

Codewards

# Hora de código

Lección de 80 minutos

Plan de clase

# Hora de código

---

80 minutos

- Objetivos de la clase:**
1. Familiarizarse con la dinámica de juego en Codewards.
  2. Una rápida introducción del término 'objeto'.
  3. Familiarizarse con los conceptos básicos de la programación y las maneras de aplicarla en la vida real.
- 

- Términos:**
1. Comandos de computadora
  2. Objetos
  3. Algoritmo
  4. Optimización
  5. Repetición
  6. Ciclo
- 

**Actividades en la computadora:** Los estudiantes resolverán tareas (desafíos) de diferentes temas. Las nuevas tareas aparecerán automáticamente conforme se van completando las anteriores.

Comandos: move, rotate, load, put  
Argumentos: right, left  
Estructura: loop...end  
Objetos: robot, crane

---

**Materiales necesarios:** Computadoras con acceso al sistema de Codewards.

---

## Estructura de la clase (Versión 1)

1. Introducción de la historia.
2. Realización de las tareas + explicación de conceptos.

### Parte 1. Introducción

Tiempo aprox. – 5 minutos

Muestre el video y cuente la historia.

**Profesor:** *«¡Estamos en una misión muy importante! Tenemos que restaurar el funcionamiento del sistema operativo de la estación sumergida bajo el agua. Mientras completamos nuestros desafíos, aprenderemos a usar comandos en la computadora para poder controlar diferentes objetos.»*

### Parte 2. Prácticas en la computadora

Tiempo aprox. – 70 minutos

Completar todas las tareas en el mapa:

- Tarea 1
- Tarea 2
- Tarea 3
- Tarea 4
- Tarea 5
- Tarea 6
- Tarea 7
- Tarea 8
- Tarea 9
- Tarea 10
- Tarea 11
- Tarea 12

Mientras terminan con las tareas, dé las explicaciones necesarias a los estudiantes.

### Parte 3. Reflexión

Tiempo aprox. – 5 minutos

**Profesor:** *«Hoy hemos usado el código para reparar la cúpula y la cañería de la estación sumergida bajo el agua. Hemos hecho un gran trabajo, pero aún queda mucho por reparar. Espero que nuestro grupo se reúna de nuevo para completar más misiones.»*

## Estructura de la clase (*Versión 2*)

### Parte 0. Primer encuentro

Todos se presentan y conocen a los demás.

### Parte 1. Introducción

Tiempo aprox. – 10 minutos

Muestre el video de la historia y cuente lo más importante.

**Profesor:** «¡Estamos en una misión muy importante! Tenemos que restaurar el funcionamiento del sistema operativo de la estación sumergida bajo el agua. Mientras completamos nuestros desafíos, aprenderemos a usar comandos en la computadora para poder controlar diferentes objetos.»

### Parte 2. Prácticas en la computadora

Tiempo aprox. – 55 minutos

**Profesor:** «Vamos a abrir nuestro panel de control y veremos cómo funciona.»

- Ir a Tarea 1

Antes de empezar con la tarea, explique el sistema de comandos.

### **Concepto #1:**

---

## **Quién + Qué + Cómo**

Explique que la manera más simple de controlar objetos es con un sistema de comandos: QUIÉN + QUÉ (tenemos que hacer) + CÓMO (cuántos pasos, en qué dirección, etc.)

Practique con el juego de acción “Te programaré”

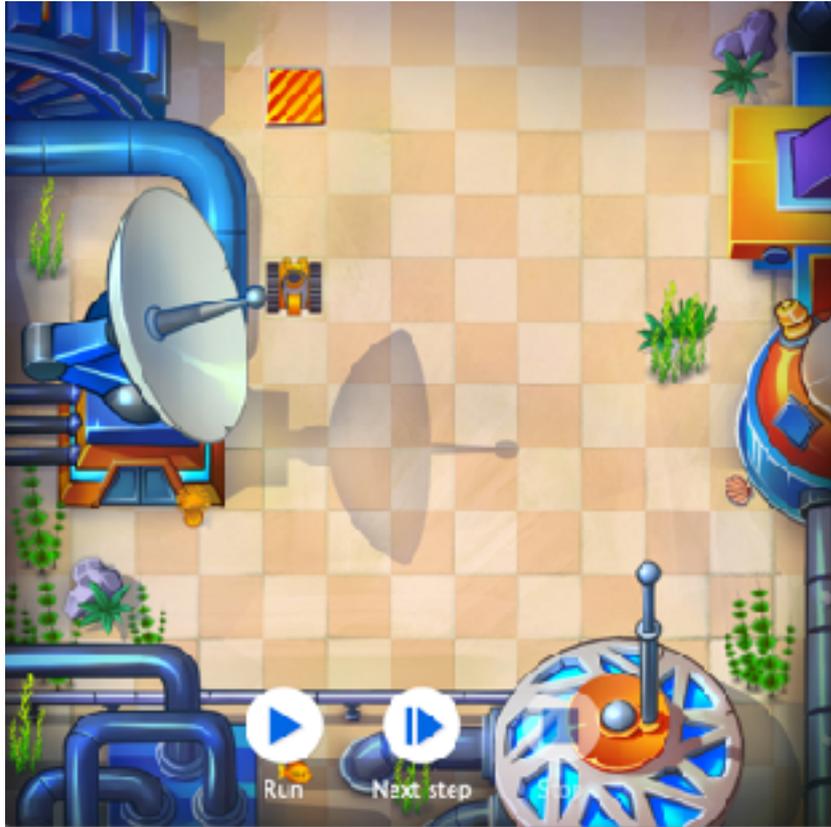
**Versión 1.** Los estudiantes deben programar al profesor usando el enfoque “QUIÉN + QUÉ + CÓMO”. Los niños dicen los comandos y el profesor los realiza (por ejemplo, el profesor gira a la derecha).

**Versión 2.** Uno por uno, cada estudiante se acerca al pizarrón con una tarjeta con un robot. Otros niños programan al robot y el profesor apunta los comandos en el pizarrón.

