



Repetición Simple, comunicación y legibilidad

Teoría 1.C.

A trabajar

Realizamos del libro del “Ciclo de Secundaria”, el ejercicio “No me canso de saltar”, de la sección “Autómatas, Comandos, Procedimientos y Repetición”.



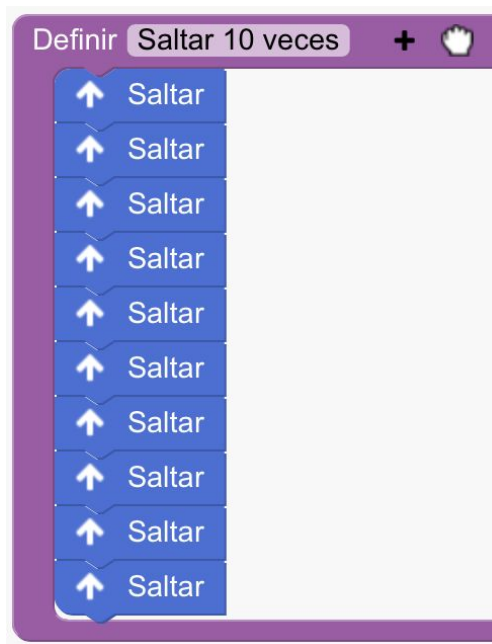
No me canso de saltar: ¿Qué aprendimos?



Primera aproximación
!!!Horrible!!!

Ya a esta altura, es claro que esta no es
una solución adecuada.

No me canso de saltar: ¿Qué aprendimos?



Segunda aproximación

Mucho mejor. Hace uso de las herramientas aprendidas, y deja más clara la idea. Se puede mejorar haciendo más procedimientos.

No me canso de saltar: ¿Qué aprendimos?

Al empezar a ejecutar

Saltar 30 veces

Definir Saltar 10 veces

Saltar 5 veces

Saltar 5 veces

Definir Saltar 30 veces

Saltar 10 veces

Saltar 10 veces

Saltar 10 veces

Definir Saltar 5 veces

↑ Saltar

↑ Saltar

↑ Saltar

↑ Saltar

↑ Saltar

Tercera aproximación

¡Muy bien!. Las ideas quedan claras, y bien expresadas en términos de procedimientos, pero no descubrieron la nueva herramienta disponible.

No me canso de saltar: ¿Qué aprendimos?



Cuarta aproximación

¡Muy bien!. Descubrieron la nueva herramienta disponible. La repetición es una nueva forma de organizar el código. Sin embargo, no descubrieron que se podía modificar el número de veces a repetir.

No me canso de saltar: ¿Qué aprendimos?



Quinta aproximación

¡Excelente!. Esta sería una excelente solución a este problema, que hace muy buen uso de la nueva herramienta.

No me canso de saltar: ¿Qué aprendimos?

Pero no olvidar que una buena solución debe ser clara a la hora de comunicar y la mejor forma de hacerlo es trabajar con procedimientos.

PROPÓSITO
(Qué resuelve
nuestro programa)

ESTRATEGIA
(Cómo lo resuelve)



Una **única**
invocación a un
procedimiento cuyo
nombre deje claro
qué hace nuestro
programa.

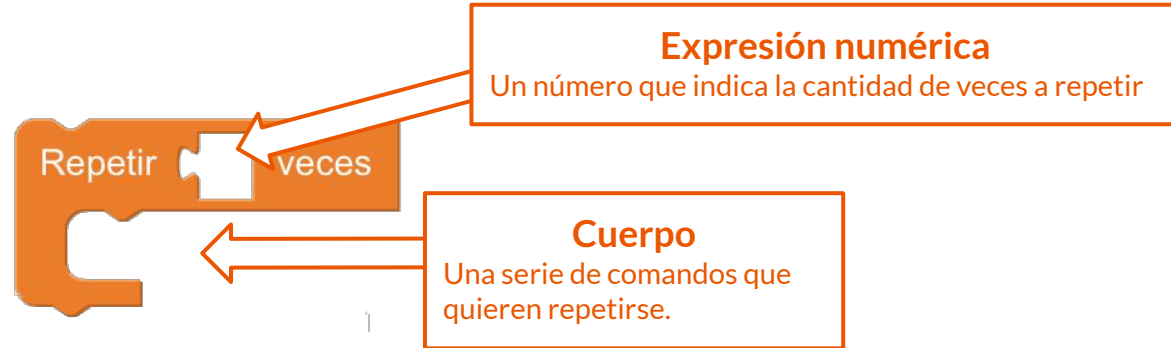
A partir de ahora vamos a
distinguir propósito de
estrategia de este modo



