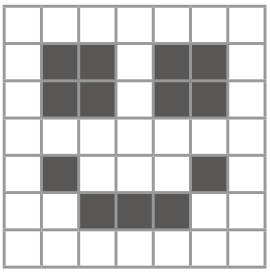
Si quisieran agregar un programa para que dibuje alguna de las caras que ya tienen, pero con un gorro, ¿qué podrían hacer para aprovechar al máximo lo que ya programaron?

Esta pregunta busca dar cuenta de que los **procedimientos son una herramienta fundamental para reutilizar** fragmentos de programas en programas distintos, que es la novedad que introduce esta actividad. En este caso, podría definirse un procedimiento que contenga el programa que ya hicieron para alguna de las caras y crear un programa nuevo que utilice ese procedimiento, además de uno nuevo "dibujar gorro".

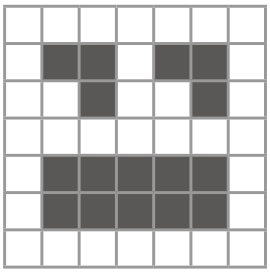
En una segunda instancia, con el propósito de generar **una situación en la que se ponga en juego la reutilización de programas como parte de una construcción colectiva**, proponemos a los grupos que piensen otros dibujos que podrían hacer en una cuadrícula similar, aprovechando los programas que fueron creados por todo el curso durante la actividad. Para esto, deberán relevar los programas elaborados por sus compañeros y compañeras y revisar la definición de los procedimientos en los propios para que otros grupos puedan reutilizarlos. Podemos registrar este relevamiento, por ejemplo, en el pizarrón, para que quede a disposición de todo el curso durante la actividad.



```
Al ejecutar:
Dibujar ojos cua-
drados
Ir al comienzo de
la boca
Dibujar boca
alargada
```



```
Al ejecutar:
Dibujar ojos
cuadrados
Ir al comienzo
de la boca
Dibujar boca
sonriente
```



```
Al ejecutar:
Dibujar ojos
felices
Ir al comienzo de
la boca
Dibujar boca
alargada
```

Ejemplo de cómo dibujar una nueva cara (izquierda) reutilizando procedimientos definidos en los programas preexistentes (derecha).

Luego, invitamos a que compartan los dibujos que imaginaron y los programas que los dibujan, señalando dónde intervienen los programas ya hechos, de quiénes los tomaron y cuál era su función original. Si se cuenta con el tiempo, proponemos que los grupos se intercambien los programas que escribieron, los ejecuten y luego corroboren el dibujo resultante. En caso de no obtener el dibujo esperado, alentamos a que los grupos conversen e identifiquen si se trató de un problema de ejecución o de programación e intenten corregirlo.

Cierre >

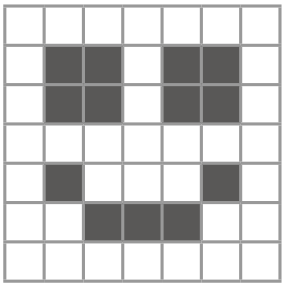
El **propósito de este momento** es recuperar la experiencia de la solución del desafío para asociar la legibilidad y la definición de procedimientos con la posibilidad de reutilizar programas hechos por otras personas para resolver nuevos problemas.

Orientaciones

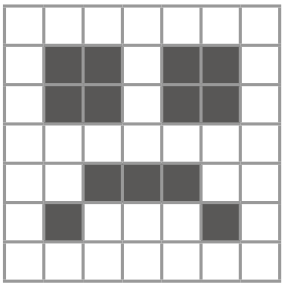
¿Qué características de los programas de las compañeras y los compañeros facilitaron la construcción de los nuevos programas?

A partir de esta pregunta, buscamos reforzar, por un lado, la importancia de la **definición de procedimientos** para aislar fragmentos de programas que pueden ser utilizados por otras personas y, por otro, que la **legibilidad** de un programa facilita su **reutilización** y, por lo tanto, la **construcción colectiva**.