**Profesor:** “Vamos a tratar de hacer lo mismo en las computadoras- empiecen las Tareas 1 y 2”



## Tarea #1



**Tarea:** Avanza a la casilla señalada. Escribe el código.

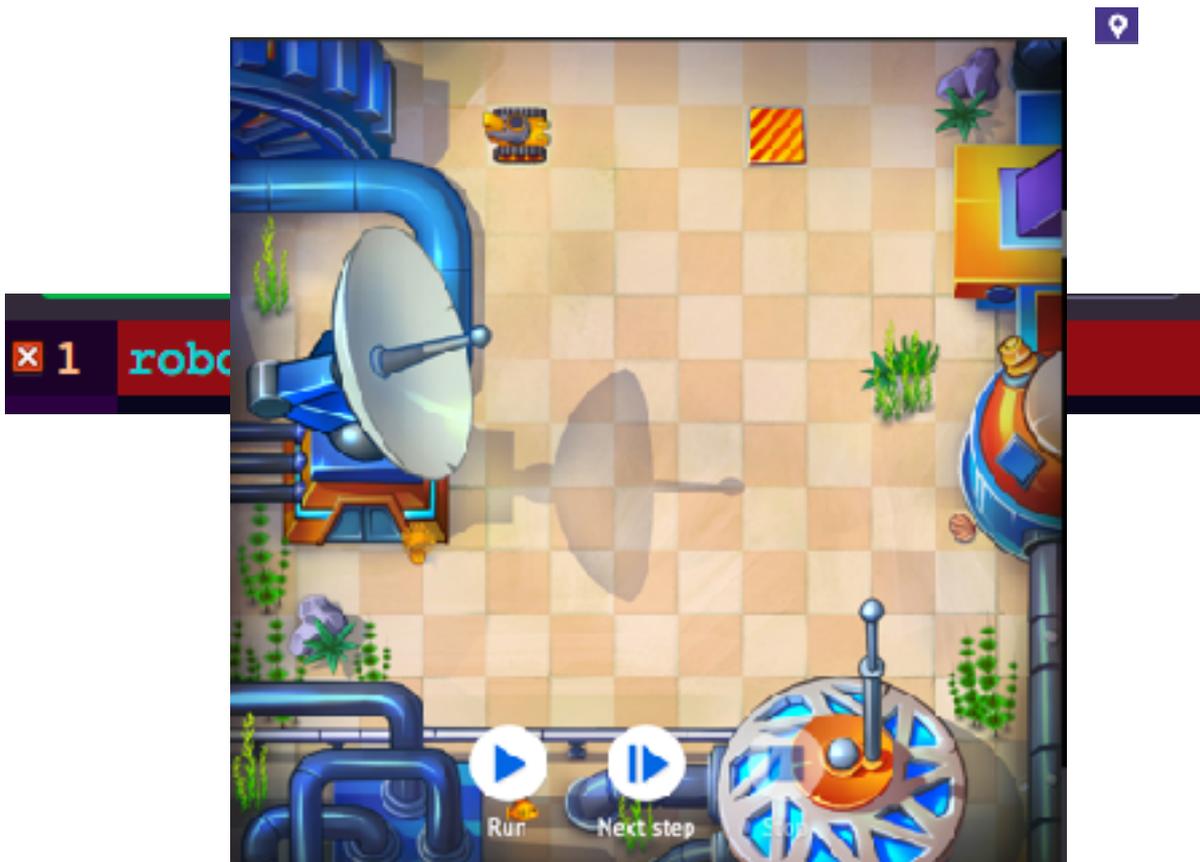
*El código no está escrito en esta tarea. Los estudiantes deben escribir el código y ejecutar el programa.*

**No hay código inicial**

**Código final:**

1. `robot.move 3`

## Tarea #2



**Tarea:** Avanza a la casilla señalada. ¡Tú sabes cómo se hace!

*El código no está escrito en esta tarea. Los estudiantes deben escribir el código y ejecutar el programa.*

**No hay código inicial**

**Código final:**

1. `robot.move 4`

## Consejos y trucos:

- Si el estudiante no puede pasar a la Tarea 2 - muestre el botón de navegación
- Si el estudiante no puede calcular la distancia, proponga un ejemplo con los pasos de una persona o los del juego de ajedrez

Cuando las tareas estén listas, muestre en la pantalla cómo resolver la Tarea 4. Al hacerlo haga deliberadamente un error en el código. La línea con el error será señalada:

Explique a los niños que nosotros no somos robots y hacemos errores bastante a menudo. En el mundo de las computadoras, los errores son nuestros amigos. Nos muestran qué es lo que necesitamos corregir para que podamos llegar a nuestro objetivo. Muestre cómo Codewards señala el error.

**Profesor:** *“Es simple cuando el objeto que controlamos sólo conoce un comando. Pero hay mecanismos mucho más complicados. Por ejemplo, una lavadora puede lavar lana usando un programa y lavar pantalones usando otro totalmente distinto. Un microondas puede calentar o descongelar la comida, etc.”*

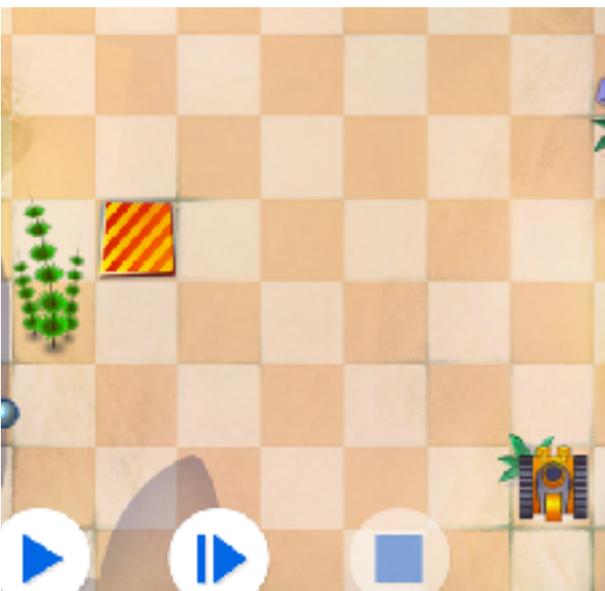
**Si un mecanismo puede realizar múltiples acciones, ¿qué “parte” del sistema de comando tendrá más de una opción: “quién”, “qué” o “cómo”?**

*La mayoría de los niños dará la respuesta correcta: “qué”. Felicítelos y muestre que tenemos dos “zonas variables” en el sistema de comandos:*

1. “cómo” (ellos vieron en tareas anteriores cómo el número de pasos del robot puede ser diferente)
2. “qué” (si el robot puede realizar varios comandos).

