

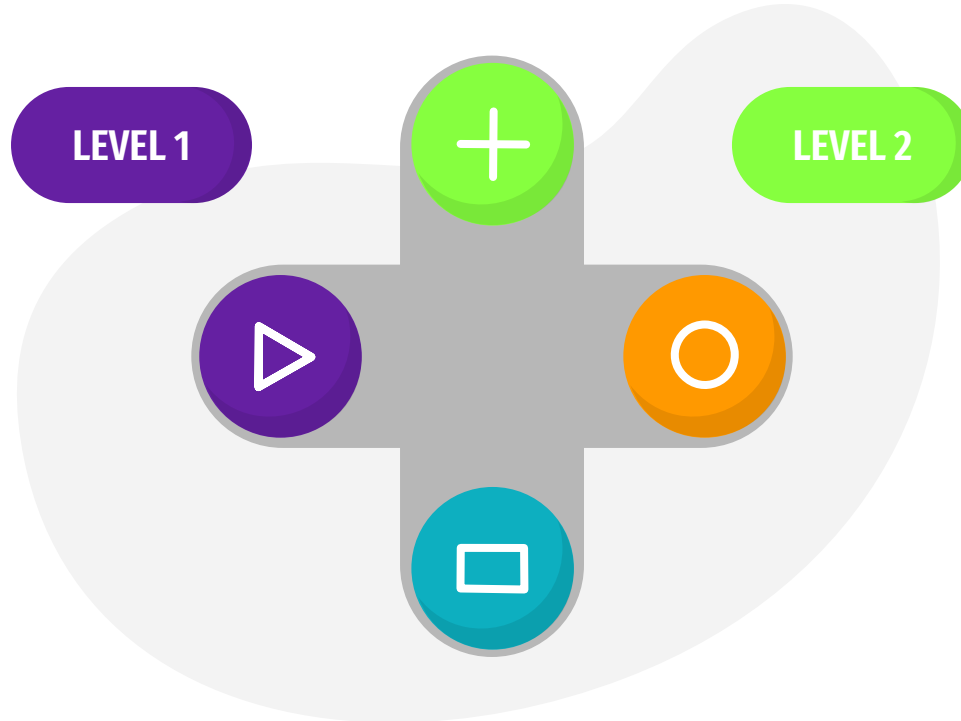


Fundamentos de Programación en Videojuegos I

Clase 17

Contenido de la clase

Definición de expresiones.



Repetición condicional.

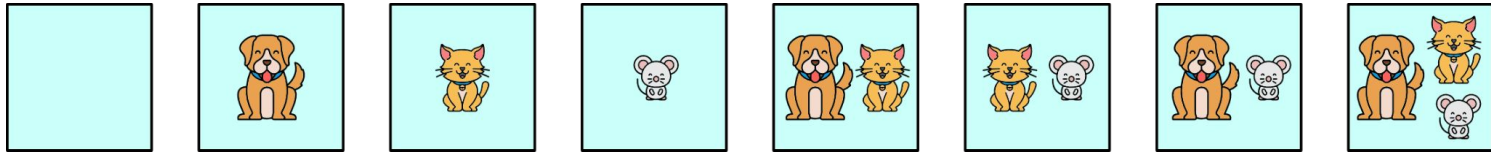
Definición de expresiones.



Ejercicio simple

- Se pide que identifique en qué escenarios la siguiente expresión es verdadera.

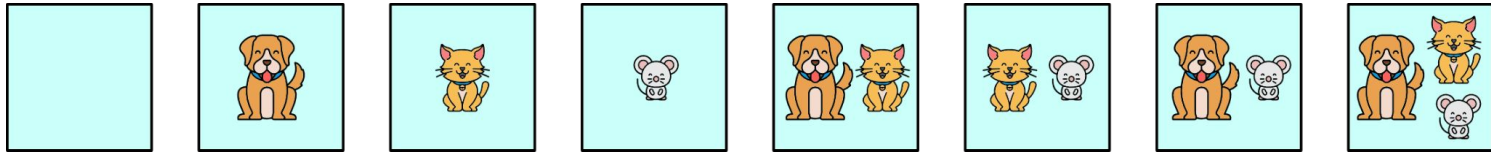
$(\text{¿hay perro?} \wedge \text{¿hay gato?} \wedge \neg \text{¿hay ratón?}) \vee (\text{¿hay gato?} \wedge \text{¿hay ratón?} \wedge \neg \text{¿hay perro?}) \vee$
 $(\neg \text{¿hay gato?} \wedge \text{¿hay ratón?} \wedge \text{¿hay perro?})$



Ejercicio simple

- Intentemos nuevamente, ahora con esta expresión.

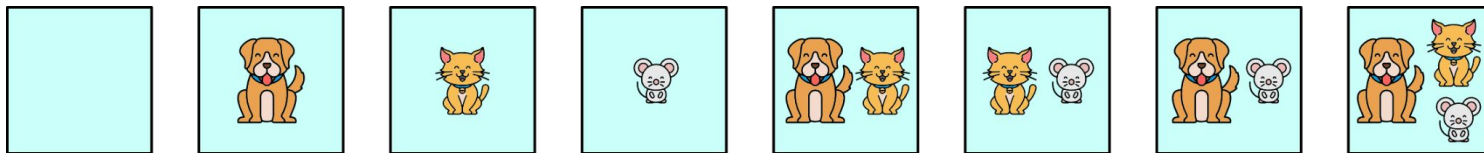
¿hay perro y gato solos? ∨ ¿hay gato y ratón solos? ∨ ¿hay perro y ratón solos?



Ejercicio simple

- Una más...

¿hay solo dos animales?



¿Comunicar a la máquina o a las personas?

Cuando exponemos todas las conectivas y los sensores básicos, ¿En quién pensamos? ¿Estamos pensando en seres humanos que van a leer eso? ¿O estamos pensando en satisfacer a la máquina?

Queremos que las expresiones comuniquen adecuadamente la idea que representan, igual que lo hacen los comandos.

En el mundo de los comandos, creábamos procedimientos y les dábamos nombres adecuados para que comuniquen bien las ideas.

¿Se puede hacer lo mismo con las expresiones? Spoiler, claro que se puede.

Definiendo expresiones

Hay que recordar que, a diferencia de los comandos que describen acciones, las expresiones describen datos.

La definición de una nueva expresión, va a estar dada en términos de equivalencia entre la nueva expresión que estamos definiendo, y expresiones de las que ya disponíamos. Ej.

Definir ¿hay perro y gato solos? = ¿hay perro? \wedge ¿hay gato? \wedge \neg ¿hay ratón?

La gracia es poder dar mejores nombres a grupos de expresiones compuestas de varias partes y conectivas.

Nombrando nuevas expresiones


Como vamos a definir únicamente expresiones de valores de verdad, la regla es simple. El nombre debe contener signos de interrogación, y corresponderse a una pregunta que se responda con sí o no.

No son verbos, ya que no hay acciones involucradas en una expresión. Es solo una pregunta, que contestamos en términos de otras preguntas.

Deben estar relacionados a lo que se quiere responder, y debe poder leerse claramente por una persona que no conoce el contexto del problema.

Sintaxis usada

Definir ¿hay perro y gato solos? = ¿hay perro? \wedge ¿hay gato? \wedge \neg ¿hay ratón?



Comenzamos por la palabra clave “Definir”, ya que al igual que con los procedimientos, estamos definiendo una cosa que antes no existía.

Sintaxis usada

Luego el nombre, que como dijimos, es una nueva pregunta.



Definir ¿hay perro y gato solos? = ¿hay perro? \wedge ¿hay gato? \wedge \neg ¿hay ratón?



