



# Recorridos simples

Teoría 2.B.

# ¿Hicieron la tarea?

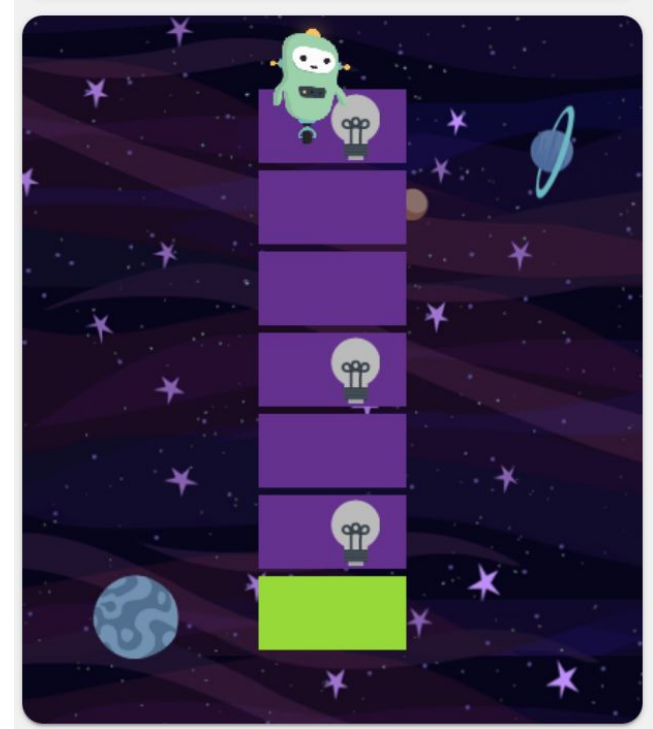
---

## Tito recargado

Ya vimos el hecho de que el escenario cambia, y hay que entenderlo bien para poder solucionar adecuadamente el problema.

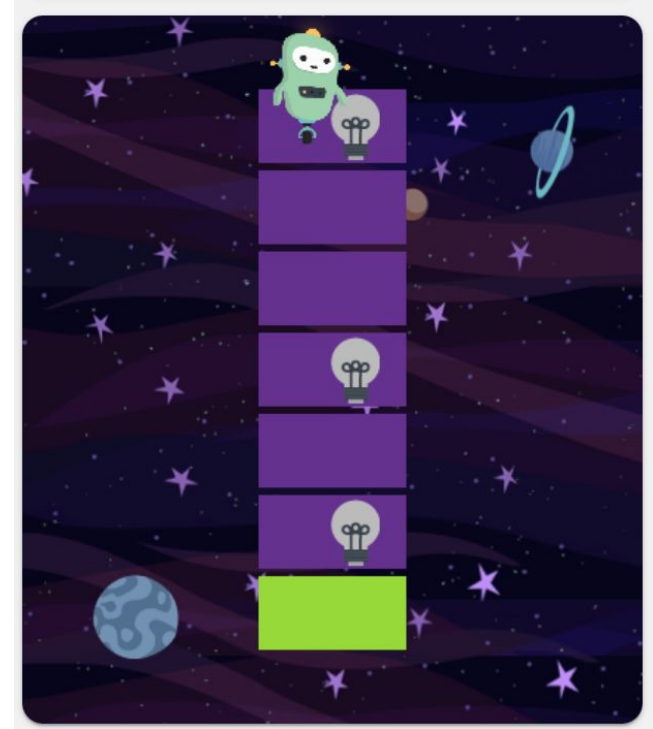
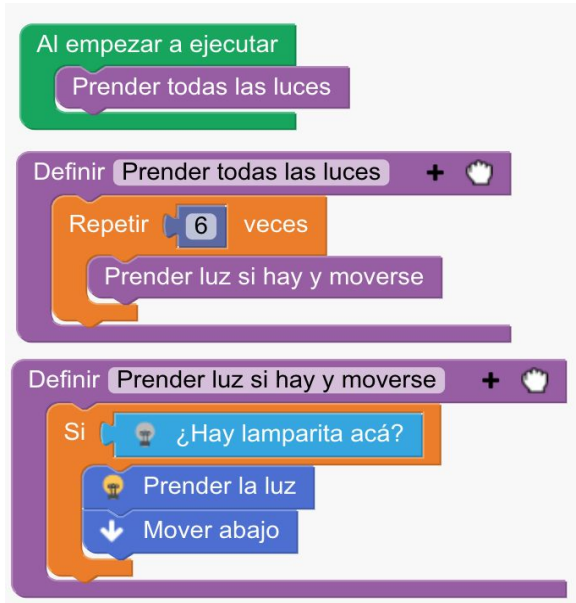
En este caso, las luces pueden aparecer en cualquiera de los lugares violetas, incluso donde arranca Tito.

Siempre hay 6 posiciones violetas en donde se debe entonces prender las luces.



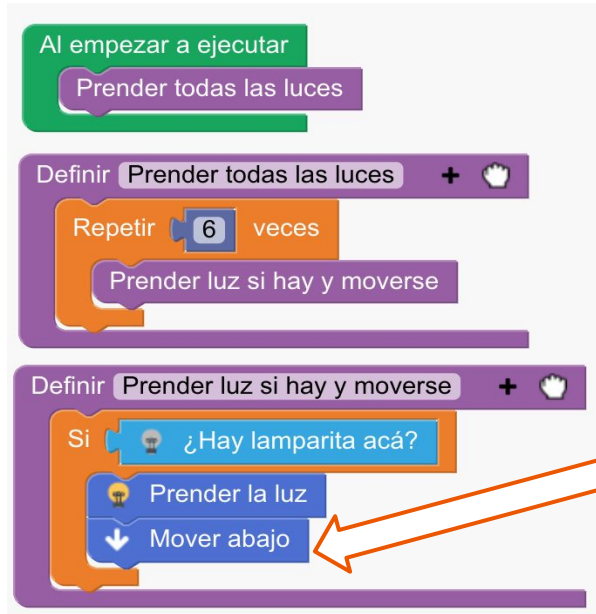
## Tito recargado

Analicemos una primera solución. ¿Qué problema tiene?