¿Cuántas acciones debe hacer el robot para alcanzar el objetivo?

El robot debe hacer 3 acciones realizadas por dos comandos diferentes y ejecutados uno por uno. Anteriormente usamos sólo un comando para alcanzar el objetivo.



Cuando los comandos se escriben en una secuencia con el objetivo de alcanzar la meta, a esa secuencia se le llama **algoritmo** - secuencia de comandos ejecutados uno por uno.

### ▪ Ir a Tarea 3

Vamos a aprender un nuevo comando – **rotate**

`rotate` – girar. Este comando siempre va junto a la dirección - girar a dónde - izquierda o derecha.

El comando completo tiene que ser así:

```
robot.rotate left  
robot.rotate right
```

Si quieres saber a dónde dirigir tu robot, primero hay que definir en dónde está la cabeza de tu robot, luego ponte en su lugar y piensa hacia dónde tienes que girar.

Hay que prestar atención a que antes de escribir un programa, tenemos que imaginar cómo se moverá el robot hacia su objetivo. Luego dividimos este proceso en secuencias simples que el robot reconoce. Y, finalmente, escribimos aquellos pasos en lenguaje de código.

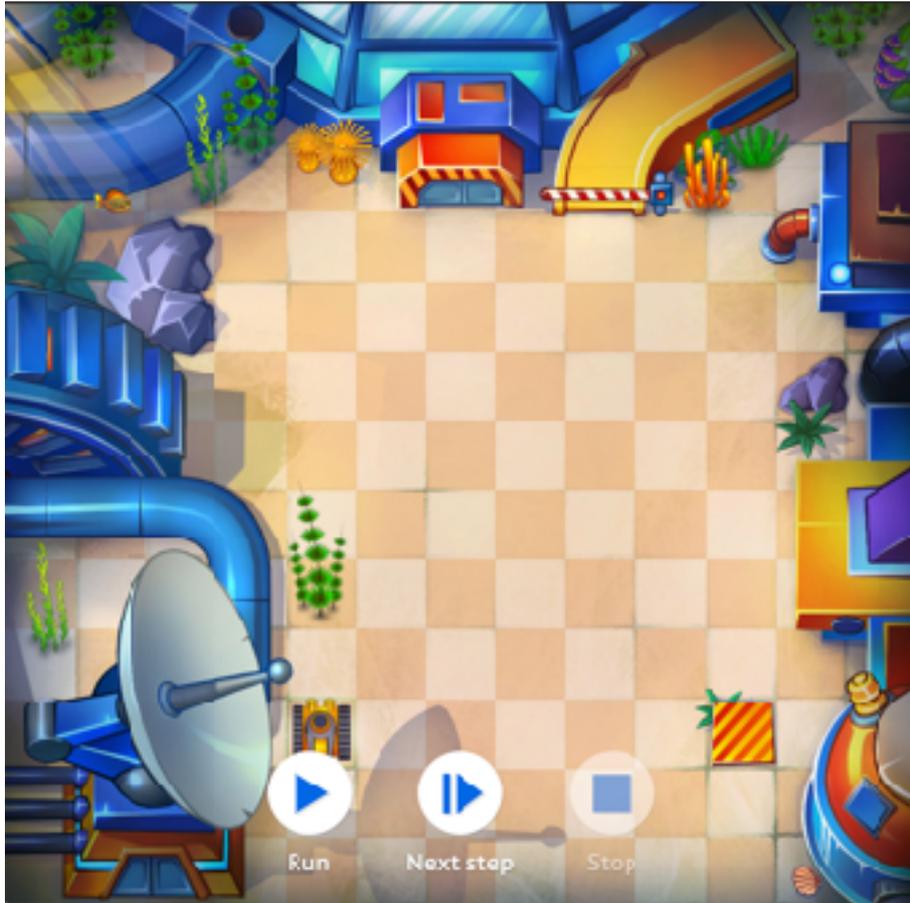
Básicamente, es lo que todos los programadores hacen - escriben algoritmos especiales (secuencias de comandos) para que los robots hagan lo que se les ordena..

Hacer Tareas #3, #4 y #5.

Para ayudar a los estudiantes con sus tareas, siga preguntando:

- ¿Cuántas acciones debe realizar el robot?
- ¿Qué comando responde a la acción?
- ¿Cómo debemos modificar el comando?

## Tarea #3



**Tarea:** Ir a la casilla señalada. Escribe el código.

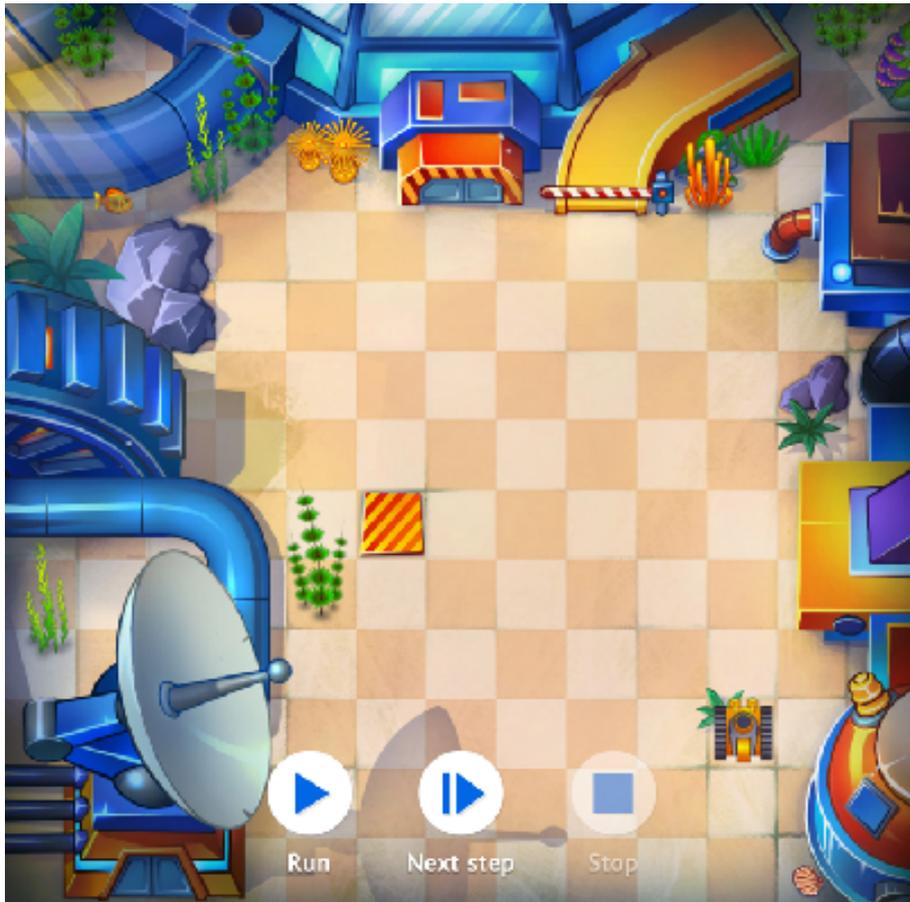
*El código no está escrito en esta tarea. Los estudiantes deben escribir el código y ejecutar el programa.*