Comenzamos por la palabra clave “Definir”, ya que al igual que con los procedimientos, estamos definiendo una cosa que antes no existía.

Sintaxis usada

Luego el nombre, que como dijimos, es una nueva pregunta.

Definir ¿hay perro y gato solos? = ¿hay perro? \wedge ¿hay gato? \wedge \neg ¿hay ratón?

Comenzamos por la palabra clave “Definir”, ya que al igual que con los procedimientos, estamos definiendo una cosa que antes no existía.

El signo igual indica que la pregunta que definimos, es equivalente a realizar las otras preguntas de la derecha.

Sintaxis usada

Luego el nombre, que como dijimos, es una nueva pregunta.

Finalmente, indicamos a qué es equivalente la pregunta que acabamos de definir. Mínimo dos partes unidas por conectivas.

Definir ¿hay perro y gato solos? = ¿hay perro? \wedge ¿hay gato? \wedge \neg ¿hay ratón?

Comenzamos por la palabra clave “Definir”, ya que al igual que con los procedimientos, estamos definiendo una cosa que antes no existía.

El signo igual indica que la pregunta que definimos, es equivalente a realizar las otras preguntas de la derecha.

Sintaxis usada

Consiste en una definición de equivalencia, que se lee de izquierda a derecha.

Definir ¿hay perro y gato solos? = ¿hay perro? \wedge ¿hay gato? \wedge \neg ¿hay ratón?

Sintaxis incorrecta

La definición debe leerse de izquierda a derecha. Si la palabra definir se pone del lado derecho de la igualdad, se pierde, y no se entiende la definición. Por más que haya un igual, no funciona como en matemática, donde se pueden cambiar los elementos de lado.

$\text{¿hay perro?} \wedge \text{¿hay gato?} \wedge \neg \text{¿hay ratón?} = \text{Definir ¿hay perro y gato solos?}$

Múltiples definiciones

Definir ¿hay perro y gato solos? = ¿hay perro? \wedge ¿hay gato? \wedge \neg ¿hay ratón?

Definir ¿hay perro y ratón solos? = ¿hay perro? \wedge \neg ¿hay gato? \wedge ¿hay ratón?

Definir ¿hay ratón y gato solos? = \neg ¿hay perro? \wedge ¿hay gato? \wedge ¿hay ratón?

Podemos realizar varias definiciones, para distintas expresiones, según lo necesitemos.

Múltiples definiciones

Definir ¿hay perro y gato solos? = ¿hay perro? \wedge ¿hay gato? \wedge \neg ¿hay ratón?

Definir ¿hay perro y ratón solos? = ¿hay perro? \wedge \neg ¿hay gato? \wedge ¿hay ratón?

Definir ¿hay ratón y gato solos? = \neg ¿hay perro? \wedge ¿hay gato? \wedge ¿hay ratón?

**Definir ¿hay solo dos animales? = ¿hay perro y gato solos? \vee ¿hay perro y ratón solos?
 \vee ¿hay ratón y gato solos?**

Podemos utilizar definiciones ya realizadas para realizar nuevas. Es decir, la misma idea que con las subtareas, pero para expresiones, dejando más claro qué significa cada parte.

Usando definiciones

Al empezar a ejecutar

- | Alimentar animales solo si son dos

Definir ¿hay perro y gato solos? = ¿hay perro? \wedge ¿hay gato? \wedge \neg ¿hay ratón?

Definir ¿hay perro y ratón solos? = ¿hay perro? \wedge \neg ¿hay gato? \wedge ¿hay ratón?

Definir ¿hay ratón y gato solos? = \neg ¿hay perro? \wedge ¿hay gato? \wedge ¿hay ratón?

Definir ¿hay solo dos animales? = ¿hay perro y gato solos? \vee ¿hay perro y ratón solos?
 \vee ¿hay ratón y gato solos?

Definir Alimentar animales solo si son dos

- | Sí ¿hay solo dos animales? Entonces

- | | Dejar comida

Repetición condicional.



A trabajar



Realizamos del libro del “Nivel intermedio”, el ejercicio “LARGOS CAMBIANTES” de la sección “Repetición Condicional”.

LARGOS CAMBIANTES: ¿Qué aprendimos?

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



LARGOS CAMBIANTES: ¿Qué aprendimos?

Ahora el escenario cambia, pero de forma distinta a lo que sucedía hasta ahora. Antes, la cantidad de lugares era fija, y sólo cambiaba el hecho de que podía o no haber un determinado elemento una ubicación particular.

Ahora la cantidad de lugares del escenario es variable.

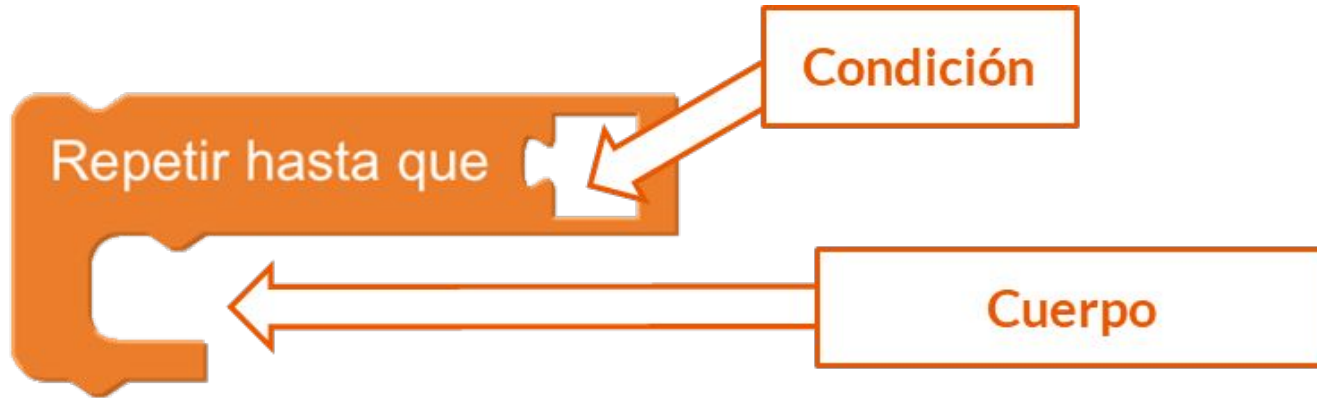
Nuevamente, es imposible solucionar este problema con las herramientas al momento. Hay que repetir para llegar al final del camino, pero ¿Cuántas veces repetir sí el escenario cambia y no sabemos exactamente dónde terminará cada vez el camino?



Repetición Condicional

- La repetición condicional es otra forma de repetición.
- Se suma entonces a la repetición simple y a la alternativa como una estructura de control del flujo.
- No es una nueva forma de organizar el código, porque sigue siendo una repetición.
- A diferencia de la repetición simple, la cantidad de veces a repetir no es fija, sino que depende de una condición.
- Se repite hasta que cierta condición se cumpla, es decir, hasta que sea verdadera.
- Es un comando compuesto, que espera una condición y un cuerpo.

Repetición Condicional



Repetición Condicional

Lo más difícil de comprender de esta estructura es cuándo se evalúa la condición.

La condición se evaluará múltiples veces:

1. Al llegar por primera vez al punto donde se encuentra la repetición condicional, la máquina evaluará la condición una primera vez. Sí es falsa, ejecutará el cuerpo (sí es verdadero no ejecutará el cuerpo y seguirá con el comando siguiente a la repetición)
2. Luego volverá a evaluar la condición. Sí vuelve a ser falsa, volverá a ejecutar el cuerpo (sí es verdadera, continuará con el comando siguiente)
3. Seguirá realizando 2 hasta que la condición sea verdadera.

