



Alternativas Condicionales

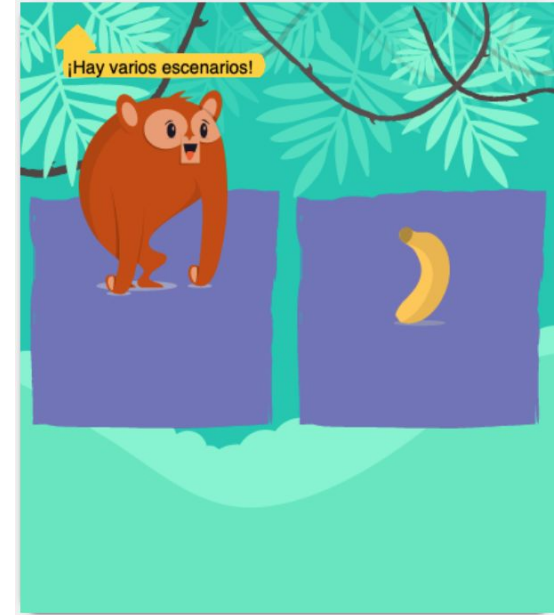
Teoría 2.A

A trabajar

Realizamos del libro del “Ciclo de Secundaria”, los ejercicios “El mono y las bananas” y “La elección del mono” de la sección “Alternativa Condicional”.

El mono y las bananas: ¿Qué aprendimos?

¿Qué diferencia hay en este problema con respecto a los que veníamos trabajando hasta ahora?

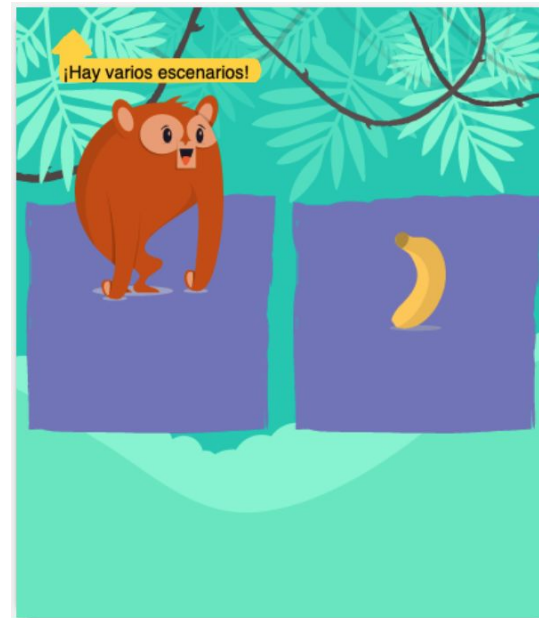


El mono y las bananas: ¿Qué aprendimos?

¿Qué diferencia hay en este problema con respecto a los que veníamos trabajando hasta ahora?

¡¡El escenario cambia!!

No hay un único escenario posible, sino que el escenario varía entre una y otra ejecución.

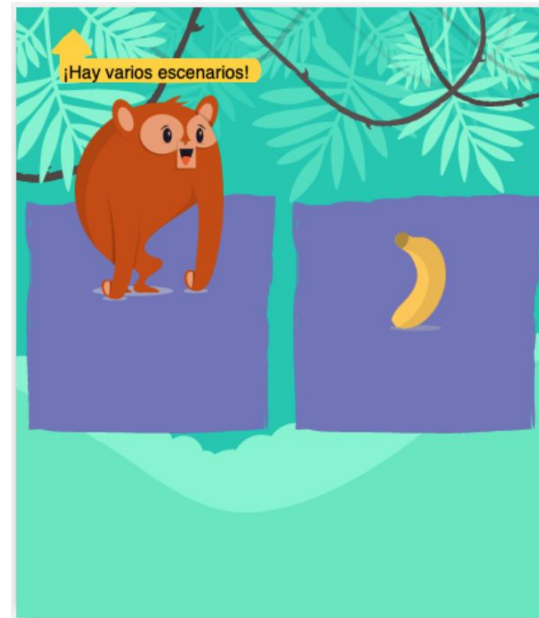


El mono y las bananas: ¿Qué aprendimos?

Nuestra solución debe funcionar en cualquier escenario posible. ¿Cómo hacer?

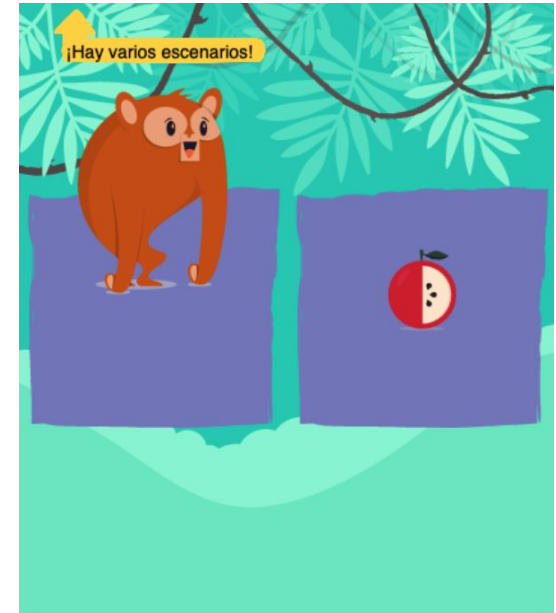
En primer lugar, hay que entender cómo cambia el escenario (no todo en el escenario es cambiante).

Siempre hay dos lugares, el mono siempre empieza en el mismo, y sí hay una banana, esta está en el lugar a la derecha. Lo único que cambia es el hecho de que puede haber una banana, o no haberla.