Repetición Simple

- La repetición simple es un **comando**. Notar que encastra dentro de otros cuerpos, igual que los comandos primitivos (los azules) o los definidos por procedimiento (los violetas).
- Este comando tiene una semántica especial, ya que permite cambiar la forma en la que se ejecuta el código (**cambia el flujo natural del programa**)
- El comando tiene 2 partes importantes, un **cuerpo** y una **expresión numérica**.
- Es lo que llamamos un **comando compuesto**, ya que es un comando que tiene un cuerpo.

Repetición Simple





Expresiones

- Una **expresión** es la **descripción de información** (de un **dato**, un **valor**).
- Las expresiones no pueden estar solas en un cuerpo (En PilasBloques es imposible hacerlo, porque directamente no encastran, ya que su forma es distinta a la de los comandos).
- Sirven **para brindar información a los comandos**, y alterar de forma coincidente la semántica del mismo.
- Hasta ahora, solo los números literales, que usamos en la repetición.

Expresiones numéricas literales



Notar que el número se puede cambiar, escribiendo cualquier otro número deseado.

El entorno permite ingresar números naturales, enteros y hasta racionales.

Por ahora solo usamos esta expresión para las repeticiones, por lo que ¿tiene sentido decir “repetir -4 veces”? ¿Y “repetir 7,2 veces”?

A pesar de que el entorno permita ciertos números, no cualquiera tiene sentido.



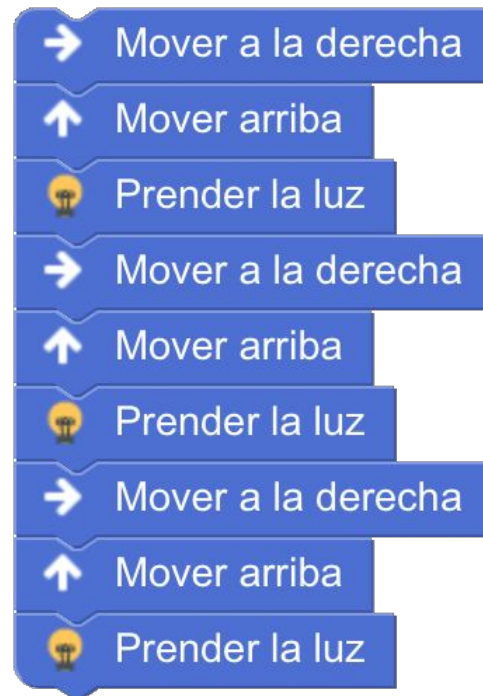
Control de flujo: Repetición

- La **REPETICIÓN** es otra forma en la que puede organizarse el código, además de la secuencia.
- La semántica de un bloque de repetición implica ejecutar los bloques del cuerpo de ese comando, tantas veces como sea necesario hasta completar la repetición. En la repetición simple, tantas veces como indica la expresión numérica asociada.
- Notar que los comandos dentro de la repetición se ejecutan en secuencia.

Control de flujo: Repetición



=



Control de flujo: Repetición



=



Control de flujo: Repetición



≠



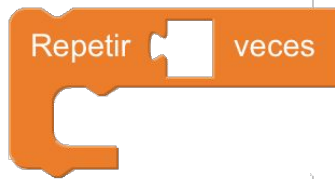
Control de flujo: Repetición



≠



En papel: nuevas palabras claves y formas



Repetir <expresión numérica> **Veces**

| <cuerpo>

En papel: nuevas palabras claves y formas

Un ejemplo concreto:



Repetir 3 Veces

Mover a la derecha

Mover arriba

Prender la luz



Momento de dudas o consultas

A trabajar

Realizamos del libro del “Ciclo de Secundaria”, los ejercicios “El recolector de estrellas” de la sección “Autómatas, Comandos, Procedimientos y Repetición”.