Repetición Condicional

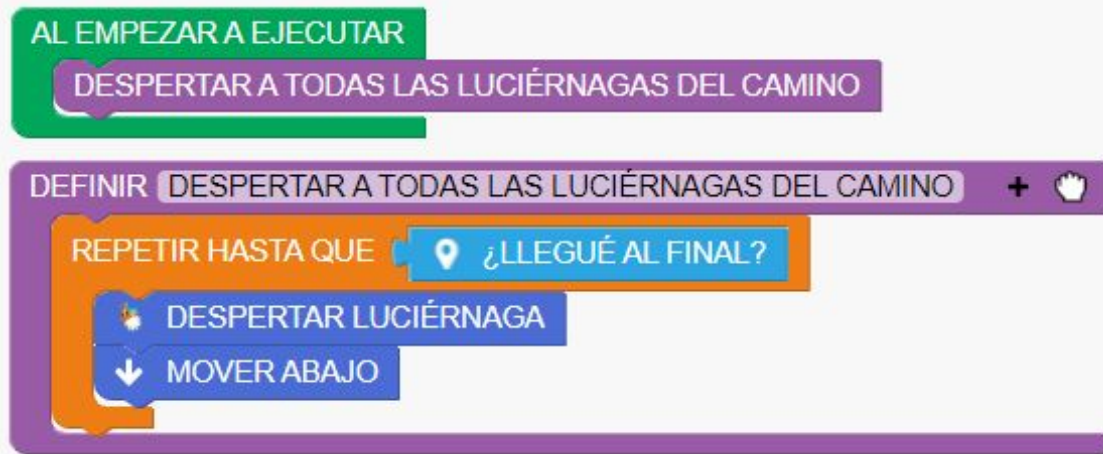
¿Cuántas veces se ejecuta el cuerpo?

Repetición Condicional

¿Cuántas veces se ejecuta el cuerpo?

- Las que sean necesarias para que se cumpla la condición (podría ser ninguna).

LARGOS CAMBIANTES: ¿Qué aprendimos?



A trabajar



Realizamos del libro del “Nivel intermedio”, el ejercicio “LUCES CAMBIANTES” de la sección “Repetición Condicional”.

LUCES CAMBIANTES: ¿Qué aprendimos?

Hay que combinar repetición condicional y alternativa condicional.

¡¡Ojo, no confundir las condiciones!!

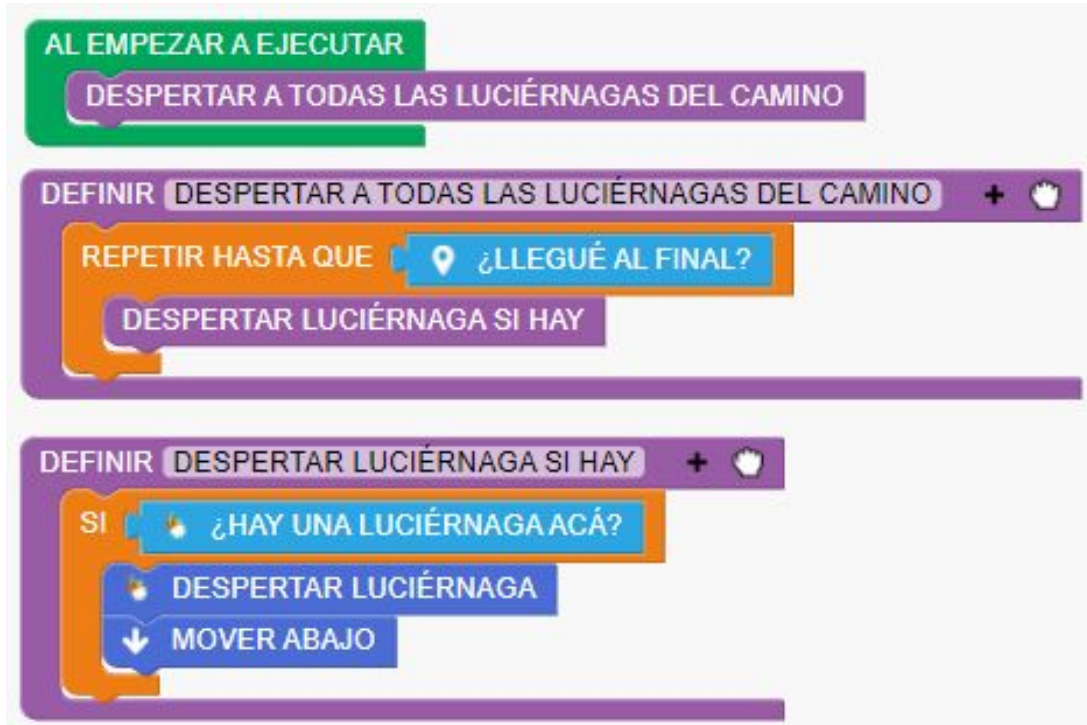
¿Cómo evitar errores?

Fácil, dividir en subtareas usando procedimientos, con nombres claros.



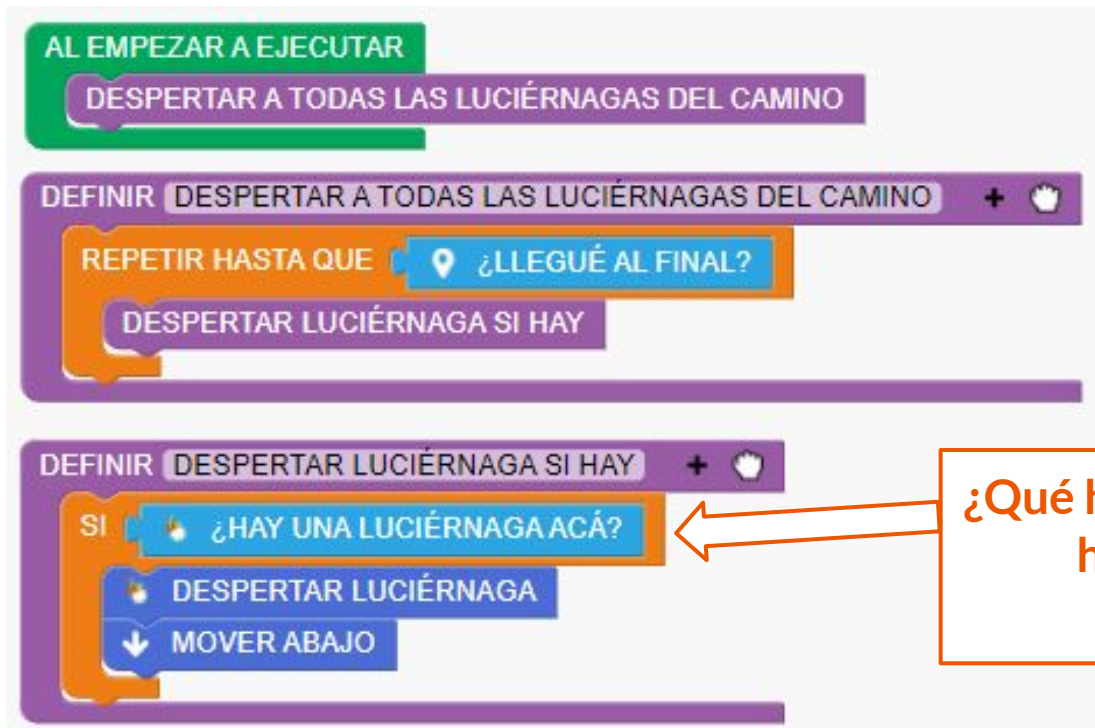
LUCES CAMBIANTES: ¿Qué aprendimos?