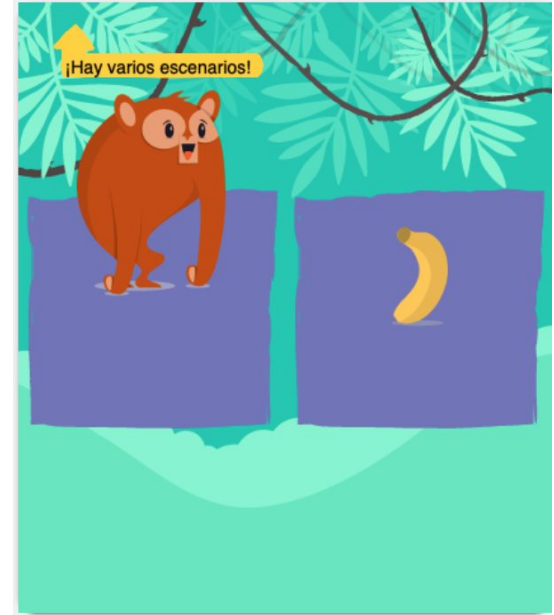
La elección del mono: ¿Qué aprendimos?

¿Y “en la elección del mono”? También el tablero cambia, ahora pudiendo tener bananas, o manzanas (pero no hay casos en donde no haya ninguna fruta).



El mono y las bananas: ¿Qué aprendimos?

Las herramientas actuales no son suficientes. Se necesita una nueva herramienta para solucionar este problema



Condición y sensores

- Una **condición** es algo que puede o no cumplirse. Podemos pensarlo como una pregunta que puede responderse únicamente con **sí** o con **no**.
- Es decir, una condición es algo que tiene un **valor de verdad**, que puede ser **verdadero** o **falso** (quienes vieron lógica ya saben de qué va. Para los demás, más adelante trabajaremos este tema).
- Un **sensor** es una primitiva que provee una actividad para hacer preguntas sobre el escenario.
- Es una forma de **expresión** (notar la forma del bloque y dónde encaja. Es más similar al bloque numérico que usamos en la repetición que a los otros bloques, y eso es porque es una expresión, describe información, datos).
- Nos permite saber cuál es el **estado actual** del escenario, y actuar en consecuencia.
- En PilasBloques, por ahora, todos los **sensores** son **condiciones**.





Una cuestión de estado

Una condición se evalúa en el momento que la computadora llega a ese punto del programa. Por eso hablamos del “**estado actual**” y no del “**estado inicial**” del escenario.

Recordemos que un programa en PilasBloques consiste en una **transformación de estados** (el estado es la situación dada por el escenario, la ubicación de los personajes y elementos). El **estado inicial** es aquel dado por el escenario antes de arrancar. Cuando se ejecuta un comando, el estado cambia, y pasa a ser otro. Cuando se terminó de ejecutar el programa, se llega al **estado final**.

Mientras el programa se ejecuta, se puede hablar del **estado actual**, que es aquel en el que el escenario se encuentra en un determinado punto del tiempo, es decir, luego de haber ejecutado una parte particular del código, pero no otra.

Así, la pregunta **¿Hay banana acá?** es una pregunta que dependerá del estado actual (de la ubicación del mono, y lo que haya junto a él). No es lo mismo hacer la pregunta con el mono en la ubicación de la izquierda que con la ubicación de la derecha, por lo que es importante moverse previo a hacer la pregunta.

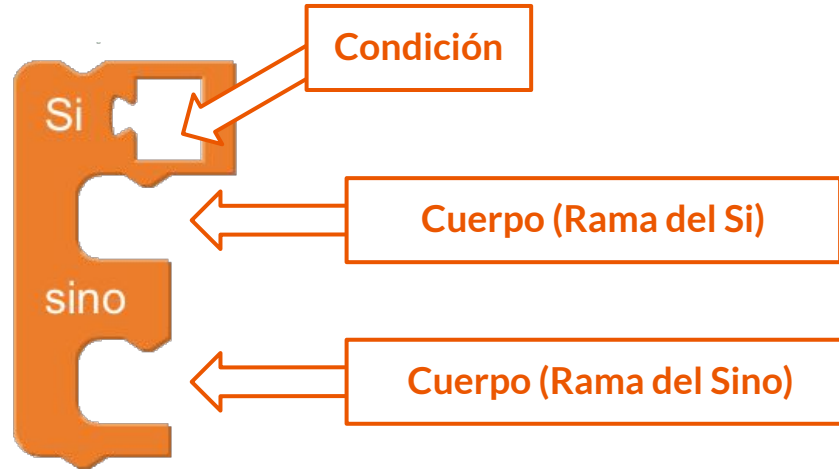
Más aún, si preguntamos dos veces lo mismo, en momento diferentes, la respuesta podría ser distinta (por ejemplo, si volvemos a preguntar luego de haber comido la banana).



Alternativa Condicional

- La **alternativa condicional** es una nueva forma de organizar el código.
- Se suma entonces a la repetición como una **estructura de control del flujo**.
- Permite **elegir** entre diversas acciones a realizar, según alguna condición que puede o no cumplirse.
- El bloque es un **comando compuesto**, por lo que tiene un cuerpo.
- De hecho es un comando compuesto que tiene dos cuerpos, a los que llamamos **ramas**.
- Espera también una **expresión**, pero en este caso no es una expresión numérica, sino una **condición**.
- En PilasBloques las condiciones van a surgir en principio de **sensores**, pero podrían haber otros tipos de condiciones.

Alternativa Condicional



Alternativa Condicional

- Una alternativa condicional siempre tiene una condición.
- En su forma completa, la alternativa condicional permite **elegir** entre dos **cuerpos**, a los que llamamos **ramas**, la **rama del si** y la **rama del sino**.
- Los comandos de la **rama del sí** se ejecutarán solamente **sí la condición es verdadera**, y en ningún otro caso.
- Los comandos de la **rama del sino** se ejecutarán solo **sí la condición es falsa**, y en ningún otro caso.



Alternativa Condicional completa

- Una alternativa condicional siempre tiene una condición.
- En su forma completa, la alternativa condicional permite **elegir** entre dos **cuerpos**, a los que llamamos **ramas**, la **rama del sí** y la **rama del sino**.
- Los comandos de la **rama del sí** se ejecutarán solamente **sí la condición es verdadera**, y en ningún otro caso.
- Los comandos de la **rama del sino** se ejecutarán solo **sí la condición es falsa**, y en ningún otro caso.