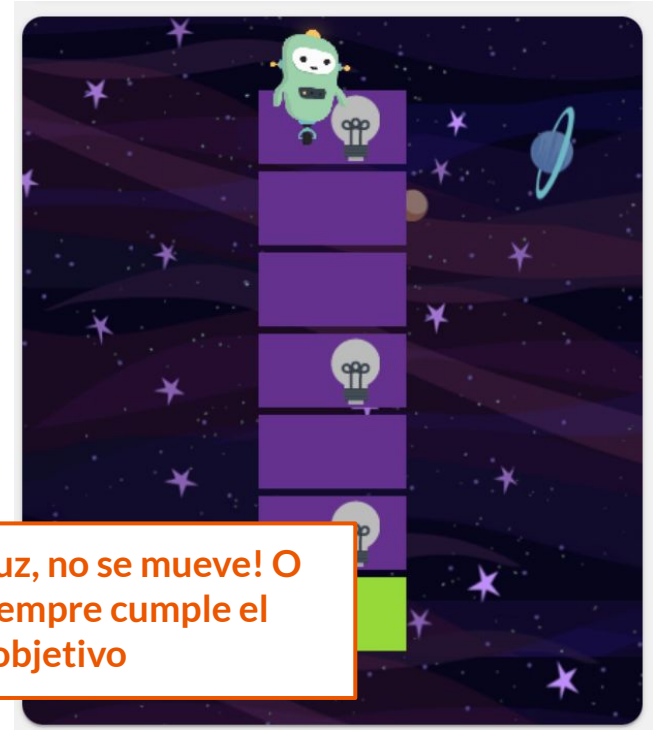


## Tito recargado

Analicemos una primera solución. ¿Qué problema tiene?

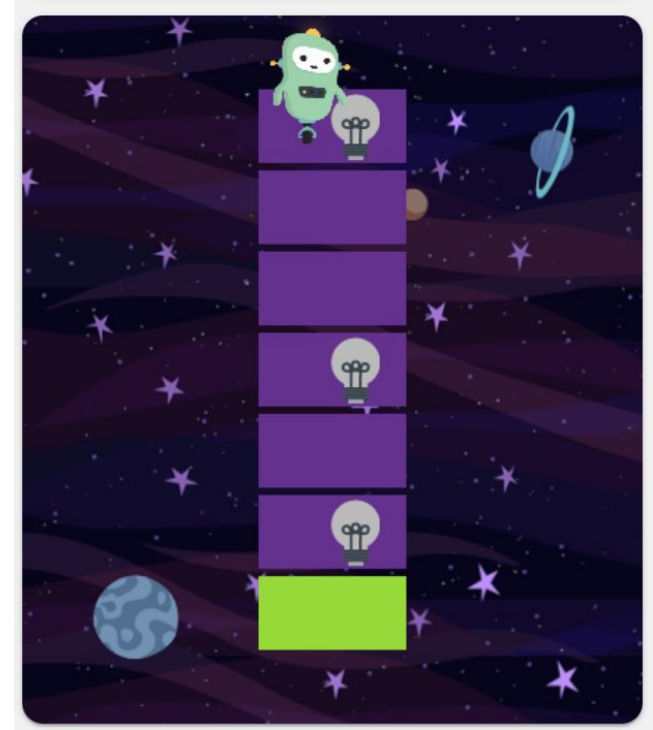
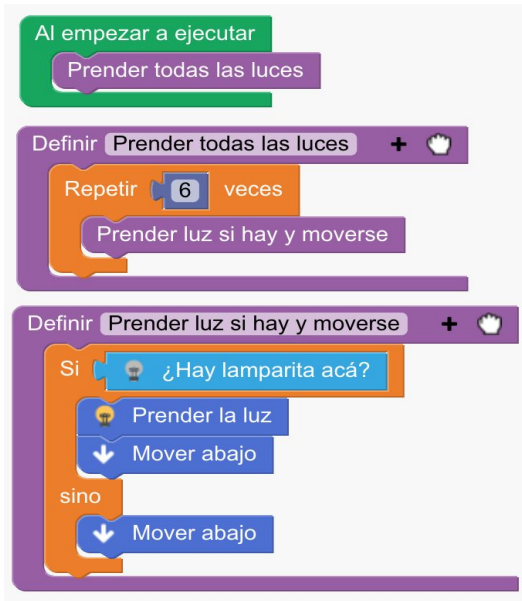


¡Si no hay luz, no se mueve! O sea, no siempre cumple el objetivo



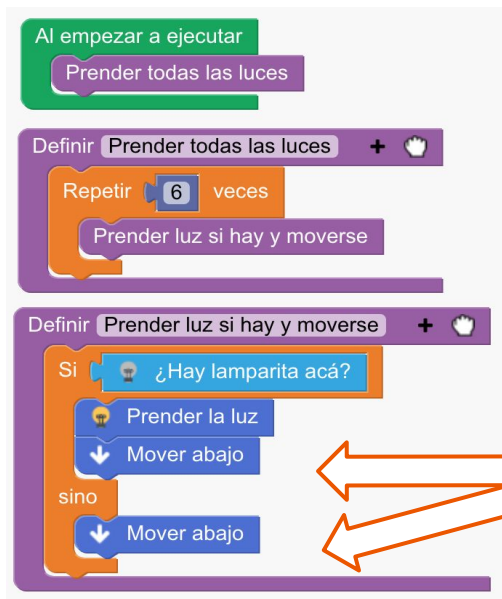
## Tito recargado

Analicemos una segunda solución.  
Nuevamente, no es adecuada ¿Por qué?



## Tito recargado

Analicemos una segunda solución.  
Nuevamente, no es adecuada ¿Por qué?