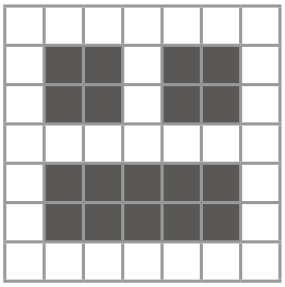
Con el propósito de evidenciar que como la **reutilización** favorece también la modificación y la mejora de los programas, mostramos los siguientes dibujos y los fragmentos de programas que los resuelven y pedimos que hipoteticen cómo habría que modificar los programas si quisiéramos que los dibujos tengan los ojos más grandes o de otra forma, y que comparen hacerlo en este caso, en el que hay un procedimiento que se reutiliza con el caso en el que cada programa tuviera las instrucciones completas para dibujar los ojos.



```
Al ejecutar:
Dibujar ojos cuadrados
Ir al comienzo de la boca
Dibujar boca sonriente
```



```
Al ejecutar:
Dibujar ojos cuadrados
Ir al comienzo de la boca
Dibujar boca triste
```



```
Al ejecutar:
Dibujar ojos cuadrados
Ir al comienzo de la boca
Dibujar boca alargada
```

En este caso, habría que modificar el procedimiento `Dibujar ojos cuadrados una única vez`, mientras que en el otro caso, habría que analizar tres programas y realizar tres modificaciones. La conclusión de esta reflexión es observar cómo la definición de procedimientos y

su reutilización hace que los programas también sean más fáciles de modificar: si el procedimiento necesita un cambio o una corrección basta con modificar el programa en un único lugar (en la definición del procedimiento). De la otra manera, para efectuar la modificación deberíamos inspeccionar la totalidad los tres programas, detectar cada vez que aparece el fragmento que queremos modificar y realizar las mismas modificaciones en todos estos fragmentos.

Actividad 4
Nuestros dibujos

Las y los estudiantes pondrán en práctica todos los conceptos tratados en las actividades anteriores a partir de pensar un dibujo para que otro grupo escriba un programa para realizarlo.

Objetivos >

Se espera que las y los estudiantes:

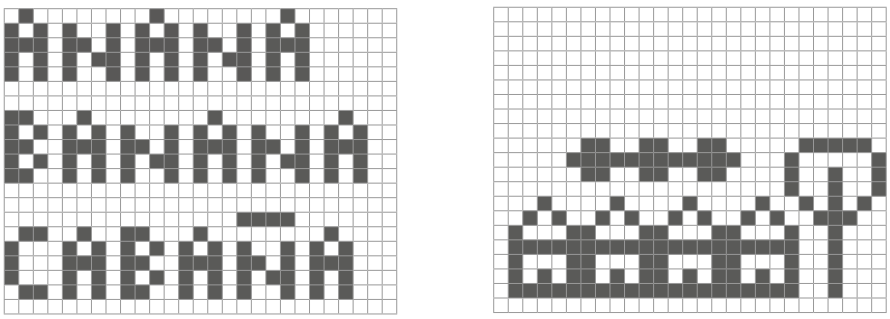
- Reconozcan qué características de un problema hacen que se vuelva clave la legibilidad y la reutilización en el programa que lo resuelve.
- Elaboren un programa legible para un problema aprovechando la reutilización de código.

Inicio >

El **propósito de este momento** es identificar en la experiencia de las actividades anteriores qué características de los dibujos se asociaban con los conceptos trabajados en la secuencia (repeticiones, legibilidad y reutilización).

Orientaciones

En pequeños grupos heterogéneos, las y los estudiantes diseñan un dibujo que se pueda realizar en una hoja cuadriculada con el lenguaje utilizado en las actividades anteriores.



Ejemplos de dibujos que motivan la división en subproblemas y la reutilización.

Es importante señalar que al definir el dibujo tengan en mente una estrategia de solución para realizar el dibujo, anticipen la extensión del código

e incorporen patrones que habiliten repeticiones y reutilizaciones. Recorremos el trabajo de los grupos y sugerimos mejoras para reforzar estos aspectos.