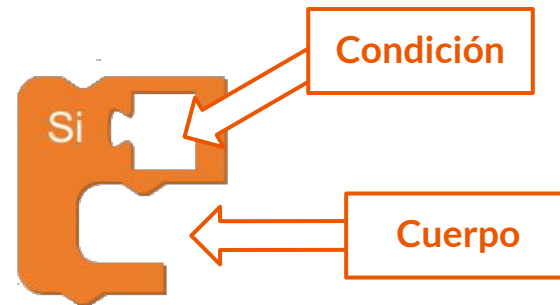


Sí efectivamente hay una manzana, entonces se ejecutará este bloque.

Sí NO hay una manzana, entonces se ejecutará este bloque.

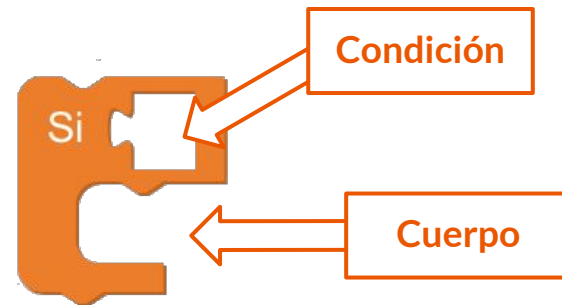
Alternativa Condicional simplificada

- Es muy común el caso en el que queremos que hacer algo sí se cumple una condición, pero en caso de que no se cumpla, no queremos hacer nada.
- Dejar una rama de la alternativa vacía es una mala práctica. No queremos eso. ¿Cómo hacer entonces?



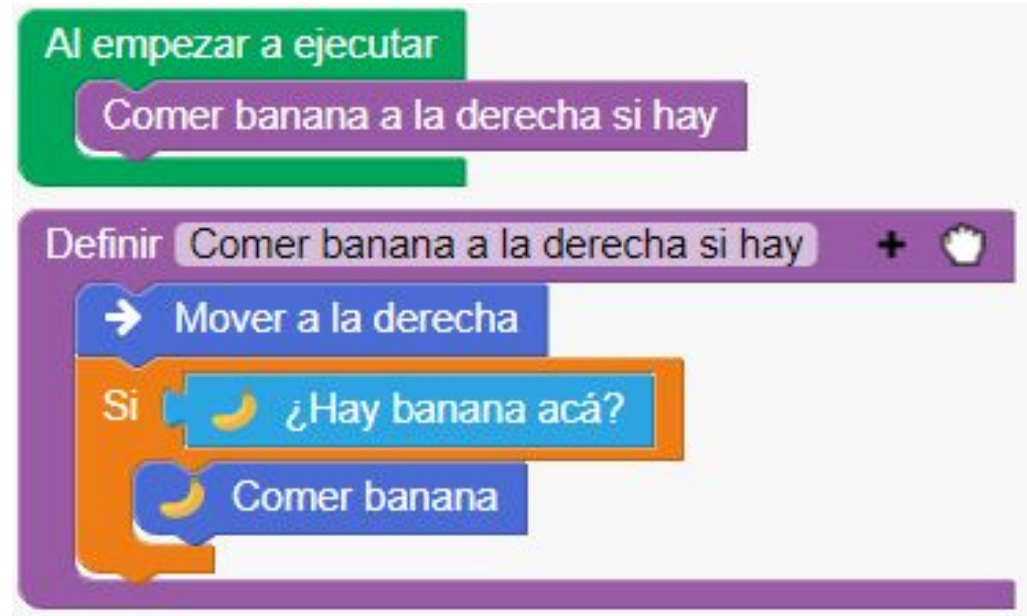
Alternativa Condicional simplificada

- La alternativa condicional tiene una versión simplificada, que carece de la rama del sino.
- En esta versión, si la condición se cumple, se ejecuta el cuerpo (rama del sí) y sino, no se hace nada.
- Notar que no es otra herramienta, sino un bloque distinto, más simple, de la misma herramienta, para contemplar un caso especial.



¡Es un comando!

- La alternativa condicional es un comando, y por tanto pueden haber otros comandos antes y otros después del mismo.
- Luego de ejecutar la rama que corresponda, el programa vuelve al flujo normal, ejecutando el comando inmediatamente debajo de la alternativa condicional.





Lo de adentro, adentro, lo de afuera, afuera

Un error común cuando arrancamos a trabajar con la alternativa condicional es el hecho de duplicar comandos que deberían ir fuera de la alternativa, dentro, en cada rama.

Veamos un ejemplo:

Lo de adentro, adentro, lo de afuera, afuera

Notar cómo “Mover a la derecha” aparece en ambas ramas.

Esto suele surgir de pensar el problema en español como “si hay una manzana, como la manzana y me muevo, sino, solamente me muevo”.

Esa es una clara indicación de que esa acción, no tiene nada que ver con la alternativa. El movimiento lo debo hacer siempre, haya o no manzana.



Lo de adentro, adentro, lo de afuera, afuera

La forma adecuada de solucionar ese mismo problema, sería usando una alternativa condicional simplificada, y dejando el movimiento como un comando fuera de la alternativa.

Podemos entenderlo como “sin importar si comí o no la manzana, me muevo a la derecha”.

Esta versión y la anterior son equivalentes en términos de ejecución (ambas resolverán el problema) pero consideraremos que la versión donde se duplica código no es adecuada.