**No hay código inicial**

**Código final:**

1. `robot.rotate right`
2. `robot.move 6`

## Tarea #4



**Tarea:** Ir a la casilla señalada. Escribe el código.

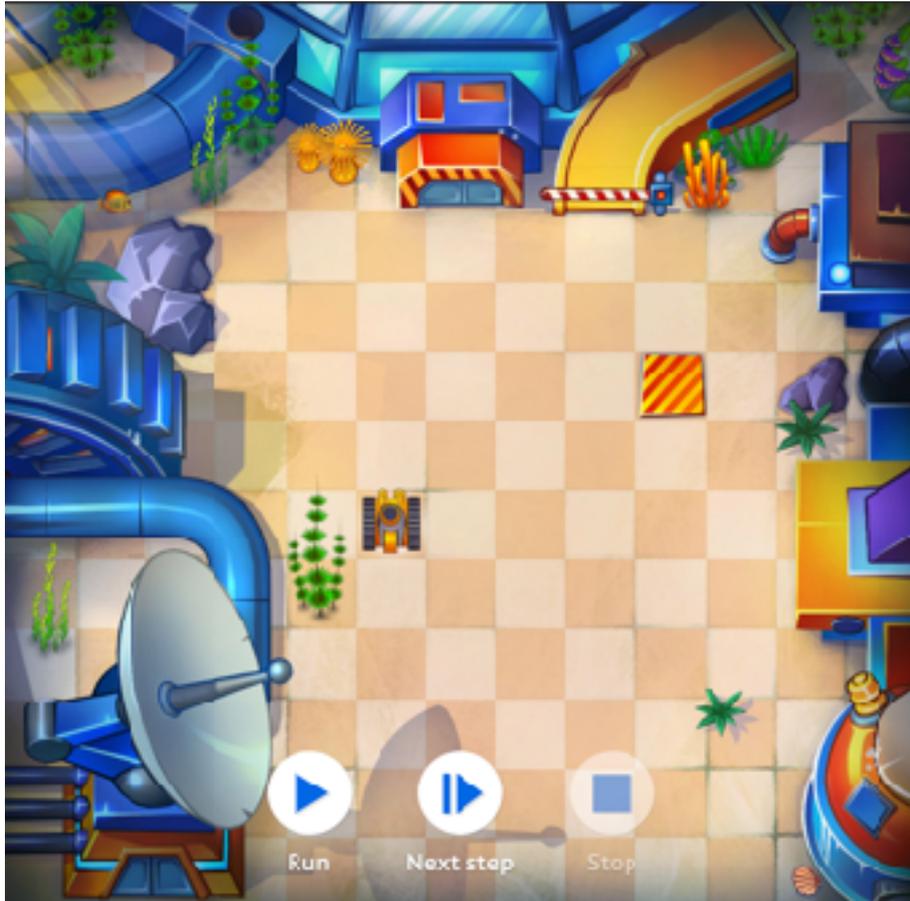
*El código no está escrito en esta tarea. Los estudiantes deben escribir el código y ejecutar el programa.*

**No hay código inicial**

**Código final:**

1. `robot.move 3`
2. `robot.rotate left`
3. `robot.move 5`

## Tarea #5



**Tarea:** Avanza a la casilla señalada. ¡Vamos!

*El código no está escrito en esta tarea. Los estudiantes deben escribir el código y ejecutar el programa.*

**No hay código inicial**

**Código final:**

1. `robot.move 2`
2. `robot.rotate right`
3. `robot.move 4`

## Consejos

- Si el estudiante ha escrito algo incorrecto y no sabe cómo borrarlo, muéstrele cómo utilizar la tecla de retroceso (Backspace) ya sea seleccionando o no un texto.
- Si el estudiante ha escrito un algoritmo que funciona pero es diferente a la solución y recibe menos de 3 estrellas, hay que explicarle que no debe preocuparse y que regresarán a esa pregunta más tarde. Es posible que su algoritmo contenga más pasos de los necesarios.
- Si el estudiante ejecuta el programa después de escribir cada paso, el sistema de Codewards indicará “Inténtalo de nuevo” - explique que antes de ejecutar el programa, hay que escribir la secuencia completa de comandos y después ejecutarlos.