Veamos un error común :



LUCES CAMBIANTES: ¿Qué aprendimos?

Veamos un error común :



¿Qué hace Yvoty cuando no hay luciérnagas?
¡No se mueve!



Fallos por no terminación

- Cuando no hay una luciérnaga, el programa “se cuelga”.
- El problema es que la máquina pregunta por la condición de la repetición, y como no se cumple, ejecuta el cuerpo. Pero el cuerpo no hace nada, porque no hay luciérnaga. Por tanto, se termina el cuerpo, y se vuelve a preguntar por la condición, que otra vez estará sin cumplirse.
- Este es otro motivo por el cual un programa puede fallar.



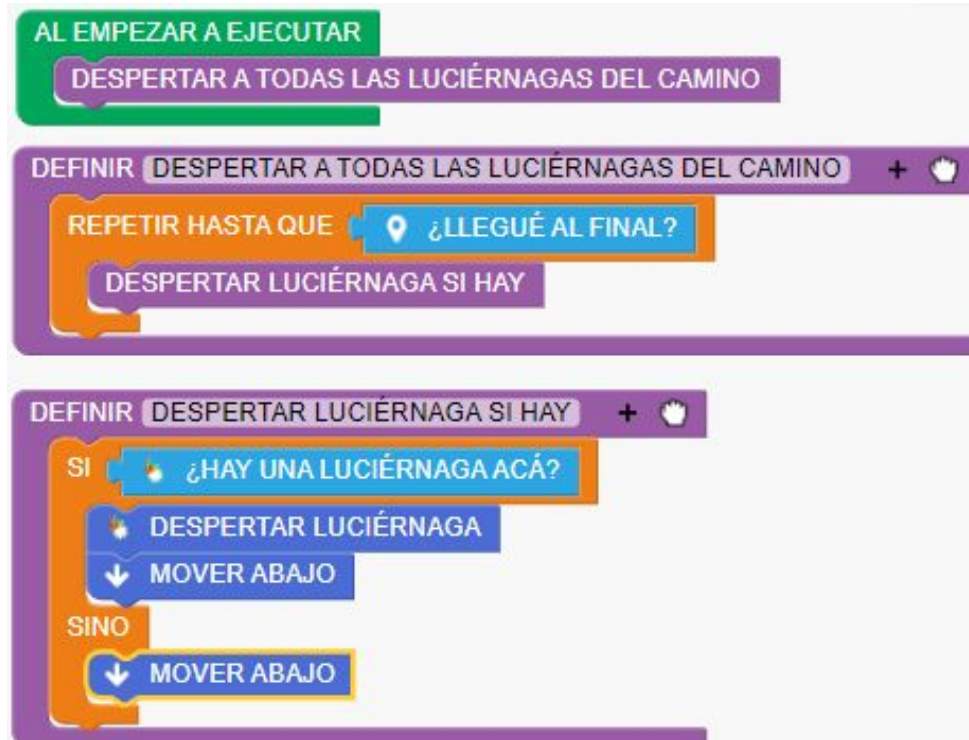
Fallos por no terminación

- Para evitar los fallos por no terminación, hay que asegurarse que dentro una repetición condicional siempre se ejecute una acción que haga que nos acerquemos a la condición de corte.
- Sí tenemos que llegar a cierta ubicación, siempre deberíamos movernos hacia esa ubicación, para asegurar que eventualmente vamos a llegar.



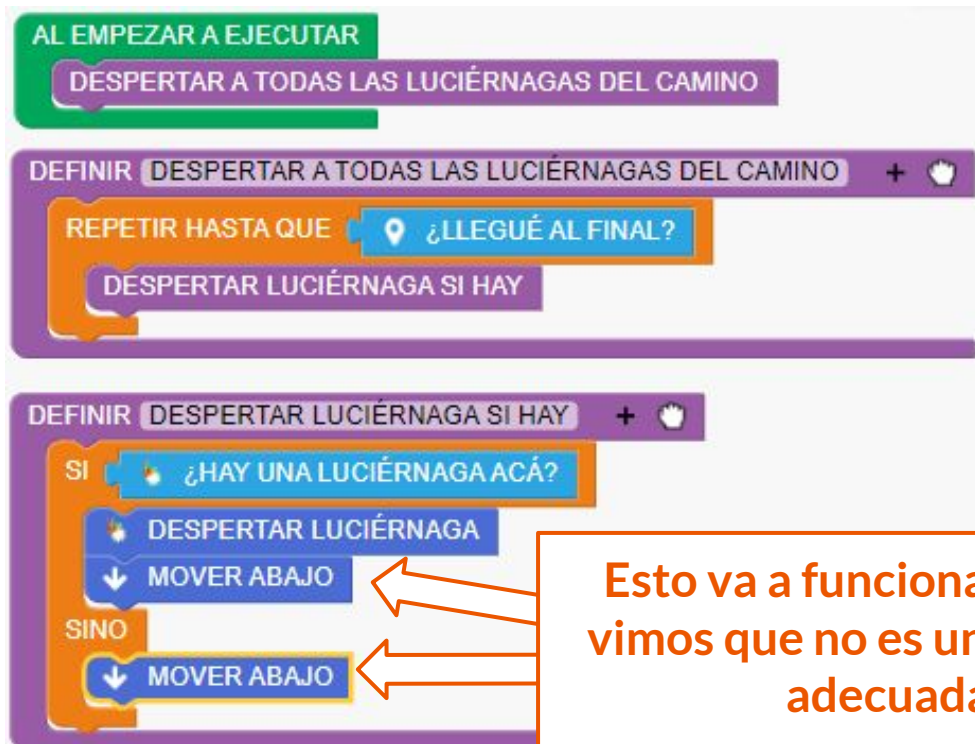
LUCES CAMBIANTES: ¿Qué aprendimos?

Una segunda aproximación:



LUCES CAMBIANTES: ¿Qué aprendimos?

Una segunda aproximación:

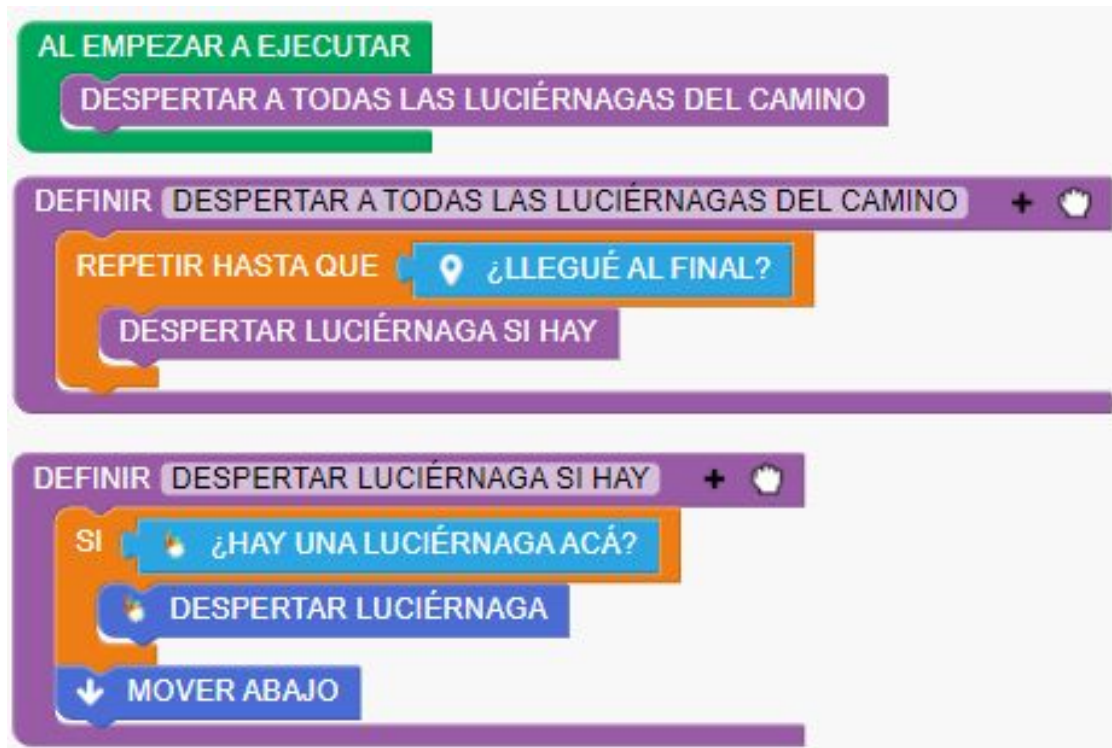


Esto va a funcionar, pero ya vimos que no es una solución adecuada.



LUCES CAMBIANTES: ¿Qué aprendimos?

Una solución más adecuada.



Fallos por no terminación

Sí seguimos las prácticas que venimos trabajando (no anidar bloques y separar el problema mediante procedimientos), los riesgos disminuyen significativamente.

En particular, la mayoría de los problemas que usan repetición condicional pueden plantearse en términos de dos partes bien claras, que van en el cuerpo de la repetición, y deben realizarse hasta que se cumpla una condición:

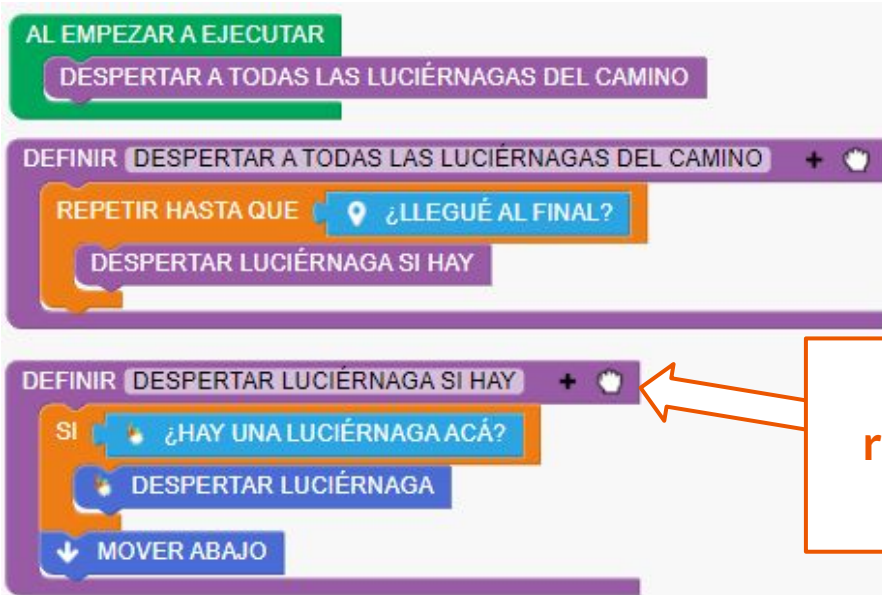
Una acción que debe ser realizada en cada ubicación (esa acción puede ser algo condicionado, por ej. “Despertar una luciérnaga, sí hay una”).

Pasar a la siguiente ubicación donde debe realizarse la acción.



Fallos por no terminación

P.D. Sí separamos bien en subtareas, dividiendo el “procesar” del “avanzar”, como vimos cuando charlamos de recorridos, entonces este error no debería ocurrirnos nunca.

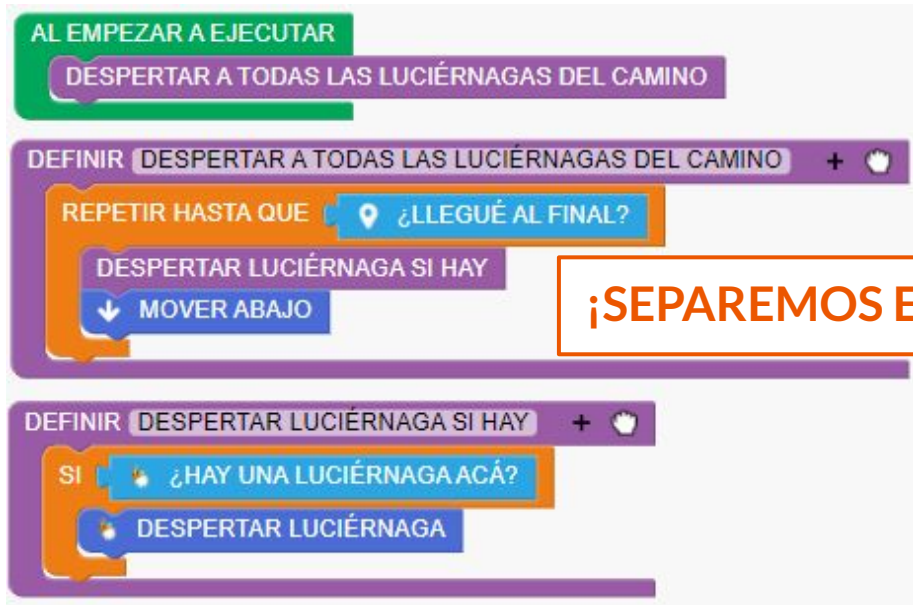


Pero...este nombre ¿es representativo de lo que hace el procedimiento?



Fallos por no terminación

Una excelente solución. El hacer la subtarea permite ver claramente la acción y el pasar al siguiente y ayuda a prevenir el error por no terminación.

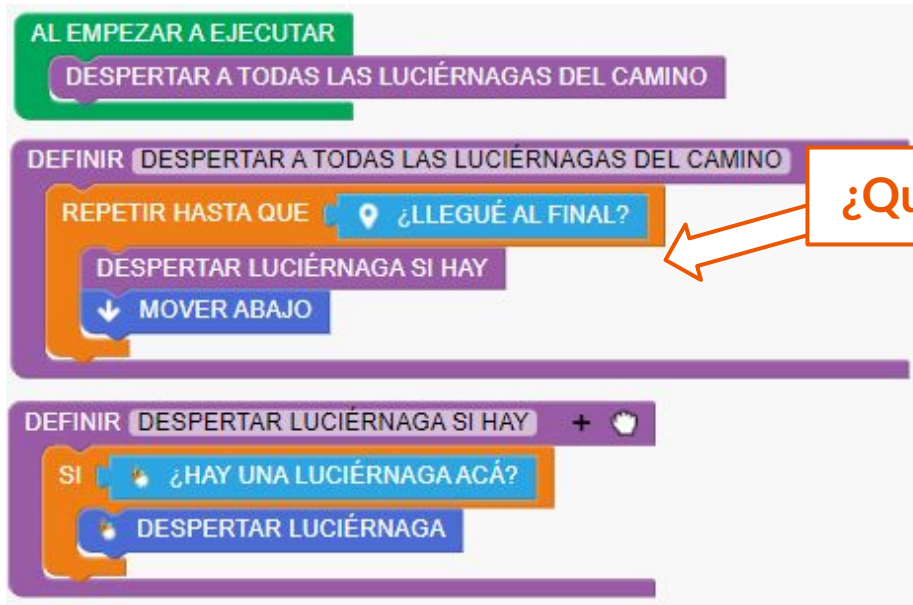


¡SEPAREMOS EN SUBTAREAS!