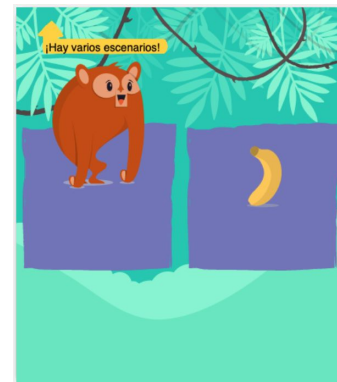
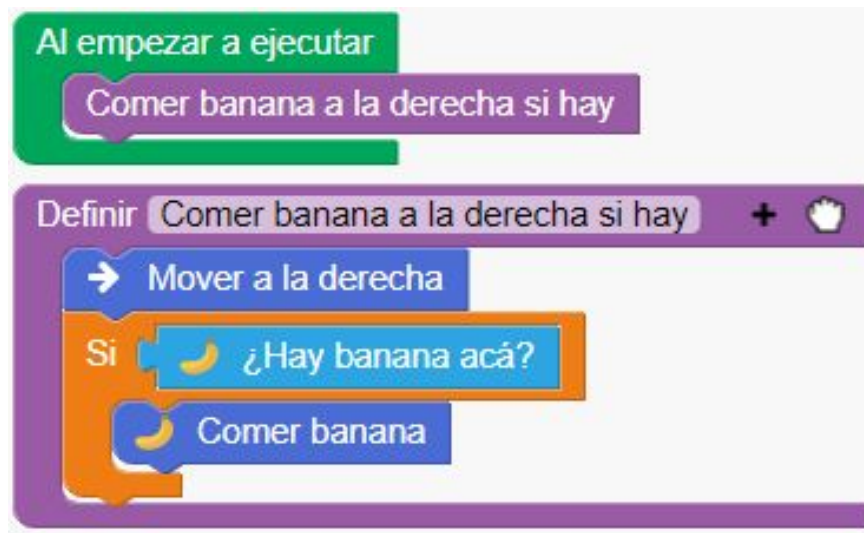




Momento de dudas o consultas

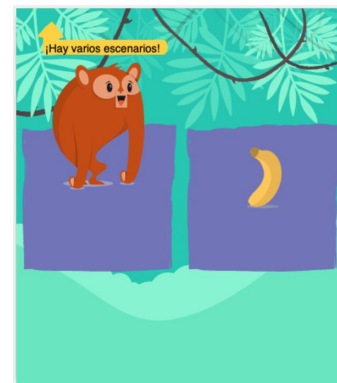
El mono y las bananas: ¿Qué aprendimos?

Una solución adecuada.



El mono y las bananas: ¿Qué aprendimos?

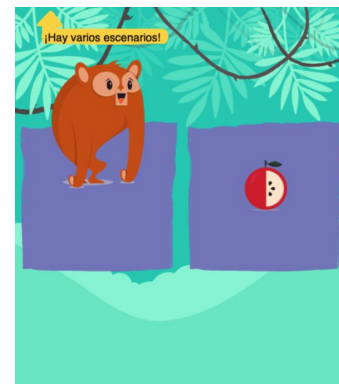
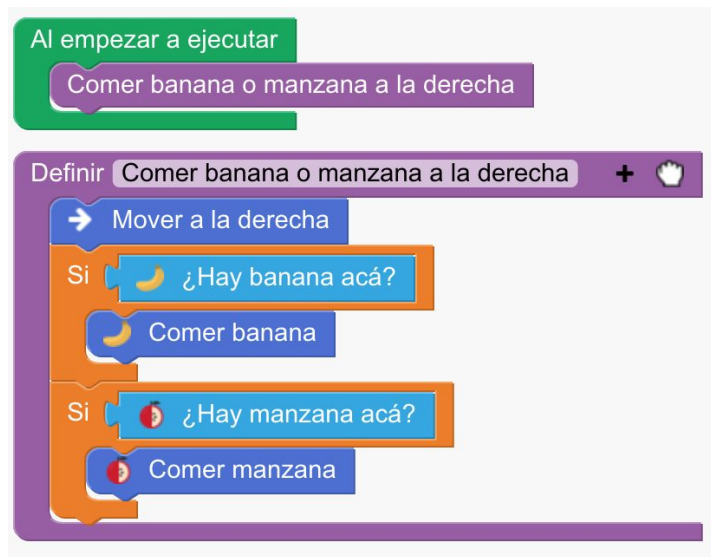
Podemos mejorarla, con una subtarea adicional.



La elección del mono: ¿Qué aprendimos?

Una primera aproximación.

Meh. Funciona, pero no descubrieron la alternativa condicional en su versión completa.

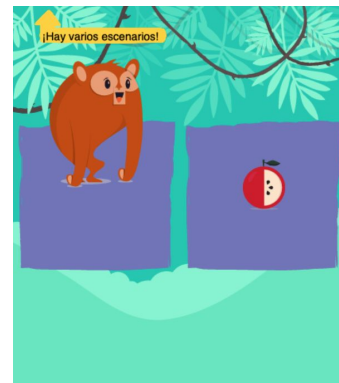
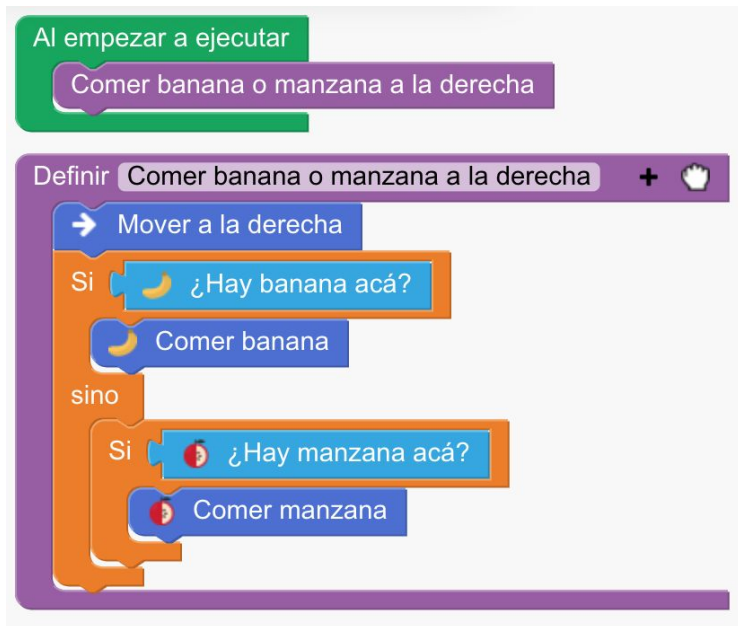


La elección del mono: ¿Qué aprendimos?