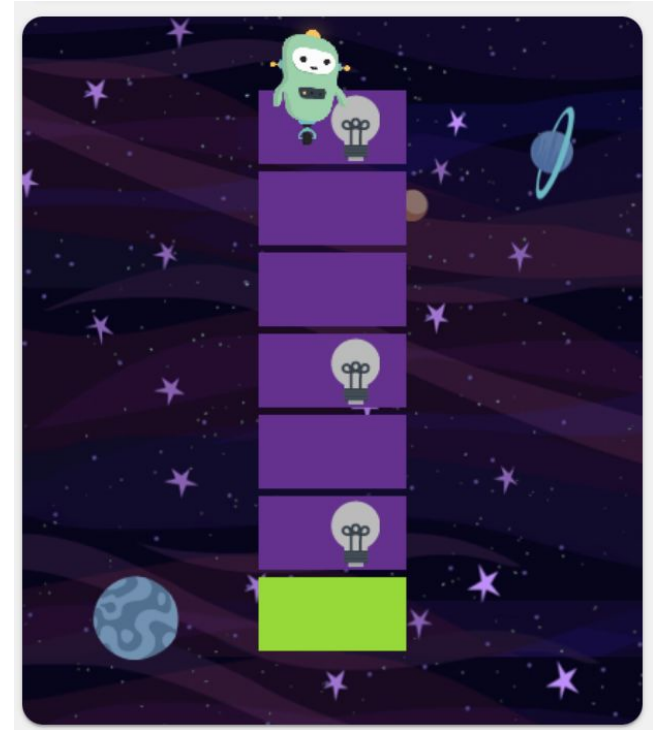
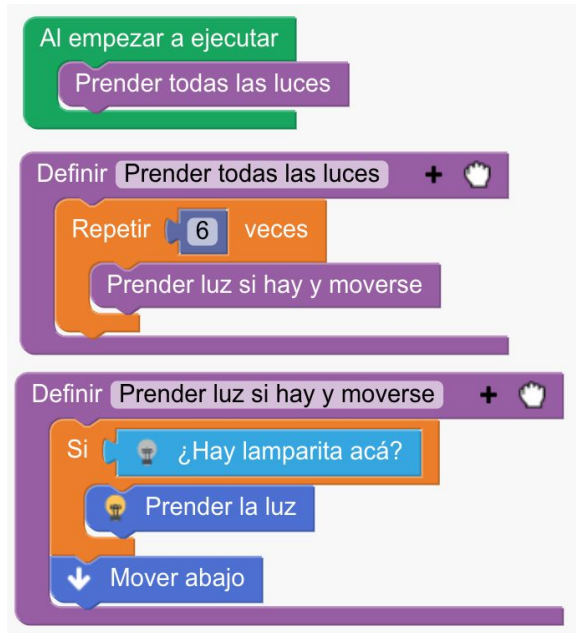


**¡Código duplicado!**  
Recordemos, si algo aparece en  
las dos ramas, es porque no  
depende de la alternativa. Debe  
ir fuera de la misma.



## Tito recargado

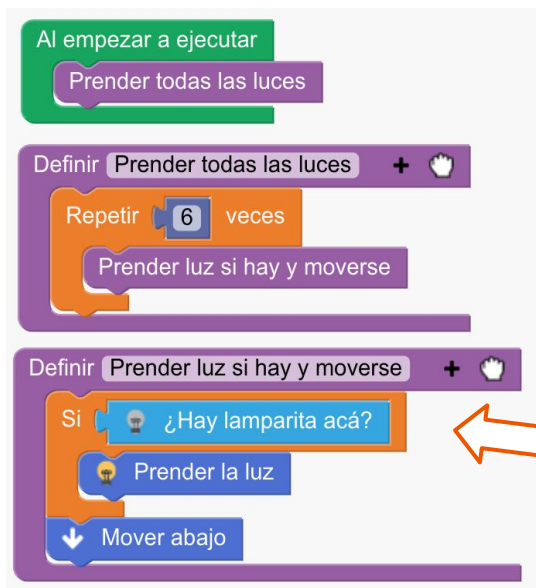
Analicemos una tercera solución. Esta es adecuada, pero...





## Tito recargado

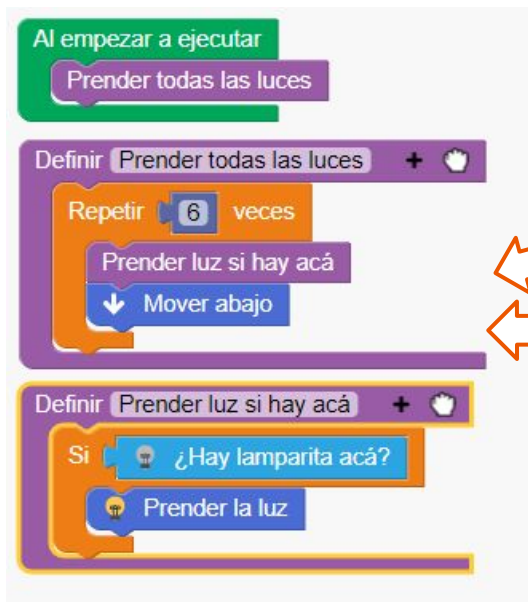
Analicemos una tercera solución. Esta es adecuada, pero...



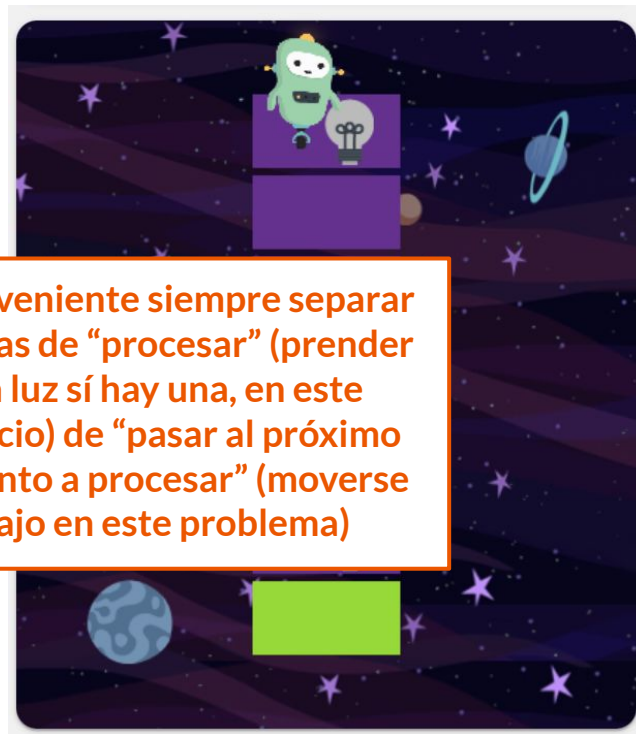
¿Cuántas tareas tiene este procedimiento?  
¿Es una buena división en subtareas?

## Tito recargado

Analicemos una tercera solución. Esta es adecuada, pero...