### ▪ Abrir Tarea 6

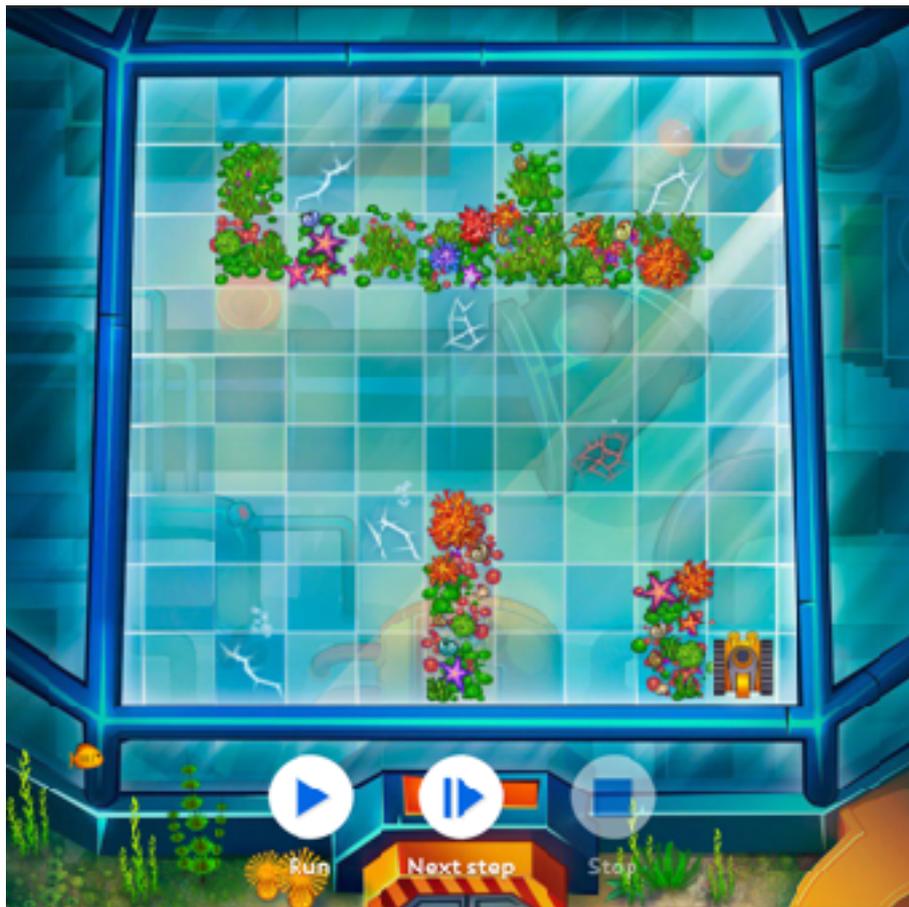
Haga las siguientes preguntas:

- ¿El robot alcanza el objetivo? - sí
- ¿Cuántas casillas se movió el robot? - 5
- ¿Cuántos comandos hay en el algoritmo? - 5
- ¿Es posible reescribir el programa para reducir el número de líneas o el número de casillas que se mueve el robot? - sí

Explique que si queremos que nuestros robots trabajen de manera eficiente, tenemos que escribir programas **óptimos**, que ahorrarán su tiempo y otros recursos.

Completar Tareas #6 y #7

## Tarea #6



**Tarea:** ¡Ejecuta el programa para reparar la grieta en la cúpula!

*En esta tarea, el código ya está escrito. Los estudiantes deben dar clic en “Run” y ejecutar el programa. Haga énfasis en cuál es la grieta a la que debe dirigirse el robot. La grieta que necesita ser reparada está señalada.*

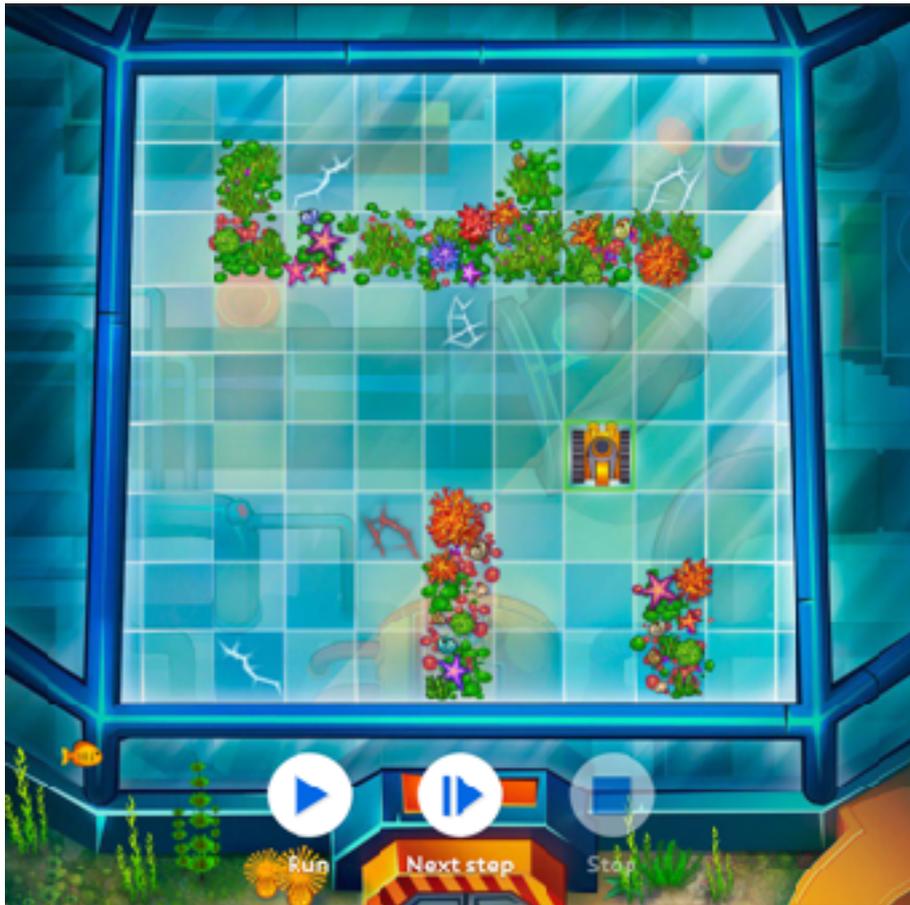
### Código inicial:

1. `robot.move 3`
2. `robot.rotate left`
3. `robot.move 2`

### Código final:

1. `robot.move 3`
2. `robot.rotate left`
3. `robot.move 2`

## Tarea #7



**Tarea:** Repara la grieta. Escribe el código que falta.

*En esta tarea, el código no está terminado. Los estudiantes deben acabar las líneas y ejecutar el programa. La grieta que necesita ser reparada está señalada.*

### Código inicial:

1. \_\_\_\_\_
2. `robot.move 3`
3. \_\_\_\_\_
4. `robot.move 1`

### Código final:

1. `robot.rotate left`
2. `robot.move 3`
3. `robot.rotate left`
4. `robot.move 1`