Desarrollo >

El **propósito de este momento** es que las y los estudiantes pongan en práctica sus conocimientos de programación y, a la vez, profundicen las asociaciones entre características de los problemas y la necesidad de aplicar los conceptos abordados en la secuencia.

Orientaciones

En un **primer momento**, las y los estudiantes escriben un programa para realizar el dibujo que pensaron. En esta instancia es probable que detecten que el dibujo necesita modificaciones para que el programa incluya las características deseadas (estrategia en partes, patrones para reutilizar código, etc.). Es importante que, tanto el dibujo como el programa, no sean conocidos por el resto de los grupos y estén registrados en hojas o documentos separados.

En un **segundo momento**, cada grupo intercambia su dibujo con otro para que escriba un programa para realizarlo. A medida que las y los estudiantes escriben los programas, sugerimos observar e identificar el uso de procedimientos, nombres representativos y repetición. En el caso de observar la ausencia de algunos de estos conceptos, como motivación les podemos recordar que la solución será leída por otras personas para realizar el dibujo que pensaron. También es importante valorar aquellos programas donde estén presentes estos conceptos que aporten a la legibilidad.

En un **tercer momento**, a medida que los y las estudiantes finalizan sus programas, los deberán intercambiar con otro grupo para que lo ejecuten para obtener el dibujo resultante. Luego, deberán comparar el resultado obtenido con el dibujo esperado. Se alienta a que las y los estudiantes compartan su experiencia y valoren en conjunto aspectos positivos de los programas que favorecieron su comprensión (como la división en subtareas o la elección de nombres) y aquellas cosas que se podrían mejorar (y cómo las podrían mejorar).

En una segunda instancia, se comparan los programas que escribieron los grupos que recibieron el dibujo con los que elaboró el grupo que lo pensó para comparar estrategias. En este punto, es importante asociar características del dibujo con decisiones sobre los programas (por ejemplo, usar una repetición para un patrón repetitivo).

Esta actividad permite, por un lado, la percepción del avance en la programación en el grupo y en sus integrantes; pero además, facilita la inmersión en el concepto de legibilidad al comprobar si lo que un o una estudiante ha pensado en el desarrollo es interpretado de la misma manera por quien lo recepta o, al menos, hay una idea principal que queda clara en cada proyecto que se ha creado. Finalmente, la comparación entre los programas modelo y los obtenidos es una oportunidad para reforzar la asociación entre características de los problemas (en este caso, los dibujos) y los programas que los resuelven.

Retomaremos la estrategia de proponer la creación de problemas de programación para afianzar los conceptos ya abordados y profundizar las reflexiones sobre su uso asociado a problemas específicos en futuras secuencias, aprovechando el creador de desafíos de Pilas Bloques.

Cierre >

El **propósito de este momento** es brindar un espacio de metacognición para identificar en el recorrido de la actividad las instancias en las que tomaron decisiones que involucran los conceptos abordados en la secuencia.

Orientaciones

En un diálogo conjunto, rescatamos los distintos momentos de la actividad en los que se pusieron en juego las nociones abordadas en la secuencia. Invitamos a que relaten qué decisiones tomaron y asociadas a qué nociones.

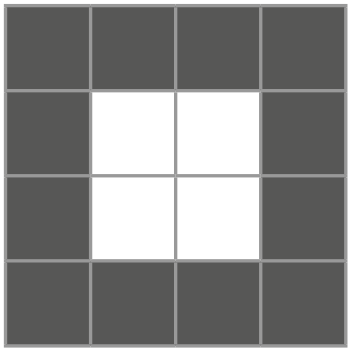
Esperamos que en el intercambio aparezcan las siguientes cuestiones.

