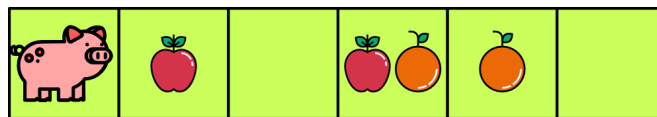


## Recorridos simples

### Ejercicio 1) Babe, el chanchito comelón

Babe, el cerdito comelón, vive en una granja en donde sus dueños disponen para él de manzanas y naranjas, todo un festín para un cerdito. Babe come todo lo encuentra a su paso, no deja absolutamente nada. Pero claro, sus dueños dejan la comida en cualquier lugar del corral, por lo que Babe debe aprender que no siempre las manzanas y las naranjas están en las mismas ubicaciones. En una ubicación puede haber solo una manzana, en otras solo una naranja, en otras pueden haber ambas frutas, e incluso en otros lugares no haber nada. El corral, desde ya, tiene siempre 6 lugares, y Babe siempre espera su almuerzo en la ubicación más a la izquierda, donde con seguridad no hay comida. Esto se muestra en el siguiente escenario:

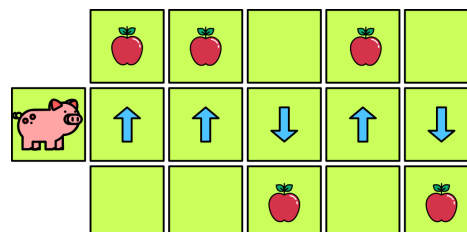


Ayuda a Babe a comer todo lo que encuentre, escribiendo un programa que lo manipule y utilizando las primitivas y sensores a continuación:

<b>Mover a la derecha</b> Mueve a Babe un lugar a la derecha. Debe haber lugar a la derecha.	<b>Comer manzana</b> Hace que Babe coma la manzana en la ubicación donde se encuentra. Falla sí no hay una manzana en la ubicación.
<b>Comer naranja</b> Hace que Babe coma la naranja en la ubicación donde se encuentra. Falla sí no hay una naranja en la ubicación.	<b>¿hay una manzana?</b> Indica sí hay una manzana en la ubicación donde se encuentra Babe.
<b>¿hay una naranja?</b> Indica sí hay una naranja en la ubicación donde se encuentra Babe.	

### Ejercicio 2) Babe, el chanchito comelón 2

Babe enfermó un poco y los granjeros lo han movido a otro corral, más ancho, para que tenga mayor movilidad, y han terminado su dieta de naranjas, para terminar con su acidez. Los granjeros ahora alimentan a Babe solo con manzanas que colocan en los bordes de arriba y abajo del corral. Nuevamente disponen manzanas a veces en un lado y a veces en otro (y se garantiza que siempre hay una manzana, o bien a uno o bien a otro lado). Para ayudar a babe, hay flechas en el centro del corral que indican hacia qué lado se encuentran las manzanas. Todo puede apreciarse en el escenario siguiente:



Nuevamente el objetivo es hacer un programa que ayude a Babe a comer todo lo que hay en el corral, pero esta vez tenemos diferentes primitivas:

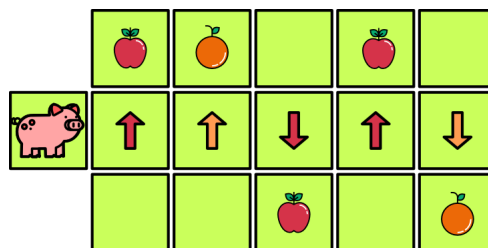
<b>Mover abajo</b> Mueve a Babe un lugar abajo. Debe haber lugar abajo.	<b>Mover arriba</b> Mueve a Babe un lugar arriba. Debe haber lugar arriba.
<b>Mover a la derecha</b>	<b>Comer manzana</b>

Mueve a Babe un lugar a la derecha. Debe haber lugar a la derecha.	Hace que Babe coma la manzana en la ubicación donde se encuentra. Falla sí no hay una manzana en la ubicación.
<b>¿la flecha apunta arriba?</b> Indica sí hay una flecha en la ubicación donde se encuentra Babe que apunta hacia arriba.	

**Pista:** Recuerde volver al camino luego de haber comido una fruta

### Ejercicio 3) Babe, el chanchito comelón 3

Ahora que Babe está curado, los granjeros decidieron devolverle su dieta normal, a base tanto de naranjas como de manzanas, pero manteniendo a Babe en el nuevo corral. Así, ahora Babe tendrá no solo flechas que indican hacia qué lado está la fruta a comer, sino que además, la flecha será de distinto color según la fruta que se debe comer. Sí la flecha es naranja, se espera una naranja, mientras que sí es roja, se espera una manzana.