El recolector de estrellas: ¿Qué aprendimos?

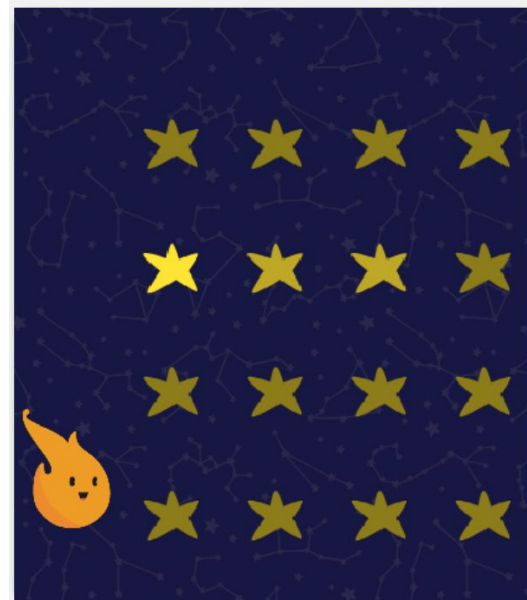
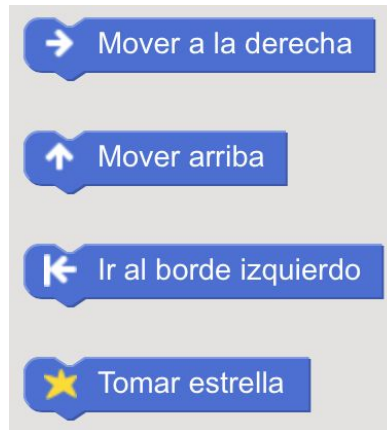
Nuevamente, ¿infinitas soluciones?



El recolector de estrellas: ¿Qué aprendimos?

Las primitivas disponibles en este ejercicio hacen que algunas estrategias no sean posibles.

Por ejemplo, no se puede elegir comenzar por las estrellas de arriba, pues no hay primitiva para mover hacia abajo.



El recolector de estrellas: ¿Qué aprendimos?



Primera aproximación
!!!Horrible!!!

¿Y los procedimientos?



iiiProgramar es comunicar!!!

Anidación de código



Vamos a incorporar otra buena práctica a la programación

Repeticiones anidadas

Cuando eso sucede decimos que se está “anidando” (la idea de colocar un bloque dentro de otro bloque, en un cuerpo).

Anidación de código



Como regla general podemos decir que:

**¡NUNCA, NUNCA, JAMÁS DE LOS
JAMASES DEBERÍA HABER UN BLOQUE
NARANJA DENTRO DE OTRO!**

Anidación de código

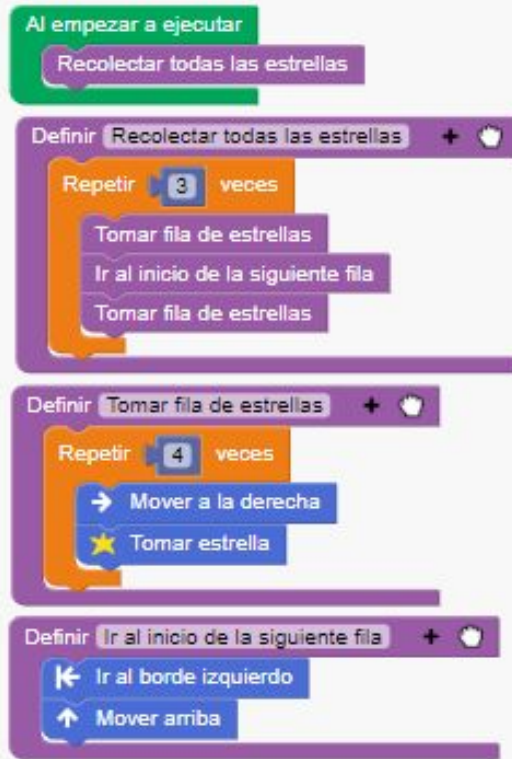


En realidad, siempre van a haber elementos anidados, como “Mover a la derecha” que está anidado dentro del repetir, pero el problema son los “bloques naranjas”.