Es conveniente siempre separar las ideas de “procesar” (prender una luz sí hay una, en este ejercicio) de “pasar al próximo elemento a procesar” (moverse abajo en este problema)





## Recorridos simples

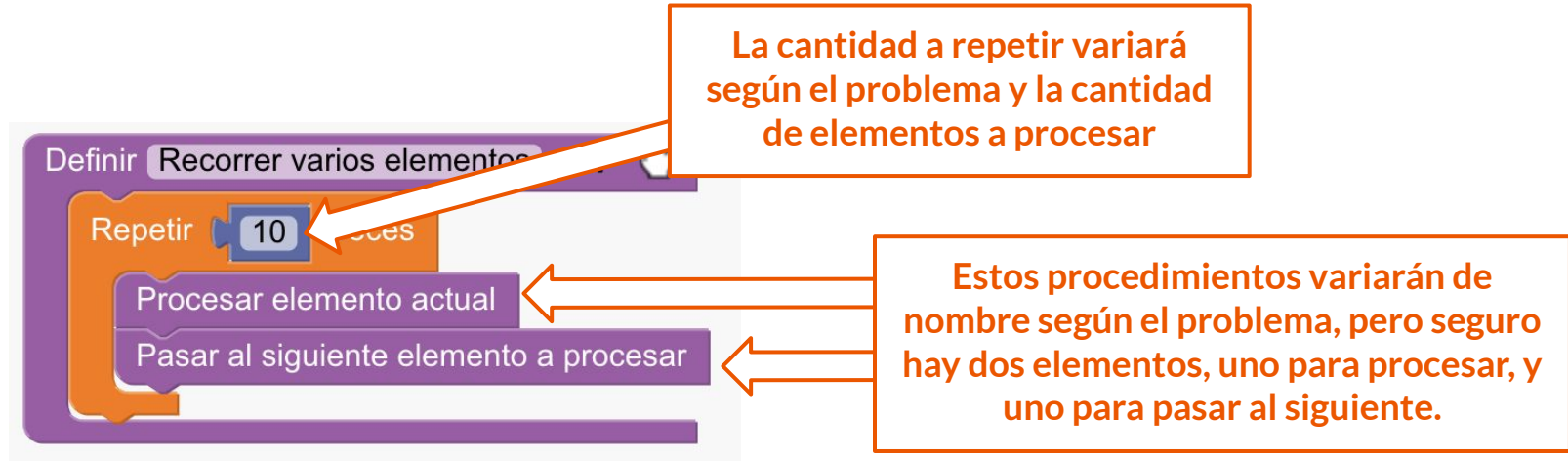
Son muchos los ejercicios que requieren “recorrer” varios elementos del escenario, realizando alguna acción en cada uno de ellos.

Decimos que un **recorrido simple** consiste en una estrategia que implica repetir una cierta cantidad de veces dos tareas:

- **Procesar elemento actual** (Hacer algo para cada elemento que se debe recorrer, lo cual puede ser condicional, por ej. prender una luz, sólo si hay una).
- **Pasar al siguiente elemento a procesar** (Pasar del elemento que se acaba de procesar al siguiente)

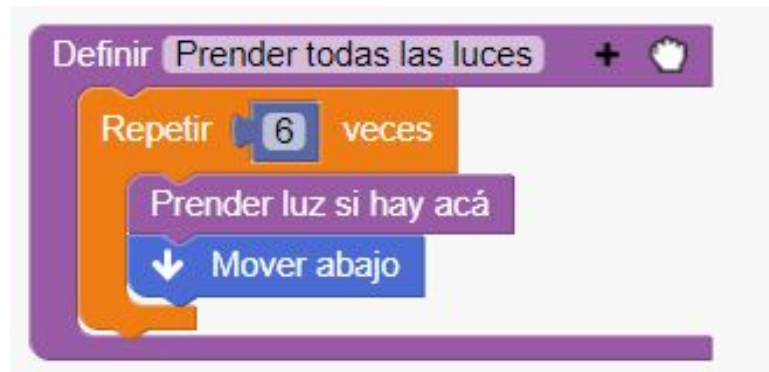
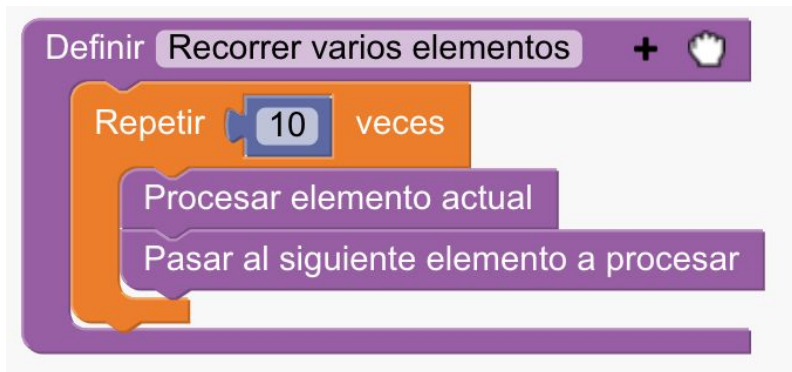
## Recorridos simples

Es decir, se contaría con un procedimiento con la siguiente estructura:



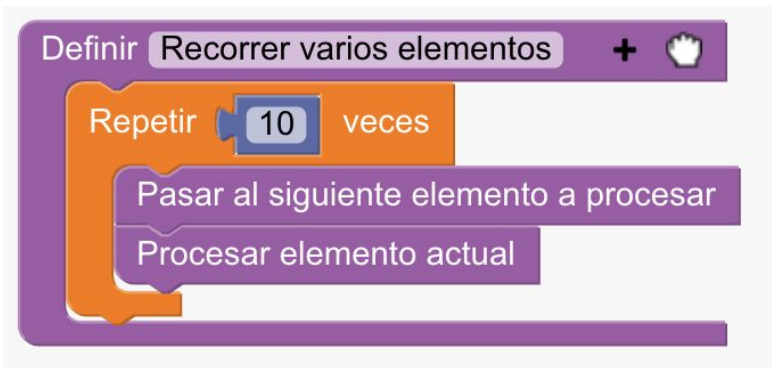
## Recorridos simples

Notar como el procedimiento “Prender todas las luces” del ejercicio de “Tito recargado” es un recorrido simple, pues cumple con el esquema propuesto.