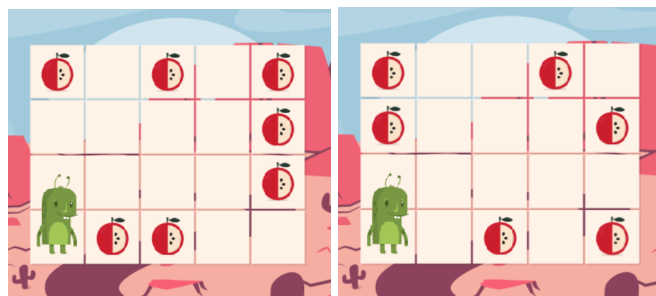


Nuevamente el objetivo es hacer un programa que ayude a Babe a comer todo lo que hay en el corral, y justo a las primitivas del ejercicio anterior tenemos la primitiva **¿la flecha es roja?** que indica sí la flecha en donde está Babe es de color rojo, además de poder **Comer naranja** como en el primer ejercicio.

### Ejercicio 4) El marciano vuelve a tener hambre

Nuevamente el marciano está en el desierto, y vuelve a tener hambre. Deberá comer todas las manzanas que haya en el desierto, las cuales ahora pueden estar en cualquier lugar del escenario. Esta vez el escenario es variable, y el lugar en donde se encuentran las manzanas no es fijo, sino que una ubicación puede o no haber una manzana, y no se sabe a priori. Incluso pueden haber manzanas en el lugar donde arranca el marciano, y también debe comerse dicha manzana.

Se muestran a continuación dos posibles escenarios iniciales, pero el escenario inicial podría ser cualquier otro que cumpla la descripción arriba mencionada.



Las primitivas son las ya utilizadas en el ejercicio "El marciano en el desierto" de PilasBloques (**Mover a la derecha, Mover a la izquierda, Mover arriba, Mover abajo y Comer manzana**). A la cual agregamos un sensor (**¿hay manzana?**) y la posibilidad de usar alternativa condicional en cualquiera de sus formas.

Se pide entonces que escriba un programa que resuelva el problema del marciano.

**Pista:** Como el camino no es lineal, se vuelve difícil moverse. Se sugiere piense dos subtarefas principales para solucionar el problema completo. Por un lado, piense como comer todas las manzanas de una única

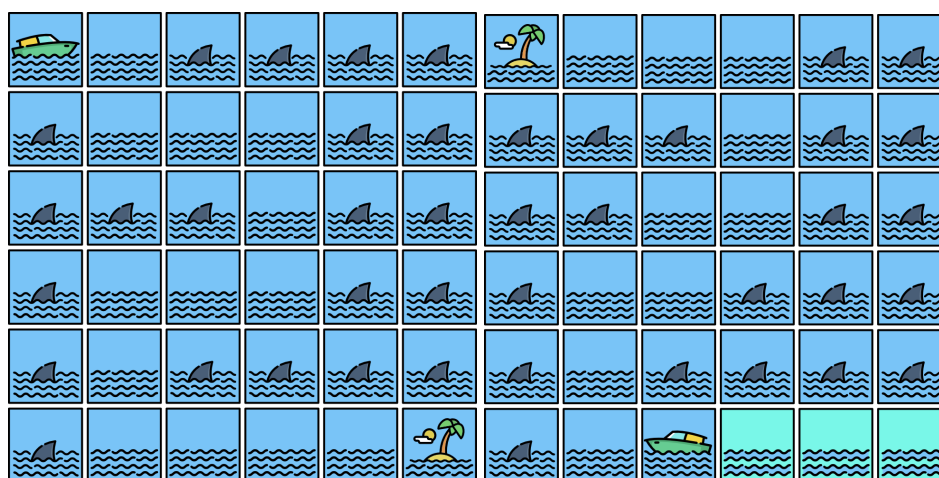
fila. Por otro lado, piense cómo ir desde el borde derecho de una fila al borde izquierdo de la fila siguiente (o sea, ir al comienzo de la próxima fila). Combine dichas subtareas para solucionar el problema principal.

## Ejercicio 5) La isla del sol

Todo está listo para llegar a la paradisíaca Isla del Sol, el bote tiene combustible, la tierra está a la vista, el mar está calmo y el día está soleado. Pero no será tan fácil, pues hay que encontrar el camino que no tiene tiburones para poder llegar de forma segura al destino.