Una segunda aproximación.

Horrible. Recordemos que los bloques naranja no pueden estar anidados, nunca!

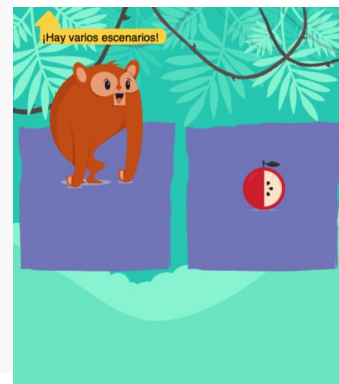
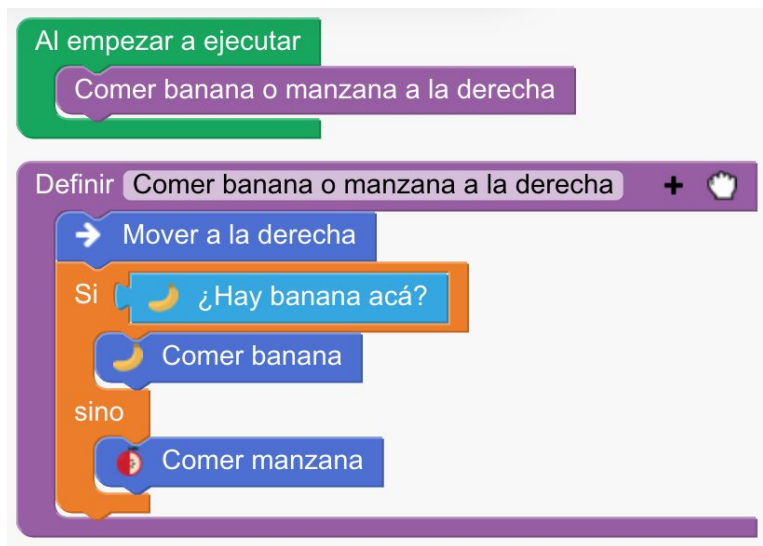
¿Puede NO haber banana y que no haya tampoco una manzana? ¿Qué sentido tiene la segunda pregunta?



La elección del mono: ¿Qué aprendimos?

Una tercera aproximación.

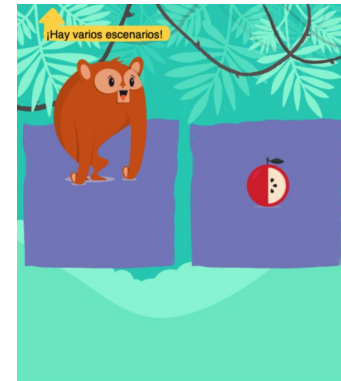
Excelente. Se descubrió el bloque en su versión completa y se usa adecuadamente, teniendo en cuenta la restricción del escenario que describe el enunciado.



La elección del mono: ¿Qué aprendimos?

Una cuarta aproximación.

Genial, se hace una subtask para procesar (comer la banana o la manzana).



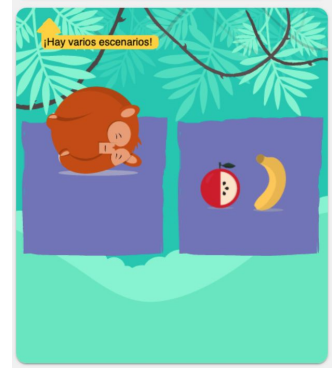
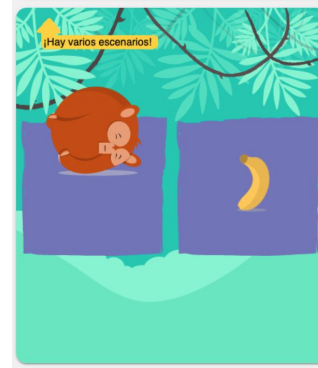
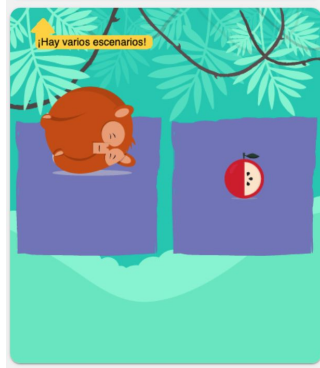
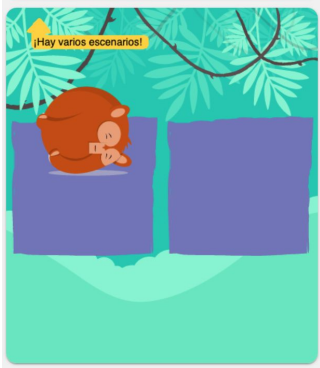


Nuevo problema

¿Qué pasaría si en el escenario pudieran haber tanto mazanas como bananas en la ubicación a la derecha, y queremos que el mono coma ámbas? ¿Y si pudiera no haber ninguna de las dos? ¿Cuáles son los posibles escenarios iniciales?

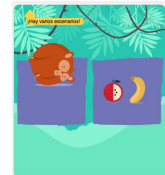
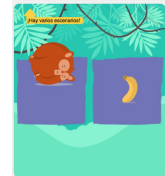
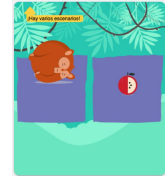
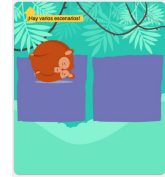
Nuevo problema

Primero siempre debemos pensar cuáles van a ser los escenarios posibles. En este caso hay 4, pero pueden ser más o menos dependiendo del problema.