## Recorridos simples

También puede darse que primero haya que moverse, y luego procesar (por ejemplo, si se tuviera el caso de que en la ubicación inicial no hubiera que procesar, pero sí en la última)



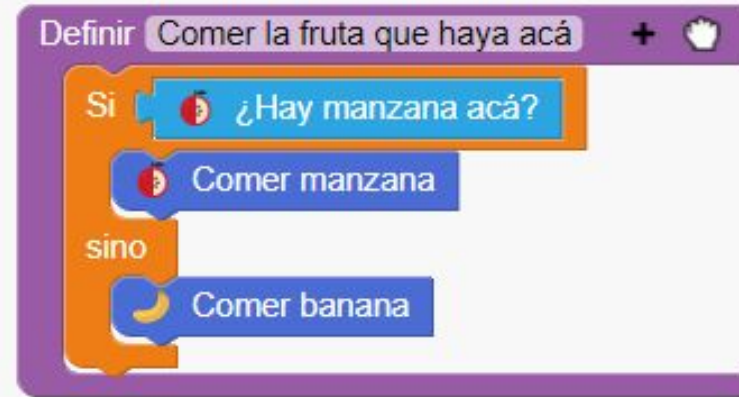
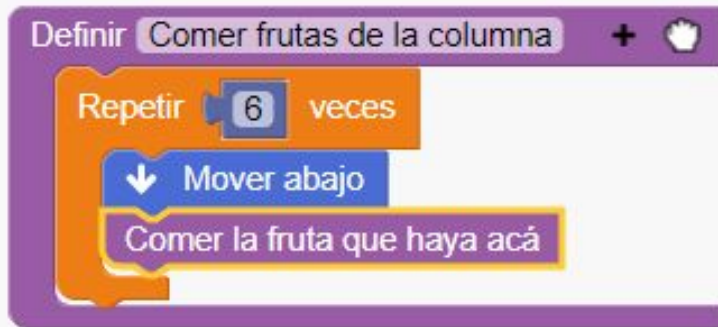
# A Trabajar

Realizamos del libro del “Ciclo de Secundaria”, el ejercicio “El festín frutal” de la sección “Alternativa Condicional”.



## Recorridos simples

En el ejercicio de “El festín frutal” para comer las frutas de una columna, hay que hacer un procedimiento con recorrido simple donde el “procesamiento” está en segundo lugar.

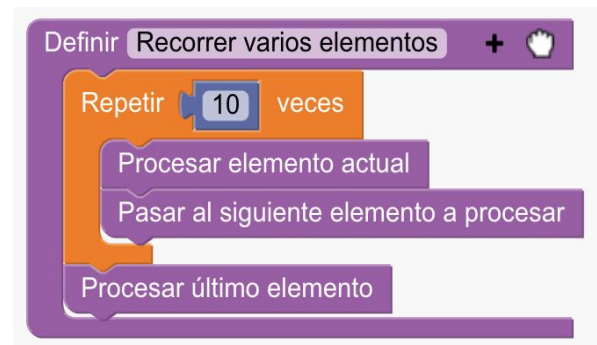
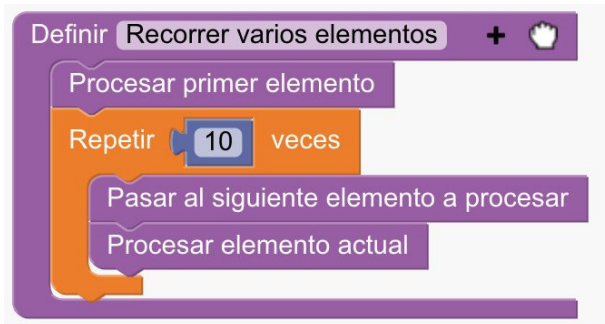




## Recorridos simples

También podemos contar con recorridos que requieran contemplar **casos de borde**, ya sea al principio (procesando de forma separa el primer elemento) o al final (procesando el último de forma separada).

El procesamiento de ese elemento puede ser diferente al del resto de los elementos (por ej. tener que hacer una acción muy distinta) o puede ser igual a procesar cualquier otro elemento. Dependerá del problema.



# A Trabajar

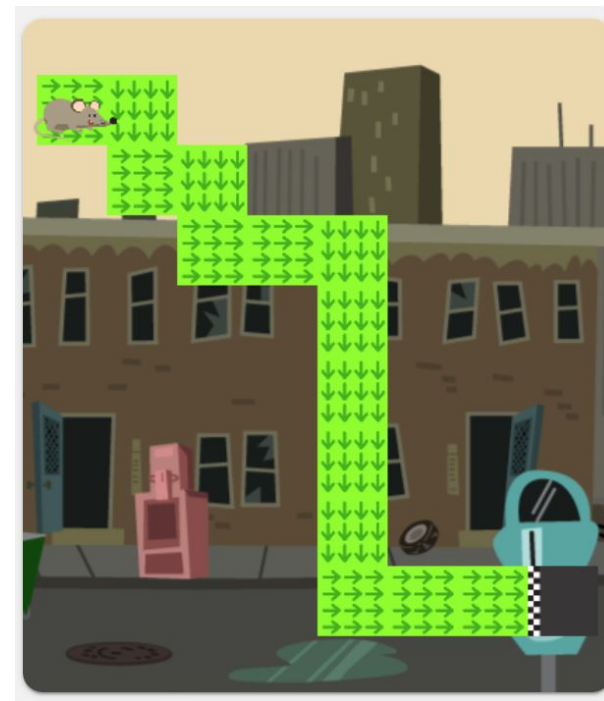
Realizamos del libro del “Ciclo de Secundaria”, el ejercicio “Laberinto largo” de la sección “Alternativa Condicional”.