**Profesor:** *“Vamos a enfocarnos en qué es lo que hemos aprendido hoy”:*

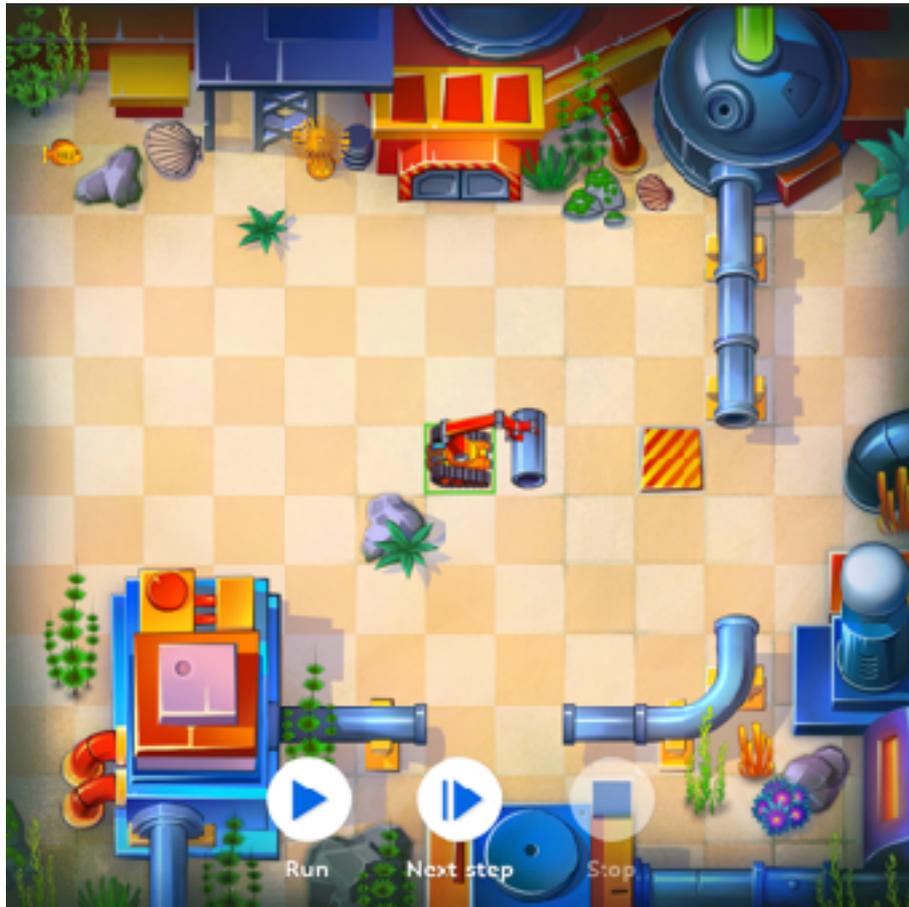
- Los mecanismos y robots son controlados por comandos con la siguiente estructura:  
QUÉ + QUIÉN + CÓMO
- Algunos mecanismos pueden ejecutar sólo un comando, pero la mayoría pueden ejecutar múltiples comandos.
- La secuencia de comandos que lleva al objetivo se llama **algoritmo**.
- El algoritmo puede ser **óptimo** o **no óptimo**. El no óptimo necesita más recursos. Debemos escribir sólo algoritmos óptimos que ahorren nuestro tiempo y energía.

**Profesor:** *“Nos queda la tarea más importante. ¡Vamos!”*

Qué es nuevo en la siguiente tarea:

1. El robot puede realizar varios comandos.
2. Nuevos comandos: **load** - toma la tubería, **put** - pone la tubería abajo.
3. Esta tarea tiene una variedad de soluciones - los niños deben encontrar la ideal.

## Tarea #8



**Tarea:** Haz que la grúa recoja una parte de la tubería y la ponga en el agujero más próximo de la cañería. Escribe el código.

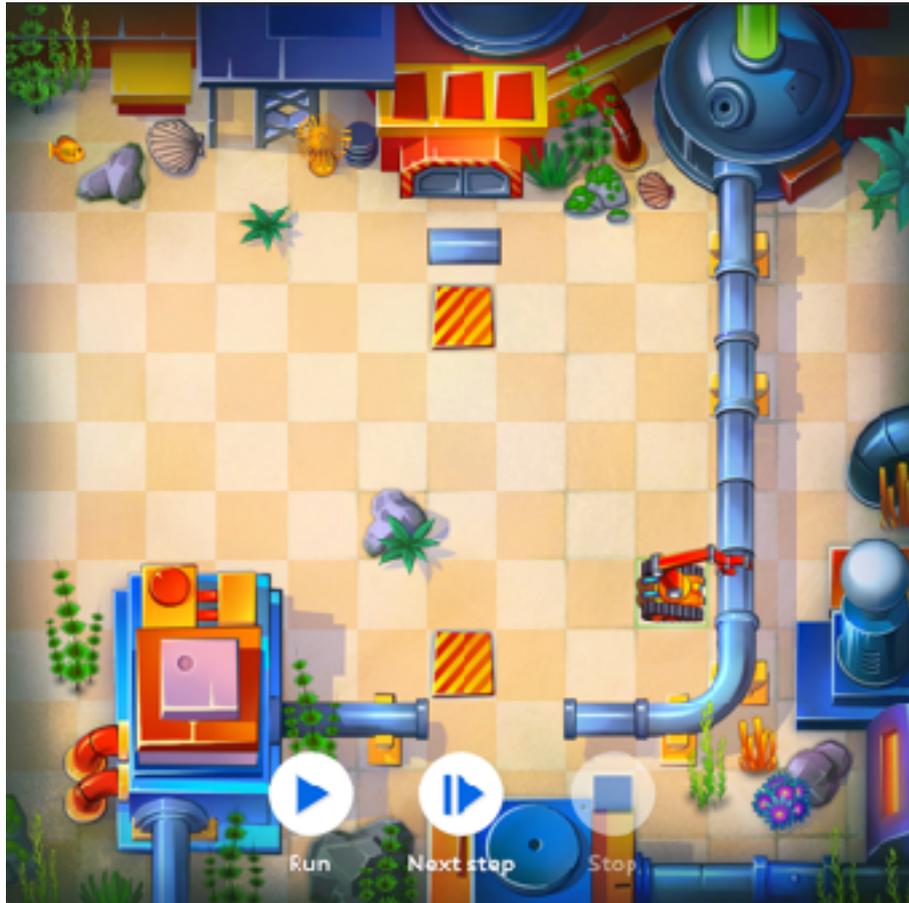
*El código no está escrito en esta tarea. Los estudiantes deben escribir el código y ejecutar el programa.*

**No hay código inicial**

**Código final:**

1. `crane.load`
2. `crane.move 3`
3. `crane.put`

## Tarea 9



**Tarea:** Toma una parte de la tubería y ponla en la rotura de la cañería señalada. Para instalar la pieza, recógela con la grúa y mueve la grúa a la casilla señalada. Escribe el código.