Estos bloques cambian el flujo del programa, y al anidarlos es difícil entender qué hace el programa sin tener que ejecutarlo en nuestra cabeza.

Dicho de otra forma, anidar estructuras de control, **no comunica correctamente las ideas**.

El recolector de estrellas: ¿Qué aprendimos?



Segunda aproximación
!!!Hermosa!!!

Los procedimientos ayudan a entender
claramente las ideas.

Notar como en la estrategia se aprecia la idea de
la repetición de recoger por filas, y se ve
claramente el caso de borde.



Momento de dudas o consultas

La importancia de los nombres



¿Y en este caso?

El código es “casi” igual que la versión anterior, pero cambian los nombres de los procedimientos.

¿Queda clara la estrategia?



La importancia de los nombres

- Hay nombres de procedimiento buenos, y nombres malos.
- Recordamos:
 - Un procedimiento es una forma de definir un comando.
 - Un comando es la descripción de una acción
 - ¿Con qué expresamos acciones en el español?



La importancia de los nombres

- Hay nombres de procedimiento buenos, y nombres malos.
- Recordamos:
 - Un procedimiento es una forma de definir un comando.
 - Un comando es la descripción de una acción
 - ¿Con qué expresamos acciones en el español?

Los **verbos** son la forma que tenemos de expresar acciones. Un buen nombre de procedimiento debería comenzar con un verbo, para describir claramente la acción que se va a realizar.

Además, usamos verbos en infinitivo (ej. Mover, en lugar de Mueve).

¿Cuáles son buenos nombres?

- Tomar las estrellas de la fila
- Recolectar
- Recolecta
- Recolectar estrellas
- Estrellas
- Recolectar estrellas fila
- Juntar estrellas de la fila
- Tomar muchas estrellas



¿Cuáles son buenos nombres?

- Tomar las estrellas de la fila
- Recolectar
- Recolecta
- Recolectar estrellas
- Estrellas
- Recolectar estrellas fila
- Juntar estrellas de la fila
- Tomar muchas estrellas



Tratemos de escribir en español,
y no sonar como neandertales.
"Recolectar estrellas de la fila"
hubiera sido buena opción.

Nombres y propósito

Todo procedimiento (y programa) tiene un propósito. Es decir, “hace algo”, resuelve un problema puntual.

El **nombre** de un procedimiento debería reflejar claramente el **propósito** del mismo.

En ese sentido “Recolectar estrellas” cumple el criterio de comenzar con un verbo, pero el propósito no queda claro. ¿Está hablando de todas las estrellas del escenario? ¿De una columna? ¿De la fila?

Lo mismo pasa con “Tomar muchas estrellas” ¿Cuánto es muchas? ¿Todas las de la fila? ¿Algunas?






Nombres y propósito

Como regla del pulgar podemos decir que un nombre es bueno si al leerlo, nos queda más que claro qué es lo que va a suceder en el escenario, sin siquiera tener que mirar el código de ese procedimiento.

Podemos decir que las reglas son:

- Comienza con un verbo en infinitivo
- Deja en claro el propósito del procedimiento
- Está escrito en español, claro y legible



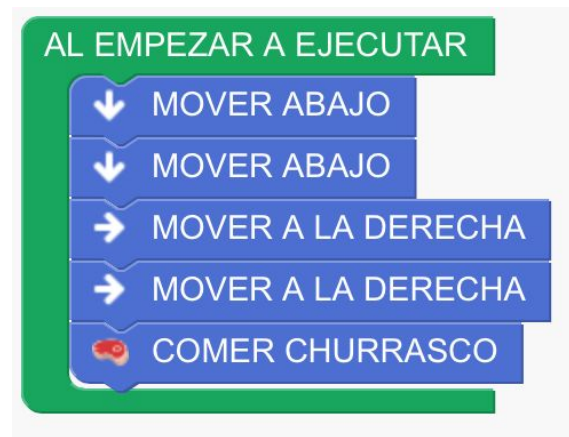
Agregamos entonces a las buenas prácticas, el de colocar buenos nombres a los procedimientos