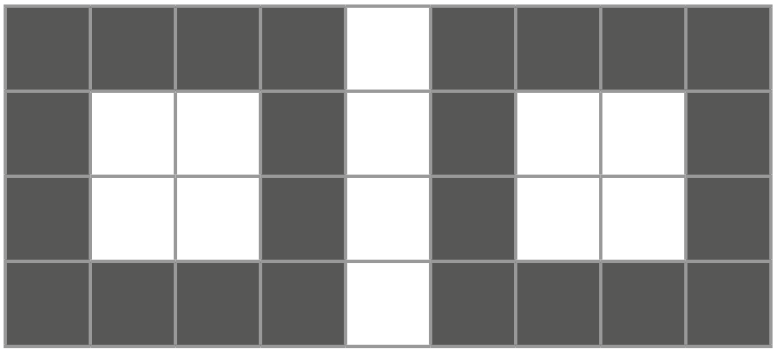
- Al diseñar el dibujo, incorporaron **patrones o elementos repetidos** que motivaron la reutilización de **procedimientos** y/o el uso de **repeticiones**. También, pueden haber elaborado un dibujo con distintos elementos o partes diferenciadas para motivar una **estrategia** con varias partes que refuerce la importancia de la definición de **procedimientos con nombres representativos**.
- Al recibir el dibujo tuvieron que identificar estas características en el problema y recuperar las nociones abordadas en la secuencia para escribir el programa.
- Al comparar el dibujo esperado con el resultante, es probable que hayan surgido errores que se debieron a que el programa no era tan fácil de interpretar. En la devolución al grupo que realizó el programa se pusieron en juego argumentos que involucran ideas de **legibilidad**.

- Al comparar el programa elaborado con el modelo, es probable que hayan surgido otras estrategias de solución más allá de la pensada por el equipo. Aprovechamos esta situación para poner de manifiesto las numerosas formas en las que se puede resolver un problema y cómo algunas de ellas aprovechan mejor determinadas herramientas y conceptos de programación.

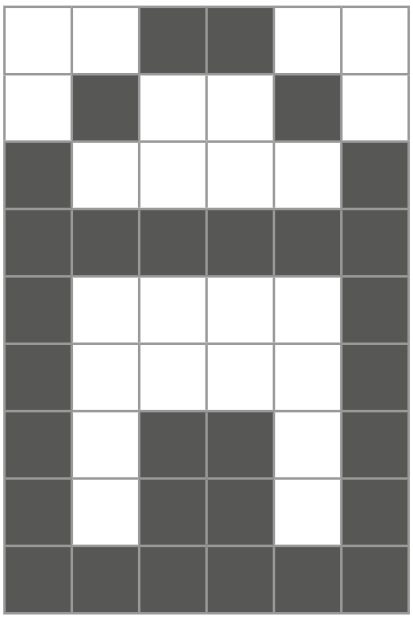
Anexo

Actividad 1





Actividad 2



Programa 1

Al ejecutar:
(■ ↑)7 ← ←(↓)3 ■ ↓ ■ → ■ ↓ ■ → → ↑
■ ↑ ■ → ■ ↑ ■ ↑ ↑ ← (■ ←)3 (■ ↑)3 (■ →)4 (↓ ■)2
(↑)3 (■ ← ←)2 ■

Programa 2

Al ejecutar:
Dibujar tallo
(←)2
(↓)3
Dibujar hoja izquierda
(→)2
↑
Dibujar hoja derecha
(↑)2
←
Dibujar pétalos

Dibujar tallo:
(■ ↑)7

Dibujar hoja izquierda:
■ ↓ ■ → ■ ↓ ■

Dibujar hoja derecha:
■ ↑ ■ → ■ ↑ ■

Dibujar pétalos:
(■ ←)3(■ ↑)3(■ →)4
(↓ ■)2
(↑)3 (■ ← ←)2 ■

Programa 3

Al ejecutar:
Dibujar una parte
(←)2
(↓)3
Dibujar parte izquierda
(→)2
↑
Dibujar parte derecha
(→)2
←
Dibujar el resto