*El código no está escrito en esta tarea. Los estudiantes deben escribir el código y ejecutar el programa.*

**No hay código inicial**

**Código final:**

1. crane.rotate right
2. crane.rotate right
3. crane.move 3
4. crane.rotate right
5. crane.move 4
6. crane.load
7. crane.rotate right
8. crane.rotate right
9. crane.move 5
10. crane.put

**Profesor:** “¿Han notado cómo algunas cosas en nuestras vidas se repiten? Vamos a pensar en algo que se repite de vez en cuando. ¿Qué puede ser?”

**Ejemplos:** temporadas, fiestas, etc.

Aquí un extracto de la famosa canción:

*We wish you a Merry Christmas;  
We wish you a Merry Christmas;  
We wish you a Merry Christmas  
and a Happy New Year.*

Marca las líneas que se repiten.

**Profesor:** “¿Será muy aburrido escribir esas líneas una y otra vez?”

Son totalmente iguales y se repiten tres veces. ¿No sería bueno tener una herramienta que repita por ti?

Esta herramienta existe, se llama **ciclo**. Para comprender cómo funciona, vamos a jugar un juego.

El profesor se convierte en objeto y tiene que moverse a una casilla, repitiendo dos comandos 4 veces.

*teacher.move4  
teacher.turn right*

Tiene que pasar 4 veces.

**Profesor:** “Vamos a ejecutar el programa”

Profesor: “El ciclo está en funcionamiento. ¿Cuántas veces se repite?”

Estudiantes: 4

Profesor: “¿Qué debo hacer?”

Estudiantes: “Dar 4 pasos”

Profesor: “¿Algo más?”

Estudiantes: “Girar a la derecha”

El profesor alcanza la casilla.

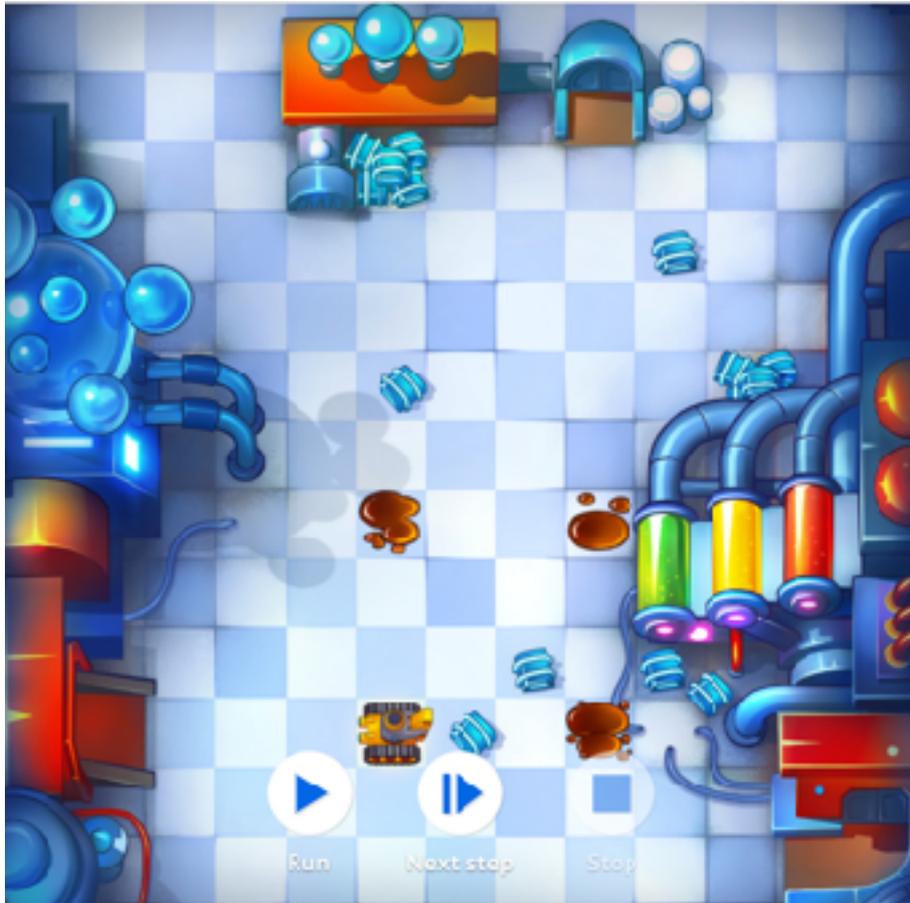
- **Ir a Tarea 10.**

**Profesor:** “Vamos a ver cómo es el código con el ciclo y qué hace el robot”

Haga énfasis en que «end command» no se ejecuta hasta que el ciclo haya terminado con el número de ciclos (“loops”) indicados. El programa procederá al último comando sólo cuando se complete el ciclo.

Tenemos que explicar que si queremos que nuestro programa haga diferencia entre dos comandos repetidos, tenemos que poner esos comandos dentro del ciclo.

## Tarea #10



**Tarea:** A veces el robot realiza las mismas acciones una y otra vez. Usamos ciclos (“loops”) para evitar escribir comandos repetidos. Vamos a ver cómo funcionan. El código ya está escrito, ejecuta el programa.