Momento de dudas o consultas

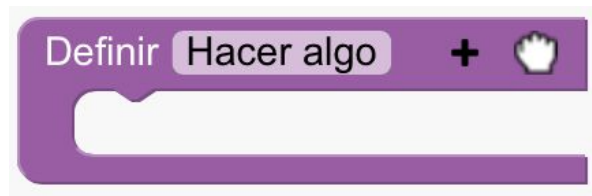
Repaso

- Un **programa** es una **descripción** de la **solución** a un **problema computacional**.
- Un **problema computacional** es aquel que puede expresarse como una **transformación de estado**.
- En PilasBloques lo expresamos mediante bloques que se encastran entre sí, para expresar un cambio de estado en el escenario.
- Todo programa tiene un **punto de entrada**.
- Los elementos fundamentales del programa son los **comandos** (descripciones de acciones).
- Los comandos se organizan en **secuencia**, y la solución se ejecuta según esa secuencia.
- Hay **infinitos** programas que solucionan un problema. Decimos que dos programas que solucionan el mismo problema son **equivalentes**.



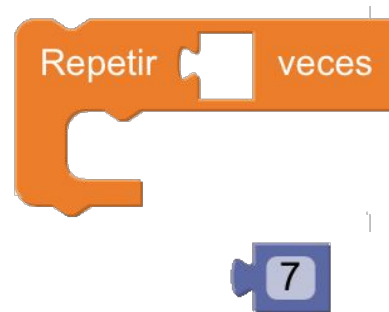
Repaso

- Los **procedimientos** son una herramienta del lenguaje que permite definir nuevos comandos.
- Tienen un **cuerpo** y un **nombre**.
- **El nombre debe ser claro y legible, comenzar con un verbo en infinitivo y estar relacionado a su propósito.**
- La **definición** va por un lado, y el **uso** (invocación o llamada) va en el cuerpo de algún otro bloque.
- Aportan claridad, legibilidad y modificabilidad al código.
- Pueden ser **reutilizados** muchas veces.
- Permiten transmitir claramente las ideas pensadas en nuestra **estrategia**.
- Permiten separar el problema (tarea) en partes más pequeñas para su más fácil resolución (**subtareas**).



Repaso

- La **repetición simple** es una herramienta del lenguaje que **permite cambiar el flujo** del programa (**estructura de control de flujo**).
- Permite estructurar el código de una forma distinta a la secuencia.
- Es un comando (y se puede usar junto con otros comandos en un cuerpo), pero tiene a su vez un cuerpo (es un **comando compuesto**)
- Espera una **expresión numérica** para indicar la cantidad de veces a repetir.
- Una **expresión** en la descripción de un **dato** (un **valor**, **información**)
- **No hay que anidar repeticiones de manera directa (ni ninguna otra estructura de control).**





Recordatorio:

iiiProgramar es comunicar!!!

Tus programas deberían quedar claros a partir de la lectura. Es importante.

Sí comunica mal, no es adecuado. Sí no es adecuado, no sirve para la materia.



Recordatorio: **iiiProgramar es comunicar!!!**

- Tus programas deberían quedar claros a partir de la lectura.
- Sí leo el punto de entrada, tiene que quedar más que explícita la estrategia elegida.
- Usamos procedimientos para la claridad, legibilidad y expresar la estrategia.
- Es importantísimo elegir nombres adecuados para los procedimientos que definimos.

Tarea

Realizamos del libro del “Ciclo de Secundaria”, los ejercicios “El marciano en el desierto” y “Tito enciende las luces” de la sección “Autómatas, Comandos, Procedimientos y Repetición”.



Repetición Simple, comunicación y legibilidad

Nos vemos la próxima