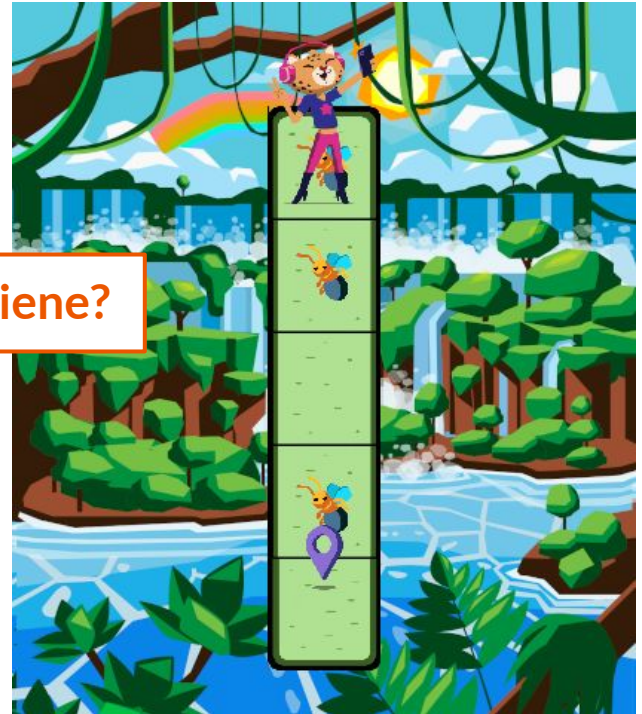


Fallos por no terminación

Una excelente solución. El hacer la subtarea permite ver claramente la acción y el pasar al siguiente y ayuda a prevenir el error por no terminación.

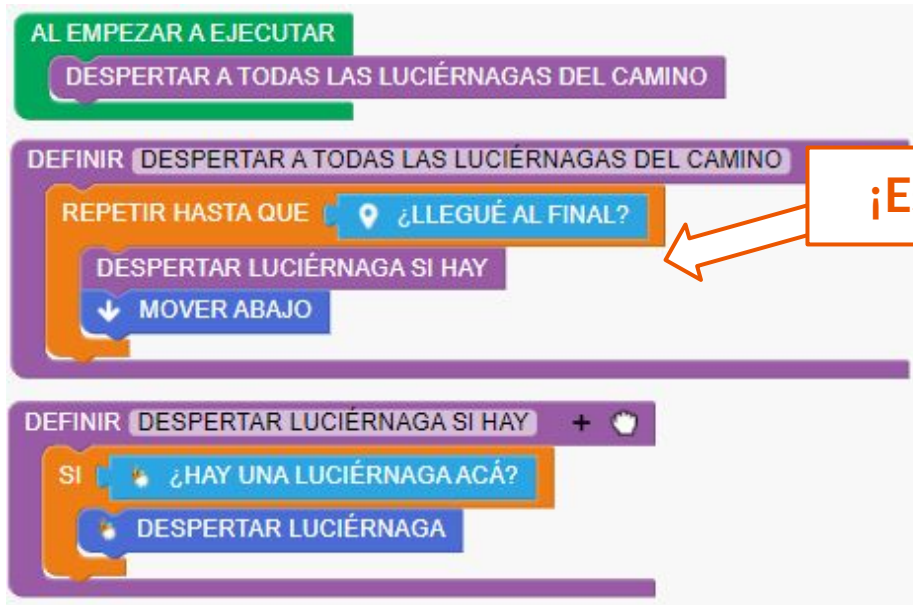


¿Qué forma tiene?



Fallos por no terminación

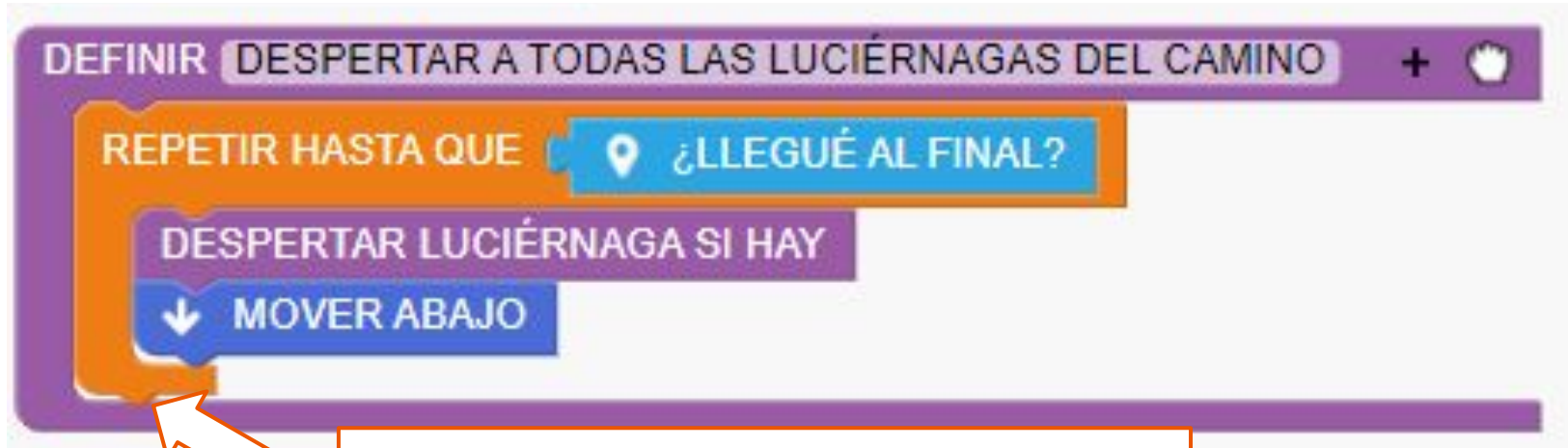
Una excelente solución. El hacer la subtarea permite ver claramente la acción y el pasar al siguiente y ayuda a prevenir el error por no terminación.



¡Es un recorrido!, ¡Claro que sí!

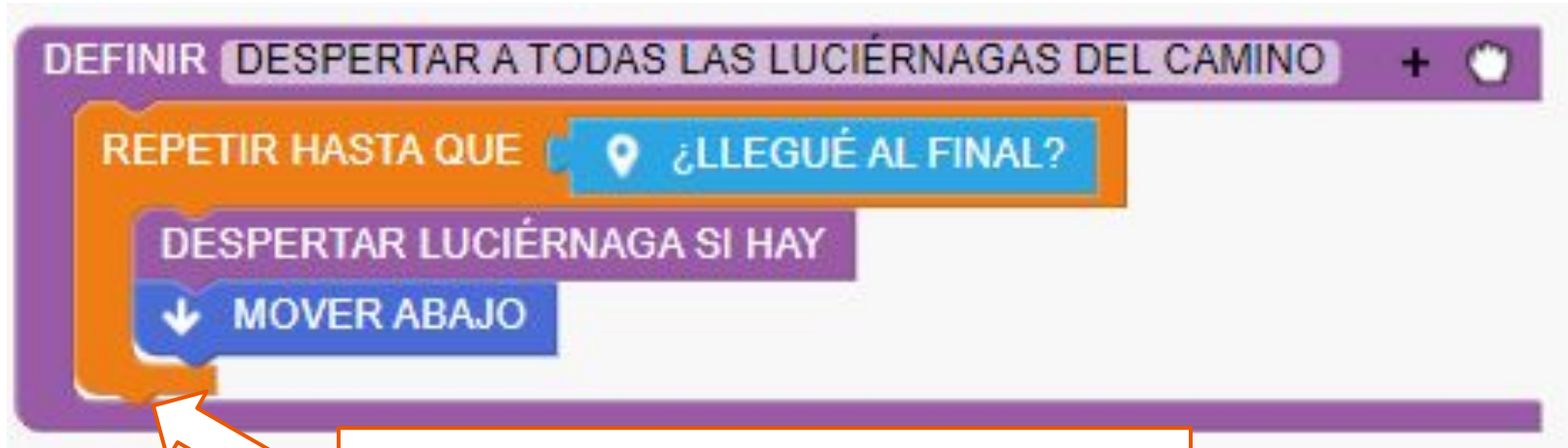


Condiciones antes, después y dentro de la repetición



¿Puede pasar que llegue a este punto del código y NO esté en el final?