Dibujar una parte:
(■ ↑)7

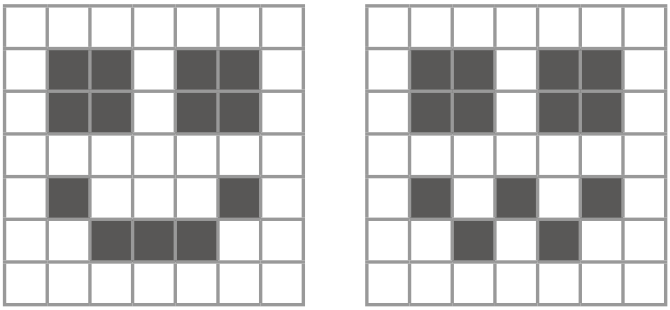
Dibujar parte izquierda:
■ ↓ ■ → ■ ↓ ■

Dibujar parte derecha:
■ ↑ ■ → ■ ↑ ■

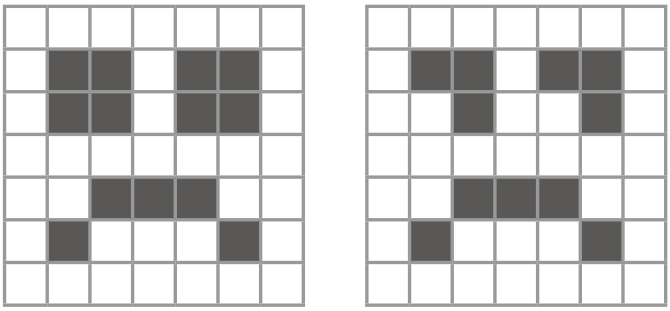
Dibujar el resto:
(■ ←)3(■ ↑)3(■ →)4
(↓ ■)2
(↑)3 (■ ← ←)2 ■

Actividad 3

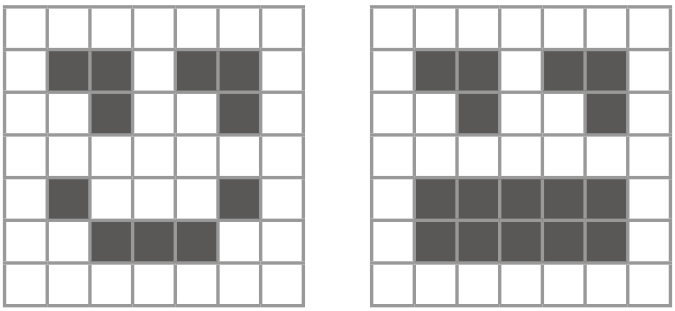
Par 1

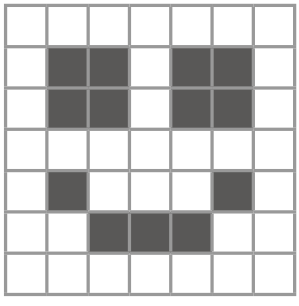


Par 2

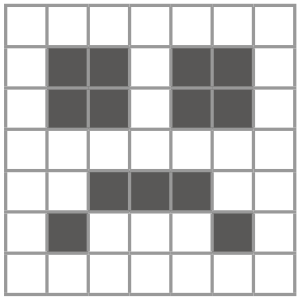


Par 3

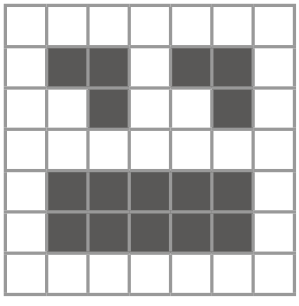




Al ejecutar:
Dibujar ojos cuadrados
Ir al comienzo de la boca
Dibujar boca sonriente



Al ejecutar:
Dibujar ojos cuadrados
Ir al comienzo de la boca
Dibujar boca triste



Al ejecutar:
Dibujar ojos cuadrados
Ir al comienzo de la boca
Dibujar boca alargada