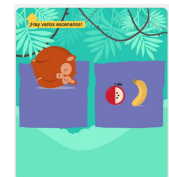
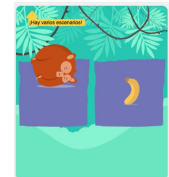
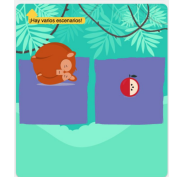
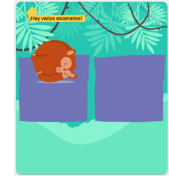
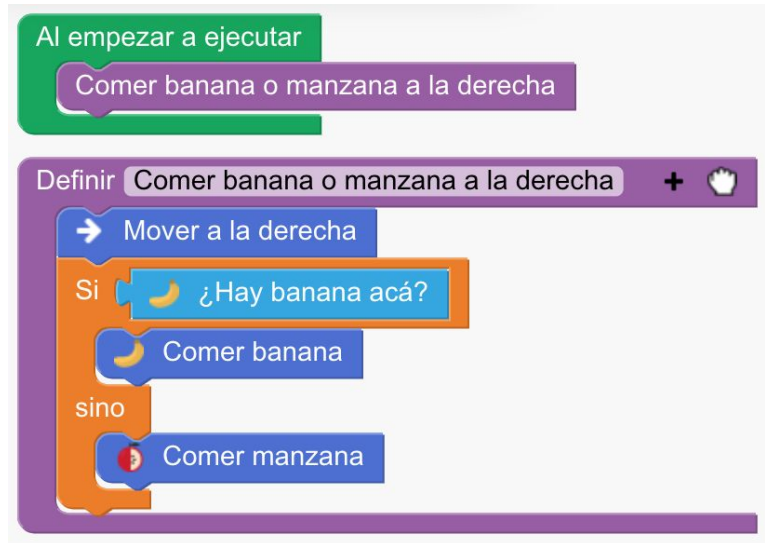
Nuevo problema

Veamos algunas soluciones posibles, y analicemos si funcionan en todos los escenarios o no.



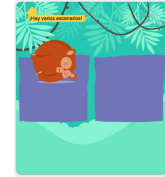
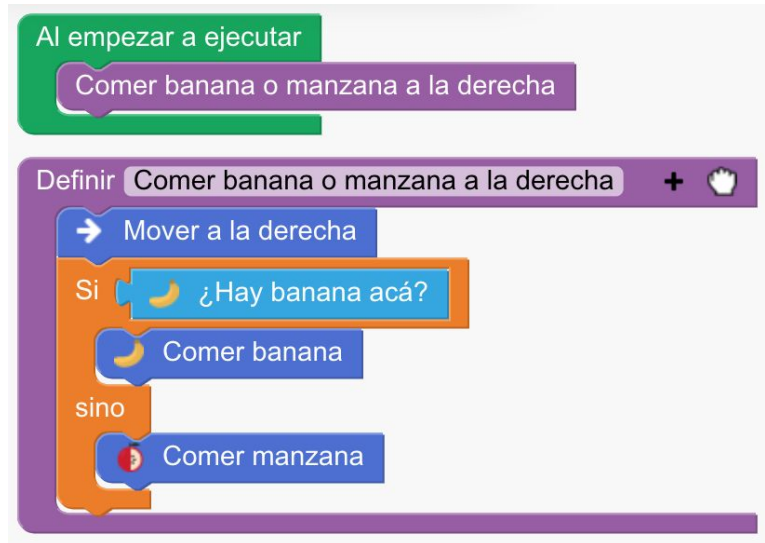
Nuevo problema

Veamos algunas soluciones posibles, y analicemos si funcionan en todos los escenarios o no.

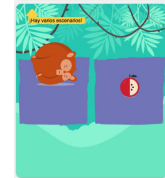


Nuevo problema

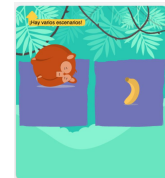
Veamos algunas soluciones posibles, y analicemos si funcionan en todos los escenarios o no.



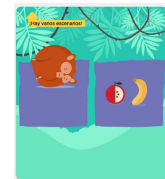
NO



SI



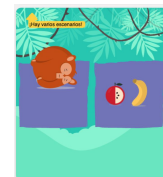
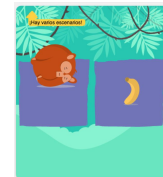
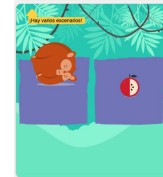
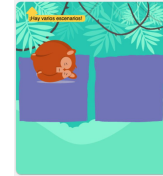
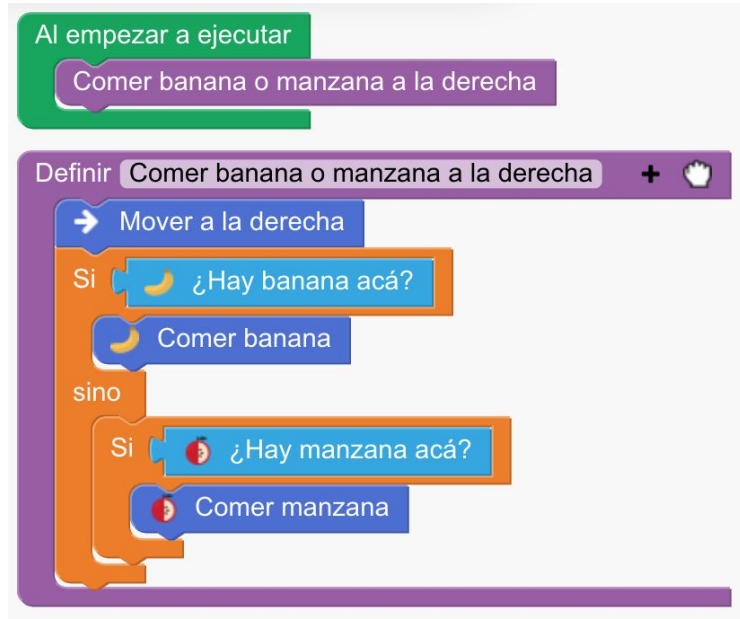
SI