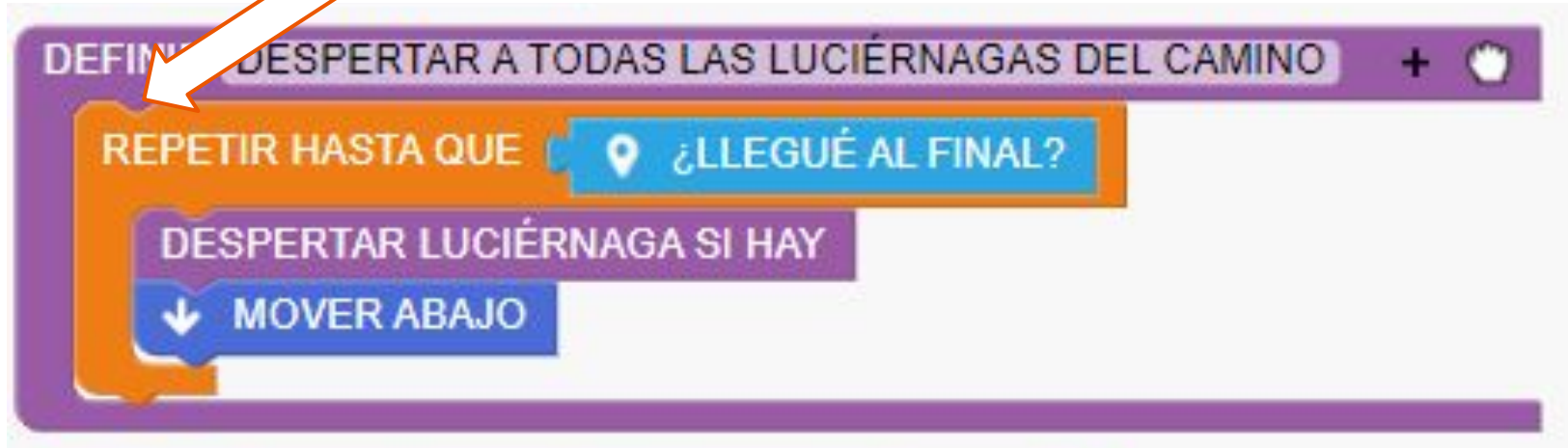
Condiciones antes, después y dentro de la repetición



NO. Si no hubiéramos llegado al final no habríamos terminado de repetir.

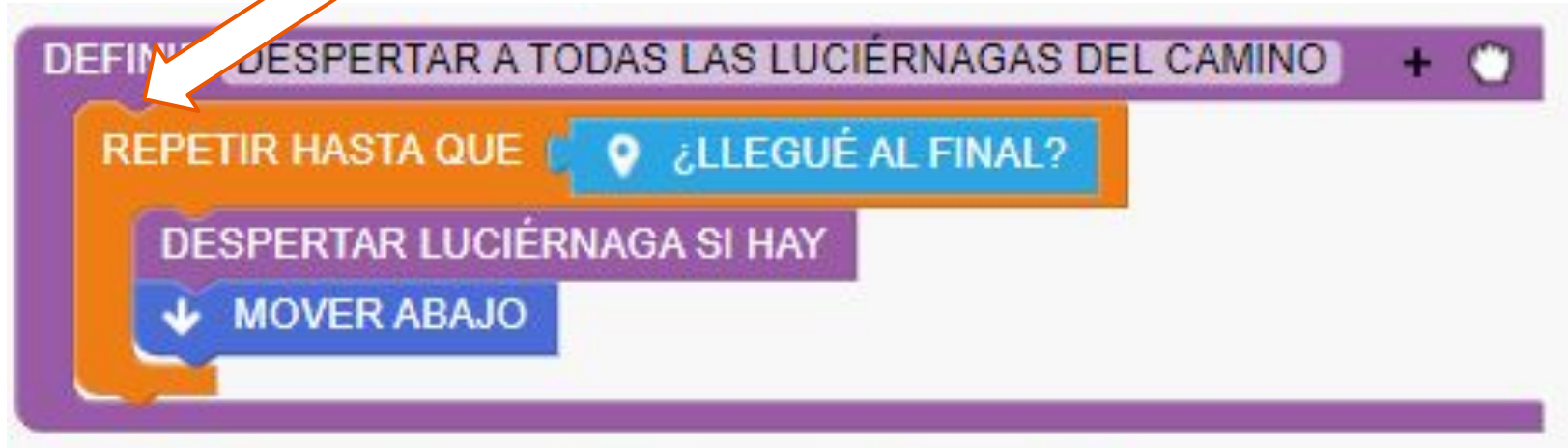
Condiciones antes, después y dentro de la repetición

¿Puede pasar que al arrancar ya esté en el final?

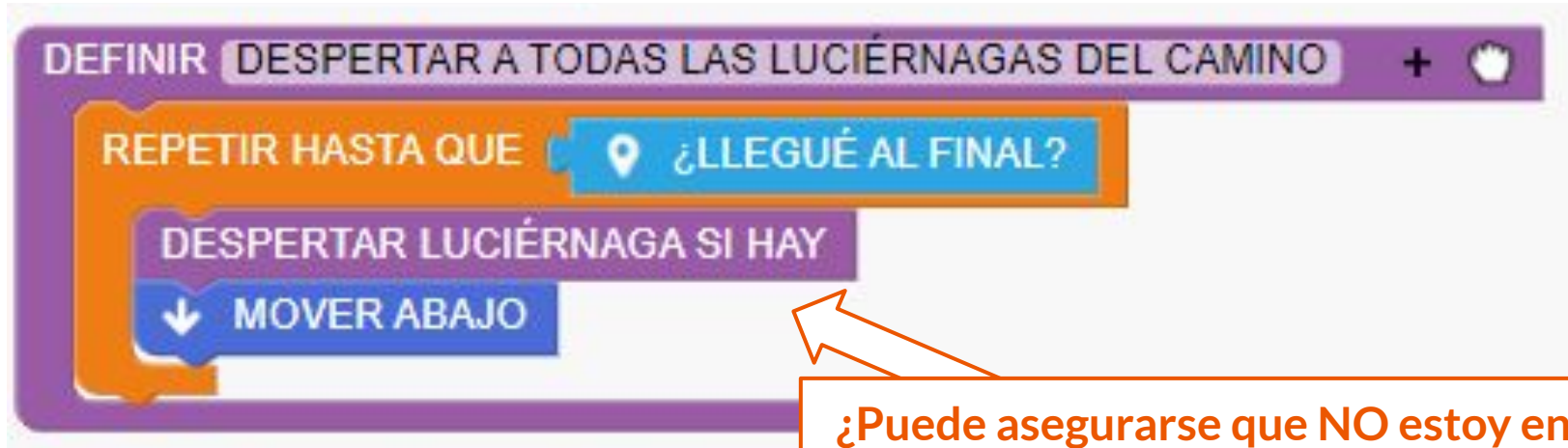


Condiciones antes, después y dentro de la repetición

Claro que sí, en ese caso nunca se entra a la repetición.