En esta actividad el escenario puede variar, tanto en el lugar de donde arranca el bote, como el lugar donde se encuentra la isla, como el camino a seguir para llegar. Lo bueno es que sabemos que la isla está a exactamente 14 ubicaciones de distancia, sin importar cual sea el camino, además de que siempre hay un camino.

Pueden verse dos ejemplos de escenario a continuación, en el primero, el bote aún no ha arrancado el recorrido, mientras que en el segundo ya ha dado 3 pasos por el camino sin tiburones.



El bote es nuestro autómatas, y entiende las siguientes primitivas:

<b>Mover arriba</b> Hace que el bote se mueva una ubicación hacia arriba. Falla si no hay más lugar arriba o si hay tiburones en dicha dirección.	<b>Mover abajo</b> Hace que el bote se mueva una ubicación hacia abajo. Falla si no hay más lugar abajo o si hay tiburones en dicha dirección.
<b>Mover a la izquierda</b> Hace que el bote se mueva una ubicación hacia la izquierda. Falla si no hay más lugar a la izquierda o si hay tiburones en dicha dirección.	<b>Mover a la derecha</b> Hace que el bote se mueva una ubicación hacia la derecha. Falla si no hay más lugar a la derecha o si hay tiburones en dicha dirección.
<b>¿sigue el camino arriba?</b> Indica si el camino del bote sigue hacia arriba (es decir, no hay tiburones o se acaba el camino hacia arriba, y el bote no viene de allí)	<b>¿sigue el camino abajo?</b> Indica si el camino del bote sigue hacia abajo (es decir, no hay tiburones o se acaba el camino hacia abajo, y el bote no viene de allí)
<b>¿sigue el camino a la izquierda?</b> Indica si el camino del bote sigue hacia la izquierda (es decir, no hay tiburones o se acaba el camino hacia la izquierda, y el bote no viene de allí)	<b>¿sigue el camino a la derecha?</b> Indica si el camino del bote sigue hacia la derecha (es decir, no hay tiburones o se acaba el camino hacia la derecha, y el bote no viene de allí)
<b>Desembarcar</b> Desembarca en la isla. El bote debe estar en la misma ubicación que la isla.	

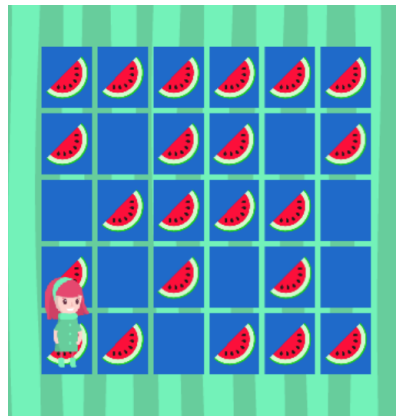
Se busca un procedimiento "Ir a la isla del sol" que lleve el bote a la isla del sol y desembarque allí.

**Pista:** Acá la parte difícil del recorrido es la de moverse. ¿Qué sucede si pongo dos alternativas consecutivas? ¿Se mueve siempre de la forma esperada? ¿Cuántas veces avanza? ¿Y si pongo una alternativa dentro de otra (separando en subtarefas adecuadamente, claro)?

#### Ejercicio 6) **María la fanática de las sandías**

María quiere volver a comer todas las sandías del escenario, que ahora tiene una nueva complejidad. Las sandías pueden estar en cualquier lugar del escenario (incluso donde está María al arranque), o bien pueden haber lugares donde no haya sandías.

Las primitivas a utilizar son las mismas del ejercicio “María y las sandías” de PilasBloques (**Mover arriba, Mover abajo, Mover a la izquierda, Mover a la derecha y Comer sandía**) a la cual se agrega el sensor **¿hay sandía acá?** que indica si hay una sandía en la ubicación donde está María. El escenario a continuación muestra un posible escenario inicial:



Cree un programa con las primitivas mencionadas que ayude a María a comer todas las sandías del escenario.

#### Ejercicio 7) **María come otras frutas**

María descubrió que además de las sandías existen otras frutas deliciosas. Estuvo probando las ciruelas y las bananas, y ahora quiere volver al ataque, comiendo todas las frutas que encuentre en el escenario.

En este caso el escenario tiene 10 columnas por 6 filas. María comienza en la esquina inferior izquierda del mismo (donde también podría haber una fruta). En cada ubicación del escenario puede haber una o más frutas, o no haber nada. Si hay fruta en una ubicación, solo hay una de ellas (es decir, una sandía, una banana o una ciruela).