NO

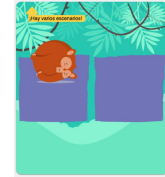
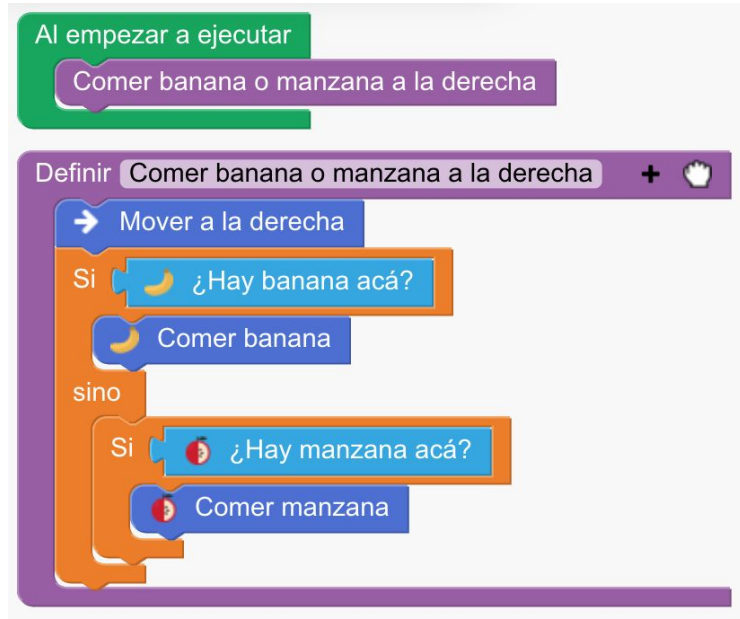
Nuevo problema

Veamos algunas soluciones posibles, y analicemos si funcionan en todos los escenarios o no.

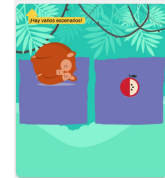


Nuevo problema

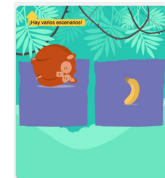
Veamos algunas soluciones posibles, y analicemos si funcionan en todos los escenarios o no.



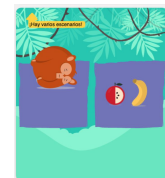
SI



SI



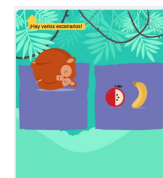
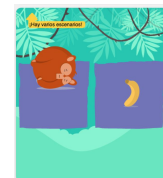
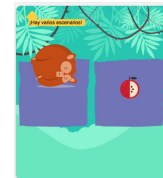
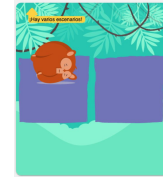
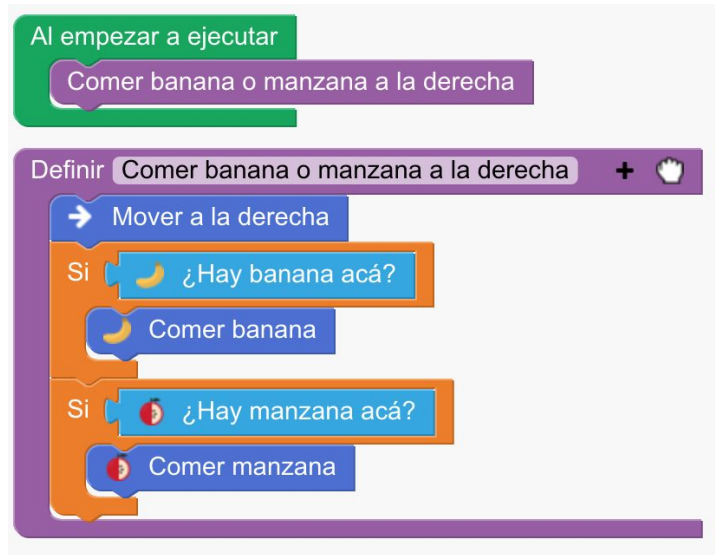
SI



NO

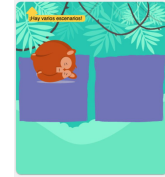
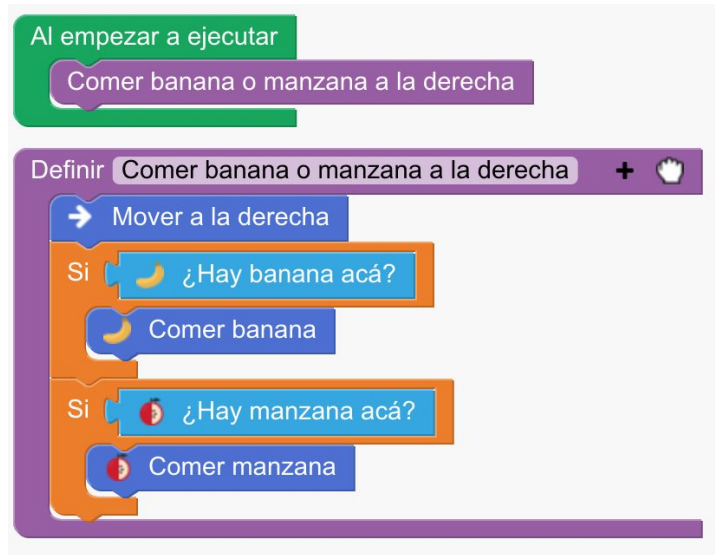
Nuevo problema

Veamos algunas soluciones posibles, y analicemos si funcionan en todos los escenarios o no.

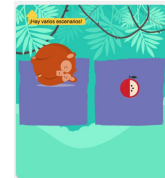


Nuevo problema

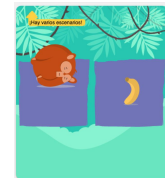
Veamos algunas soluciones posibles, y analicemos si funcionan en todos los escenarios o no.