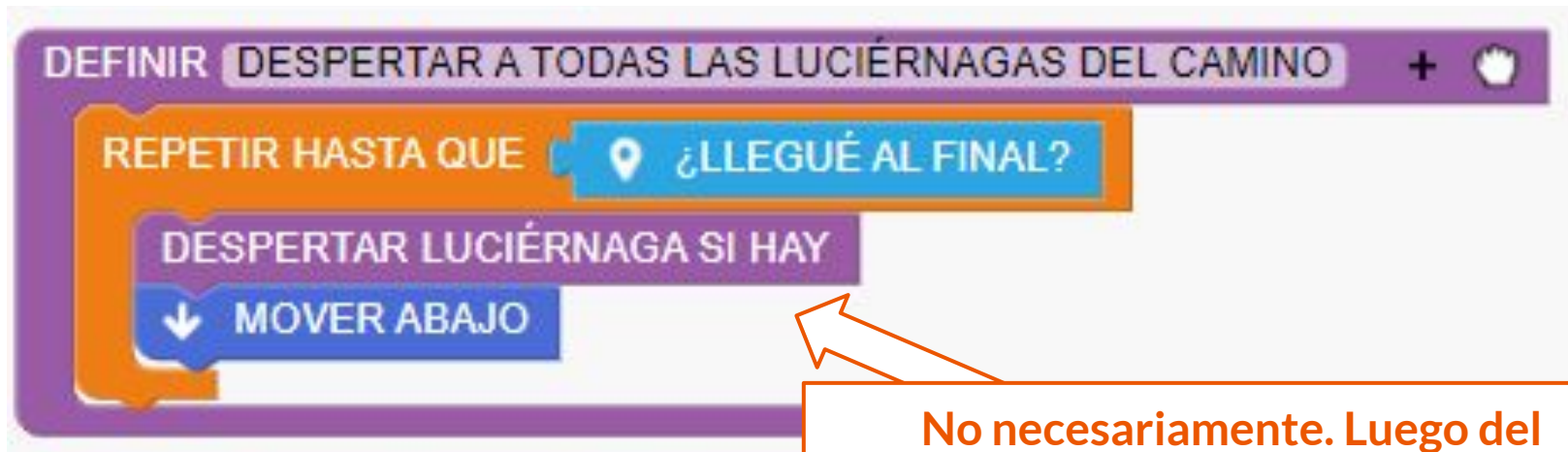


Condiciones antes, después y dentro de la repetición



¿Puede asegurarse que NO estoy en el final dentro del cuerpo?

Condiciones antes, después y dentro de la repetición



No necesariamente. Luego del movimiento puedo haber llegado al final (de hecho, pasa en el último ciclo de la repetición)

Repetición condicional y recorridos

- Hablamos de “**Recorridos simples**”, porque usábamos repetición simple.
- Ahora podemos hablar de “**recorridos**” propiamente dichos.

Repetición condicional y recorridos

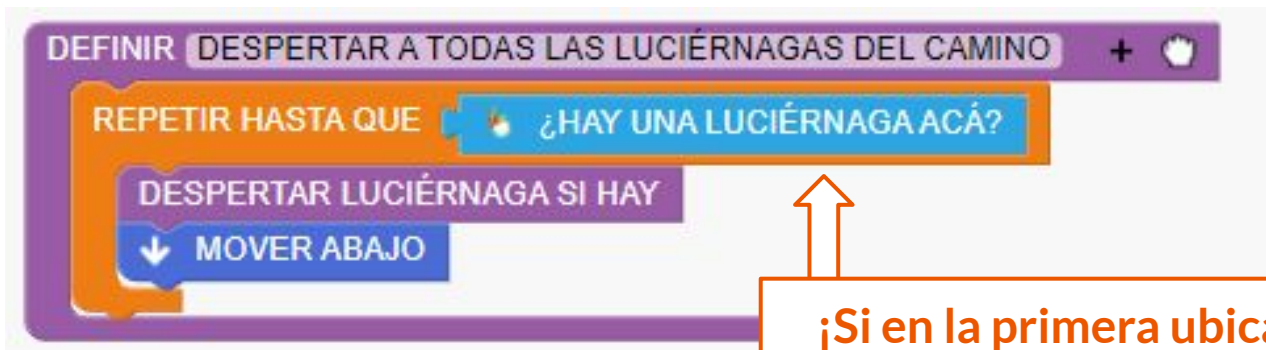
La idea conceptual es la misma, el recorrido va a tener:

- Una repetición principal (simple o condicional)
- Una subtarea (puede ya venir como primitiva) que “procesa” el elemento actual. Puede darse el caso de que en algunos recorridos no haga falta procesar nada en cada elemento. Incluso puede pasar que el “procesado” sea condicional (a veces proceso, a veces no), por lo que usaremos alternativa condicional (ej. Despertar una luciérnaga, solo sí hay una).
- Una subtarea (puede ya venir como primitiva) que “avanza” al próximo elemento en el recorrido.
- Puede incluir (o no) casos de borde, tanto antes como después.

Repetición condicional y recorridos

Un error común:

- Al tener que procesar de forma condicional, confundir la condición de “hasta cuándo recorrer” con la de “cuándo procesar”. Ej.

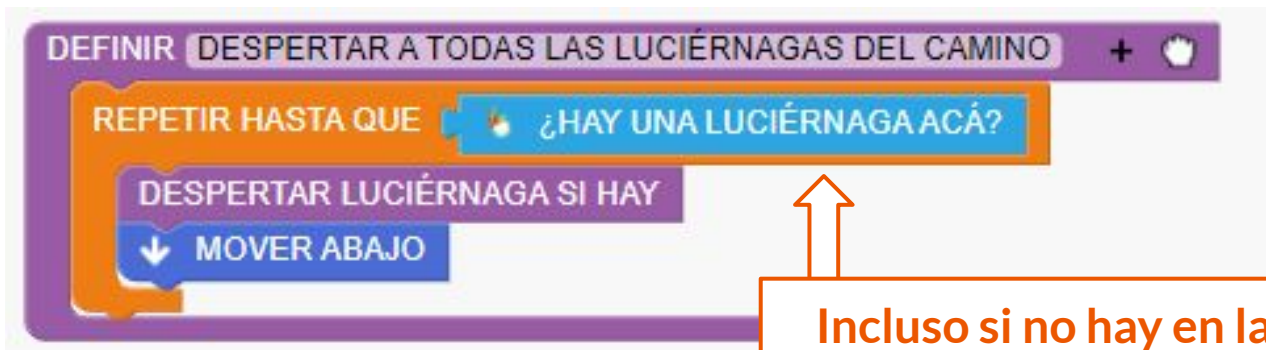


¡Si en la primera ubicación hay una luciérnaga, termina ahí!

Repetición condicional y recorridos

Un error común:

- Al tener que procesar de forma condicional, confundir la condición de “hasta cuándo recorrer” con la de “cuándo procesar”. Ej.



Incluso si no hay en la primera, termina cuando encuentre una, aunque queden un montón de ubicaciones para ver.

Sintaxis en papel



Repetir hasta que <condición>
| <cuerpo>

Tarea de la clase 17

EJERCITARIO 6 - DEFINICIÓN DE EXPRESIONES Y REPETICIÓN CONDICIONAL