## Laberinto largo: ¿Qué aprendimos?

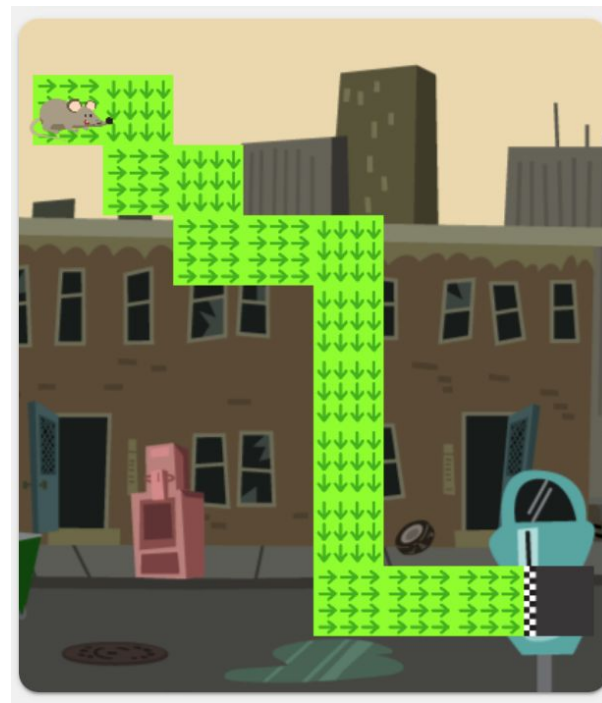
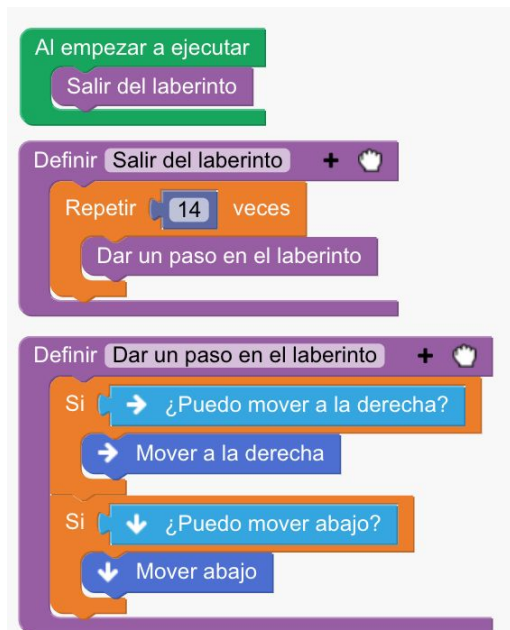
También es un recorrido, pero acá no hay que hacer nada en cada ubicación, por lo que la única parte importante es avanzar.

No hay casos de borde, porque no se procesa nada.



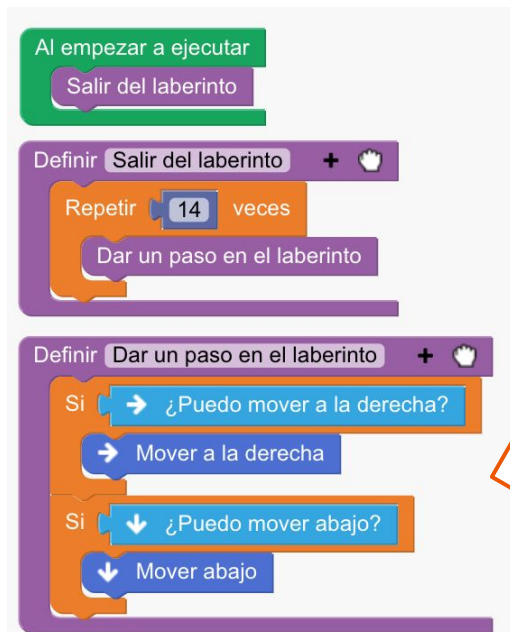
## Laberinto largo: ¿Qué aprendimos?

Aproximación 1:

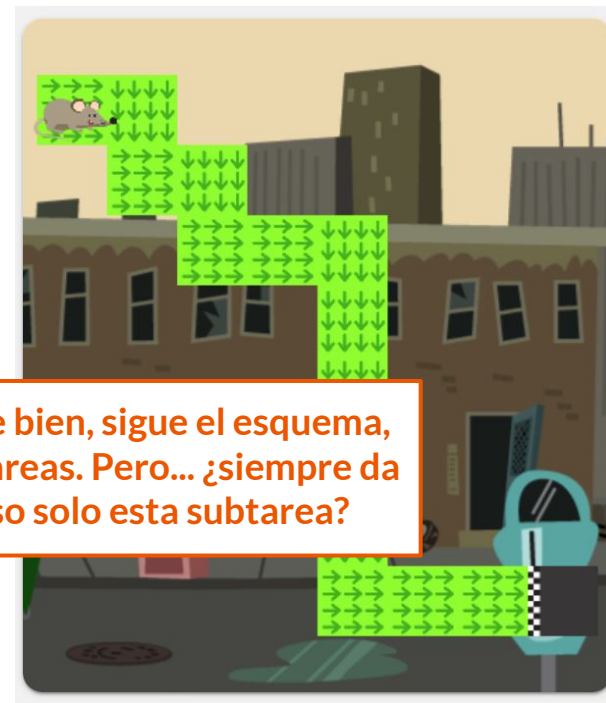


## Laberinto largo: ¿Qué aprendimos?

Aproximación 1:

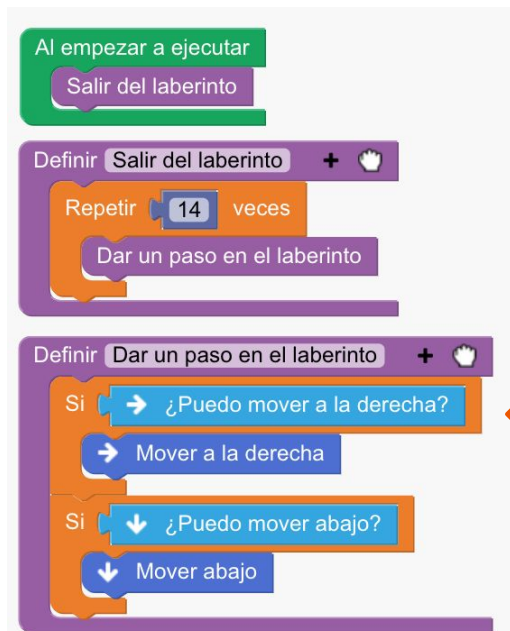


Bastante bien, sigue el esquema, usa subtareas. Pero... ¿siempre da un paso solo esta subtask?