*El código ya está escrito en esta tarea. Los estudiantes deben dar clic en “Run” y ejecutar el programa.*

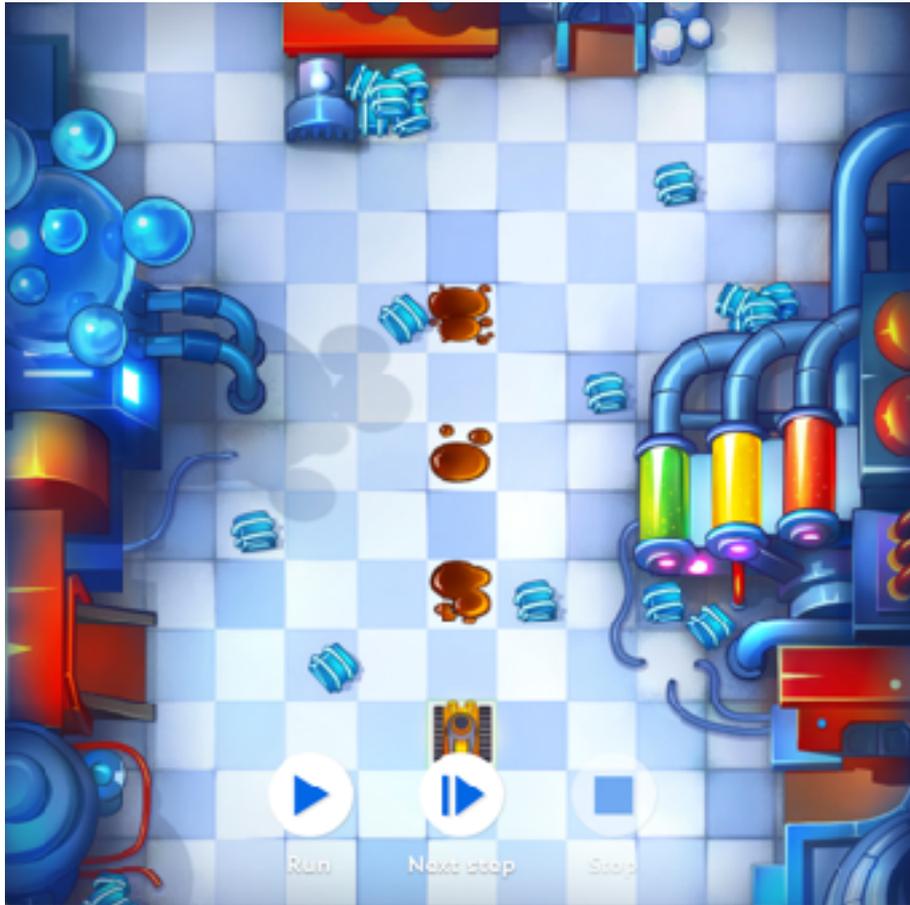
### Código inicial:

1. loop 3
2.     robot.rotate right
3.     robot.move 3
4. end

### Código final:

1. loop 3
2.     robot.rotate right
3.     robot.move 3
4. end

## Tarea #11



**Tarea:** Hay que limpiar todas las huellas de aceite. Intenta escribir el programa usando el ciclo (“loop”).

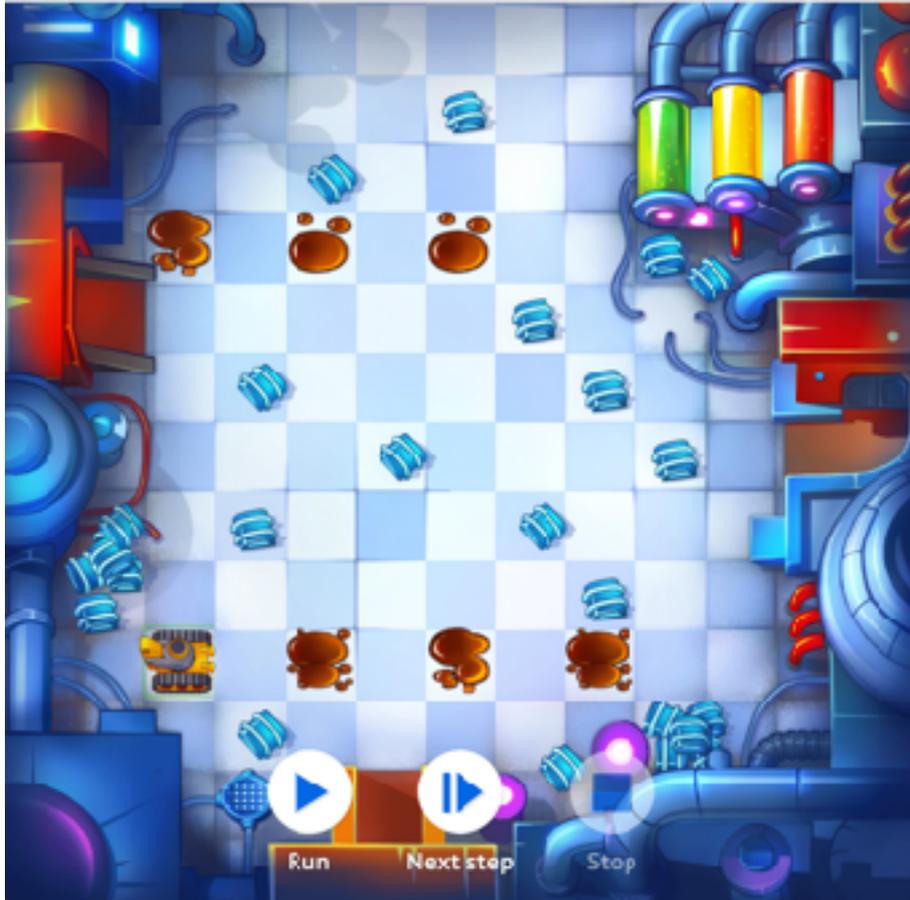
*El código no está escrito en esta tarea. Los estudiantes deben escribir el código y ejecutar el programa.*

**No hay código inicial**

**Código final:**

1. loop 3
2.     robot.move 2
3. end

## Tarea #12



**Tarea:** Hay que limpiar todas las huellas de aceite. Escribe el código.

*El código no está escrito en esta tarea. Los estudiantes deben escribir el código y ejecutar el programa.*

**No hay código inicial**

**Código final:**

```
1. loop 3
2.   robot.rotate left
3.   robot.move 6
4.   robot.rotate right
5.   robot.rotate right
6.   robot.move 6
7.   robot.rotate left
8.   robot.move 2
9. end
```

## Parte 3. Reflexión

Tiempo aprox. – 5 minutos

**Profesor:** *”Hoy hemos usado la computadora y el código para reparar la cúpula y la cañería de la estación sumergida bajo el agua. Hemos hecho un gran trabajo, pero muchas cosas siguen sin funcionar. Espero que nuestro grupo pueda cumplir grandes misiones”.*