En este ejercicio se suman a las primitivas que ya teníamos los comandos primitivos **Comer banana** y **Comer ciruela**, que hacen lo que su nombre sugiere, así como los sensores **¿hay banana acá?** y **¿hay ciruela acá?**

Se pide escriba un programa que haga que María coma toda la fruta del escenario. Luego responda:

- ¿Su código funciona en el caso de que el escenario contenga ubicaciones con más de una fruta?
- ¿Y en el caso de que no haya nada?
- ¿Cambiaría el código si como condición inicial sabemos que nunca hay más de una fruta?

#### Ejercicio 8) **Beelly la abeja**

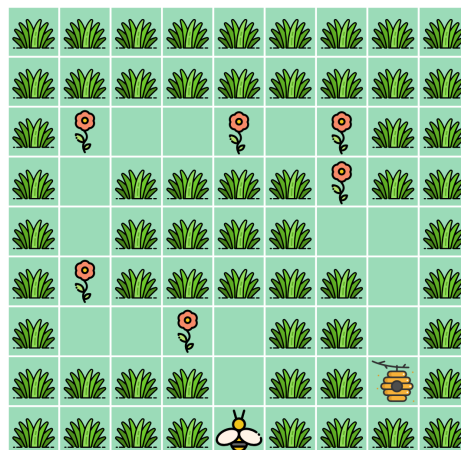
Beelly la abeja, nuestro autómatas para esta actividad, quiere llegar a su panal, recogiendo todo el polen de las flores que encuentre en el camino.

El escenario es sencillo, hay diversas ubicaciones, sectorizadas mediante una grilla cuadrículada. Hay un camino que va desde donde arranca Beelly hasta el panal, el cual es único (no hay bifurcaciones) y tiene

exactamente 21 ubicaciones (desde la ubicación donde está Beelly, incluyendo dicha ubicación, hasta la ubicación donde está el panal, inclusive). La forma del camino es aleatoria, y no sabemos qué giros deberá realizar Beelly para llegar al panal.

En las ubicaciones que no forman parte del camino, hay pasto, el cual Beelly no puede sobrevolar. Por otro lado, en las ubicaciones del camino, y de forma aleatoria, pueden haber flores (es decir, hay ubicaciones con flores y ubicaciones sin flores, sin estar claro en donde, ya que pueden variar). Incluso pueden haber flores en donde comienza Beelly, aunque seguro no hay flores sobre el panal.

A continuación se muestra un posible escenario inicial, aunque otros escenarios iniciales son posibles, siempre que se ajusten a la descripción arriba mencionada.



Lo que se pide es que realice un programa que lleve a Beelly de su ubicación actual, al panal, entrando al mismo, y recogiendo el polen de cada flor por la que pasa. Para ello se cuenta con las siguientes primitivas y sensores:

<b>Volar arriba</b> Hace que Beelly se mueva una ubicación para arriba. Falla sí hay pasto hacia arriba, o no hay más ubicaciones hacia allí.	<b>Volar abajo</b> Hace que Beelly se mueva una ubicación para abajo. Falla sí hay pasto hacia abajo, o no hay más ubicaciones hacia allí.
<b>Volar a la derecha</b> Hace que Beelly se mueva una ubicación para la derecha. Falla sí hay pasto hacia la derecha, o no hay más ubicaciones hacia allí.	<b>Volar a la izquierda</b> Hace que Beelly se mueva una ubicación para la izquierda. Falla sí hay pasto hacia la izquierda, o no hay más ubicaciones hacia allí.
<b>Recoger polen</b> Hace que Beelly recoja polen de la flor. Falla sí Beelly no está sobre una flor.	<b>Entrar al panal</b> Hace que Beelly entre al panal. Falla sí Beelly no está sobre el panal..
<b>¿continúa el camino sin pasto arriba?</b> Indica sí el camino, desde la ubicación actual de Beelly, continúa hacia arriba.	<b>¿continúa el camino sin pasto abajo?</b> Indica sí el camino, desde la ubicación actual de Beelly, continúa hacia abajo.
<b>¿continúa el camino sin pasto a la izquierda?</b> Indica sí el camino, desde la ubicación actual de Beelly, continúa hacia la izquierda.	<b>¿continúa el camino sin pasto a la derecha?</b> Indica sí el camino, desde la ubicación actual de Beelly, continúa hacia la derecha.
<b>¿hay una flor acá?</b> Indica sí hay una flor en la ubicación en donde se encuentra Beelly.	