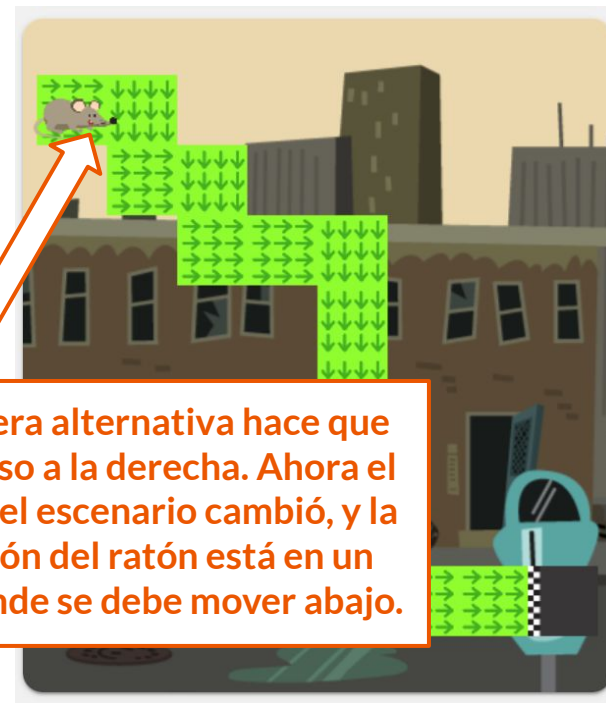


## Laberinto largo: ¿Qué aprendimos?

Aproximación 1:

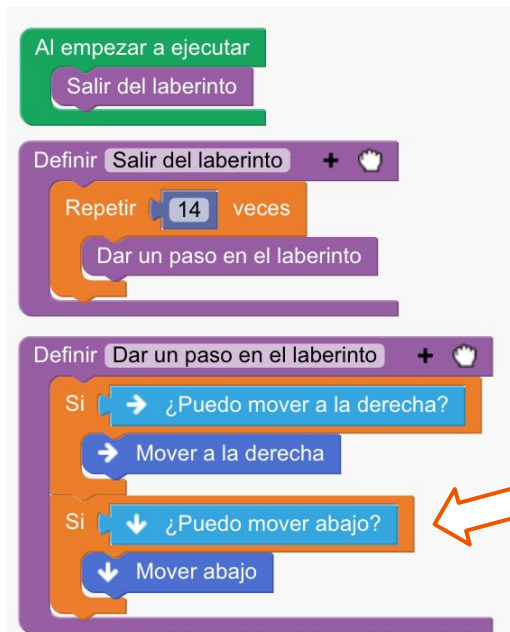


La primera alternativa hace que de un paso a la derecha. Ahora el estado del escenario cambió, y la ubicación del ratón está en un lugar donde se debe mover abajo.

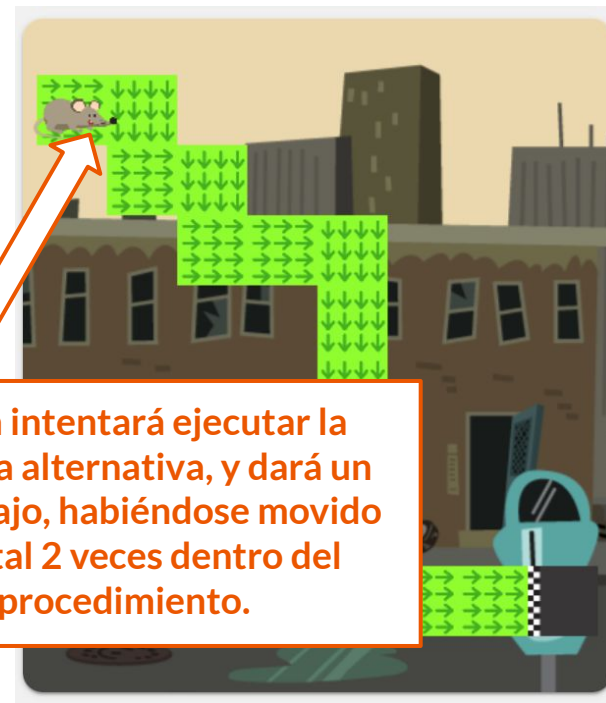


## Laberinto largo: ¿Qué aprendimos?

Aproximación 1:

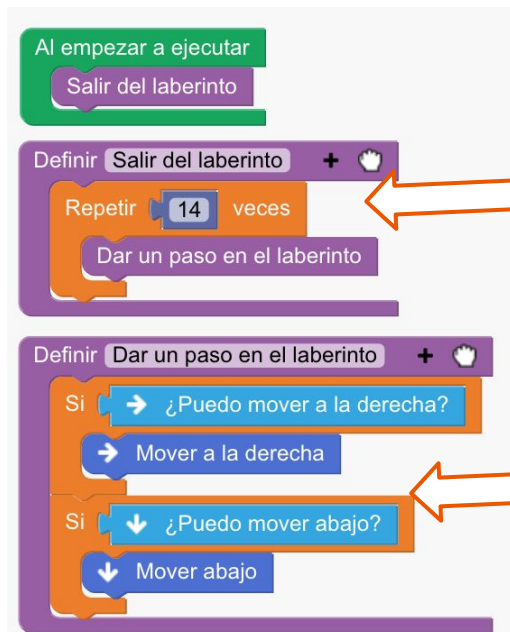


Ahora intentará ejecutar la segunda alternativa, y dará un paso abajo, habiéndose movido en total 2 veces dentro del procedimiento.



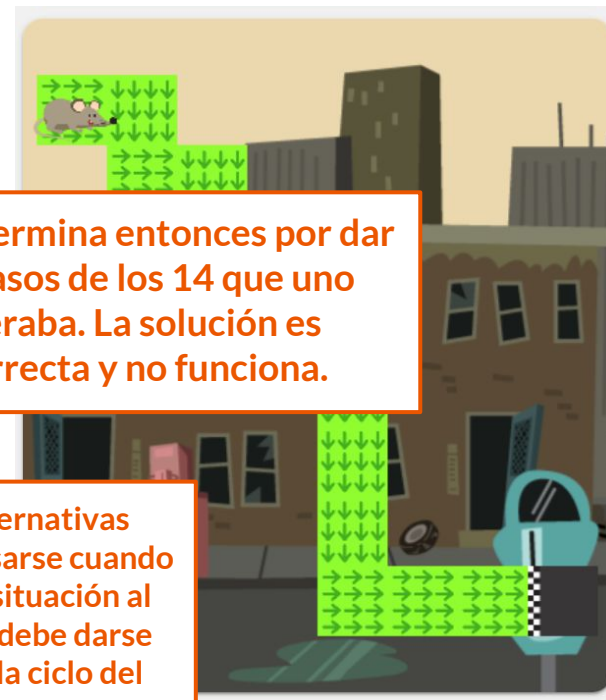
## Laberinto largo: ¿Qué aprendimos?