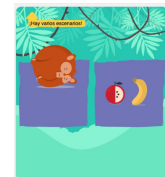
SI



SI



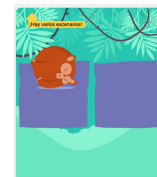
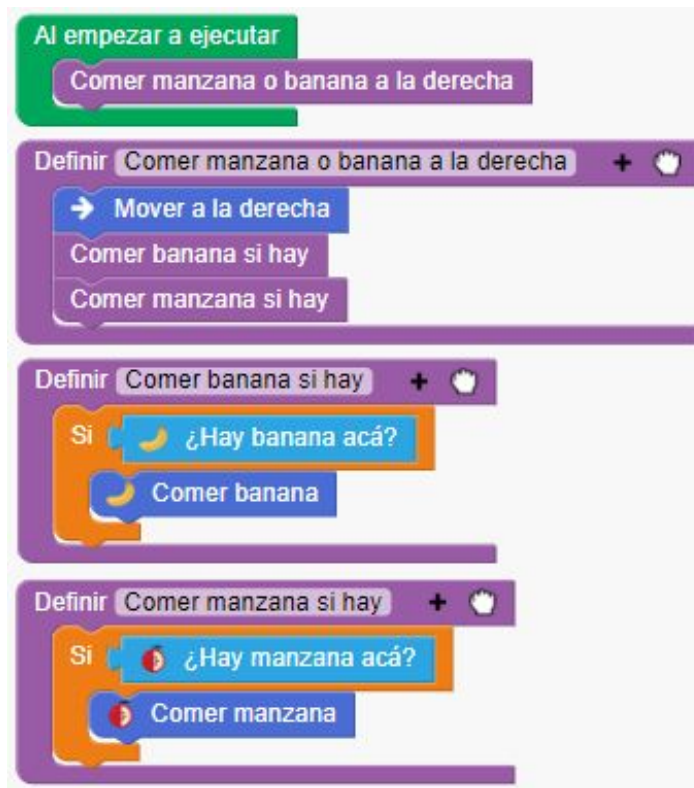
SI



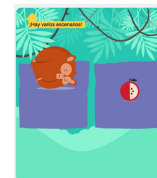
SI

Nuevo problema

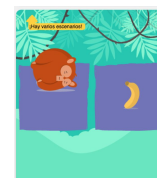
Una Solución que brinda una mejor claridad. Es equivalente a la anterior, aunque al separar en procedimientos mejora la comunicación



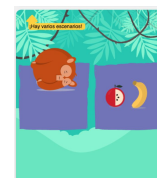
SI



SI



SI



SI



La solución adecuada, depende del problema

Soluciones adecuadas para un problema, pueden no ser adecuadas en otro problema que es “apenas diferente”. Por ej. en el ejercicio anterior, dar la posibilidad de que no hayan frutas, o que puedan haber ambas, cambia significativamente el problema en términos de código.

Debo tener presente los escenarios posibles, y elegir siempre una solución que sea adecuada según el problema a solucionar.

En papel: nuevas palabras claves y formas



Si <condición> **Entonces**

| <rama del si>

Sino

| <rama del sino>

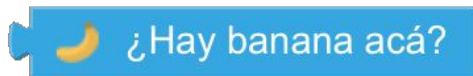
En papel: nuevas palabras claves y formas



Si <condición> **Entonces**

| <rama del si>

En papel: nuevas palabras claves y formas

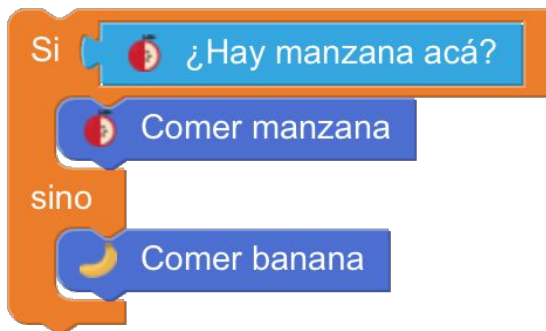


¿Hay banana acá?

Escribimos los sensores como preguntas, usando los signos de apertura y cierre de interrogación.

En papel: nuevas palabras claves y formas

Un ejemplo concreto:



Si ¿hay manzana acá? **Entonces**

| Comer manzana

Sino

| Comer banana

En papel: nuevas palabras claves y formas

Un ejemplo concreto:



Si ¿hay manzana acá? Entonces

| Comer manzana

Sino

| Comer banana

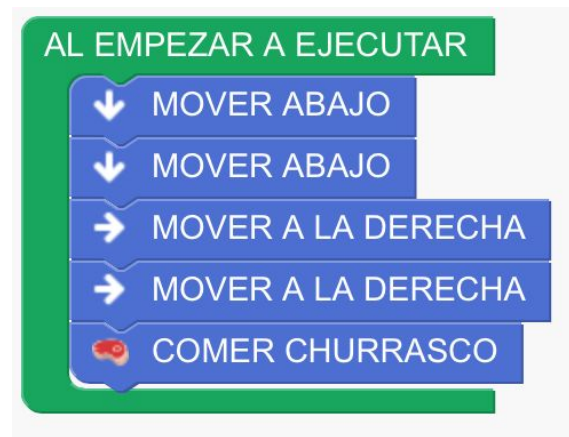
¡Ojo! El "Sino" es cuando no se cumplió la condición del "Sí", no lleva su propia condición. Y el cuerpo del "Sino" va abajo, con la línea para dejar claro dónde empieza y termina.



Momento de dudas o consultas

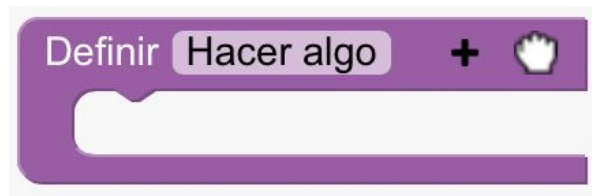
Repaso

- 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**.



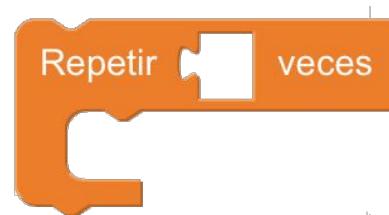
Repaso

- 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).**



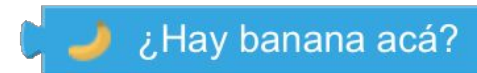
Repaso

- 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.A. Alternativas condicionales”.



Alternativas Condicionales

Nos vemos la próxima