Aproximación 1:



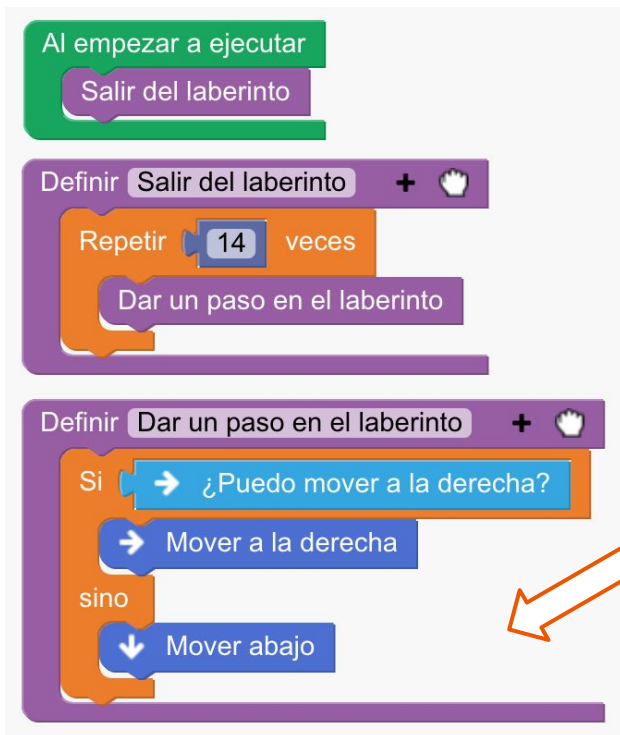
Al final termina entonces por dar más pasos de los 14 que uno esperaba. La solución es incorrecta y no funciona.

Recordemos que las alternativas secuenciales solo deben usarse cuando pueda darse más de una situación al mismo tiempo y acá solo debe darse una (un solo paso) en cada ciclo del repetir

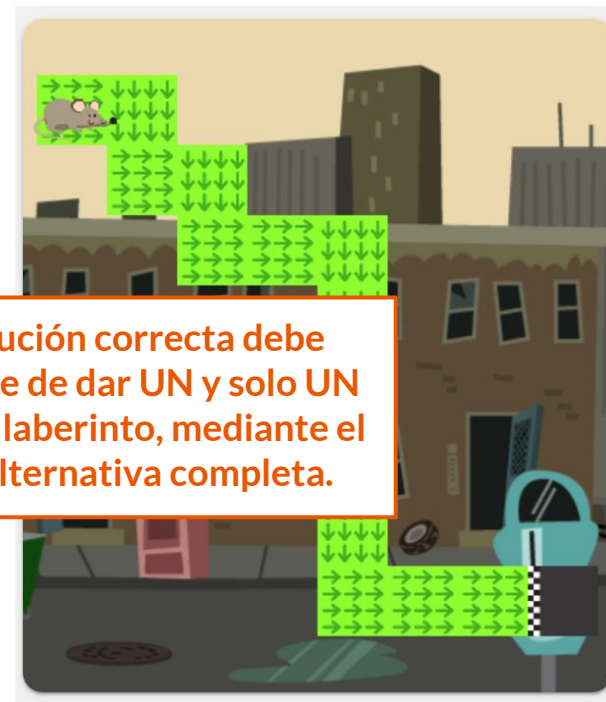




## Laberinto largo: ¿Qué aprendimos?



Una solución correcta debe asegurarse de dar UN y solo UN paso en el laberinto, mediante el uso de alternativa completa.

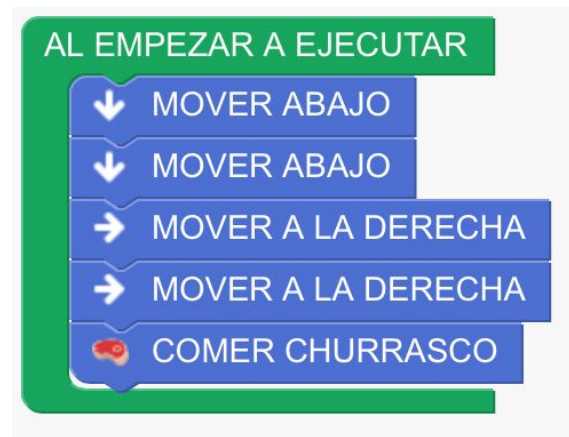




# Momento de dudas o consultas

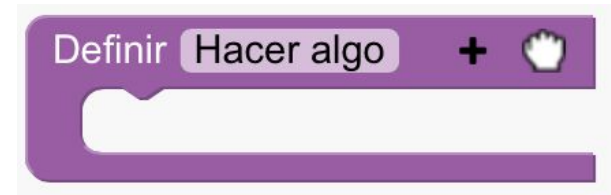
## Repaso

- 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**.



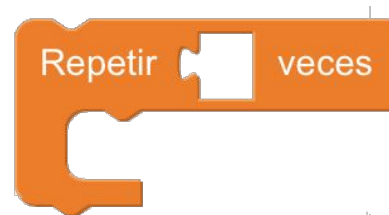
## Repaso

- 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.
- **No hay que anidar repeticiones (ni ninguna otra estructura de control).**



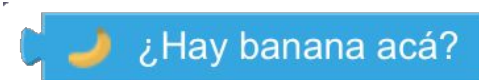
## Repaso

- La **alternativa condicional** es una herramienta del lenguaje que **permite cambiar el flujo** del programa (**estructura de control de flujo**).
- Permite **elegir** entre posibles **ramas**, según una **condición** en el estado del programa.
- 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 de valor de verdad** para indicar cuándo se elige un camino y cuando otro..
- **No hay que anidar alternativas (ni ninguna otra estructura de control).**



## Repaso

- Un **comando** es la descripción de una **acción**.
- Una **expresión** en la descripción de un **dato** (un **valor, información**)
- Puede ser **numéricas**, describen una cantidad, y las usamos en la repetición simple.
- Pueden ser de **valor de verdad**, se responden con verdadero o falso y las usamos en la alternativa.
- Los **sensores** son la herramienta mediante la cual obtenemos valores de verdad basados en el estado del programa (escenario)





## 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.
- No anidar estructuras. Usamos procedimientos para dividir el problema en partes pequeñas y darles nombres adecuados.



# Tarea

Realizamos la práctica en papel “2.B. Recorridos simples”.

---



# Recorridos simples

Nos vemos la próxima