

# Clase 01 - Introducción

IIC1103-09 - Introducción a la Programación

---

Profesor: Miguel Fadić – [mofadic@ing.puc.cl](mailto:mofadic@ing.puc.cl)

Ayudante: Geraldine Monsalve – [gnmonsalve@uc.cl](mailto:gnmonsalve@uc.cl)

Jueves 2-Marzo-2017

Departamento de Ciencia de la Computación  
Pontificia Universidad Católica de Chile

Presentación

Reglas

Algoritmo

Presentación

Reglas

Algoritmo

¡Bienvenidos!

# ¿DE QUÉ SE TRATA ESTE CURSO?

¿DE QUÉ SE TRATA ESTE CURSO?

# Aprender a programar

¿DE QUÉ SE TRATA ESTE CURSO?

# Aprender a programar



¿DE QUÉ SE TRATA ESTE CURSO?

# Aprender a programar





¿DE QUÉ SE TRATA ESTE CURSO?

# Aprender a programar



¿DE QUÉ SE TRATA ESTE CURSO?

# Aprender a programar



¿DE QUÉ SE TRATA ESTE CURSO?

# Aprender a programar



¿DE QUÉ SE TRATA ESTE CURSO?

# Aprender a programar



¿DE QUÉ SE TRATA ESTE CURSO?

# Aprender a programar



# ¿PARA QUÉ PROGRAMAR?

# ¿PARA QUÉ PROGRAMAR?

Para resolver un problema

# ¿PARA QUÉ PROGRAMAR?

## Para resolver un problema

Programar es una herramienta útil para resolver problemas



# ¿PARA QUÉ PROGRAMAR?

## Para resolver un problema

Programar es una herramienta útil para resolver problemas

¿Qué tipo de problemas?

# ¿PARA QUÉ PROGRAMAR?

## Para resolver un problema

Programar es una herramienta útil para resolver problemas

¿Qué tipo de problemas?

Cualquiera.

# ¿PARA QUÉ PROGRAMAR?

## Para resolver un problema

Programar es una herramienta útil para resolver problemas

¿Qué tipo de problemas?

Cualquiera.

- ¿Cómo llego a Plaza Italia desde San Joaquín?

# ¿PARA QUÉ PROGRAMAR?

## Para resolver un problema

Programar es una herramienta útil para resolver problemas

¿Qué tipo de problemas?

Cualquiera.

- ¿Cómo llego a Plaza Italia desde San Joaquín?
- ¿Aprobé el ramo?

# ¿PARA QUÉ PROGRAMAR?

## Para resolver un problema

Programar es una herramienta útil para resolver problemas

### ¿Qué tipo de problemas?

Cualquiera.

- ¿Cómo llego a Plaza Italia desde San Joaquín?
- ¿Aprobé el ramo?
- ¿Cómo hago un edificio que resista un terremoto?

# ¿PARA QUÉ PROGRAMAR?

## Para resolver un problema

Programar es una herramienta útil para resolver problemas

### ¿Qué tipo de problemas?

Cualquiera.

- ¿Cómo llego a Plaza Italia desde San Joaquín?
- ¿Aprobé el ramo?
- ¿Cómo hago un edificio que resista un terremoto?
- ¿Cómo gano esta partida de ajedrez?

# ¿PARA QUÉ PROGRAMAR?

## Para resolver un problema

Programar es una herramienta útil para resolver problemas

### ¿Qué tipo de problemas?

Cualquiera.

- ¿Cómo llego a Plaza Italia desde San Joaquín?
- ¿Aprobé el ramo?
- ¿Cómo hago un edificio que resista un terremoto?
- ¿Cómo gano esta partida de ajedrez?
- ¿Quién canta esta canción?

# ¿PARA QUÉ PROGRAMAR?

## Para resolver un problema

Programar es una herramienta útil para resolver problemas

### ¿Qué tipo de problemas?

Cualquiera.

- ¿Cómo llego a Plaza Italia desde San Joaquín?
- ¿Aprobé el ramo?
- ¿Cómo hago un edificio que resista un terremoto?
- ¿Cómo gano esta partida de ajedrez?
- ¿Quién canta esta canción?
- ¿Quiénes están en esta foto?



# ¿CÓMO PROGRAMAR?

Especificando una serie de pasos: un **algoritmo**

# ¿CÓMO PROGRAMAR?

Especificando una serie de pasos: un **algoritmo**

¿Algoritmo?

# ¿CÓMO PROGRAMAR?

Especificando una serie de pasos: un **algoritmo**

¿Algoritmo?

- Secuencia ordenada de instrucciones

# ¿CÓMO PROGRAMAR?

Especificando una serie de pasos: un **algoritmo**

¿Algoritmo?

- Secuencia ordenada de instrucciones
- ¿Cómo preparar un pie de limón?
  1. Preparar la masa
  2. Preparar relleno
  3. Preparar merengue
  4. Ponerlo en el horno
  5. Esperar

# ¿CÓMO PROGRAMAR?

Algoritmo para llegar a Plaza Italia

Algoritmo para llegar a Plaza Italia

1. Salir de San Joaquín
2. Tomar el metro
3. Bajarse en Baquedano

## Algoritmo para llegar a Plaza Italia

1. Salir de San Joaquín
2. Tomar el metro
  - 2.1 Sacar la tarjeta Bip!
  - 2.2 Pagar
  - 2.3 Esperar el tren
  - 2.4 Subirse al tren
3. Bajarse en Baquedano

## Algoritmo para llegar a Plaza Italia

1. Salir de San Joaquín
2. Tomar el metro
  - 2.1 Sacar la tarjeta Bip!
  - 2.2 Pagar
  - 2.3 Esperar el tren
  - 2.4 Subirse al tren
3. Bajarse en Baquedano
  - 3.1 Esperar que abran las puertas
  - 3.2 Bajar del tren
  - 3.3 Salir de la estación



# ¿CÓMO PROGRAMAR?

Usando una herramienta apropiada:

# ¿CÓMO PROGRAMAR?

Usando una herramienta apropiada: Un computador

# ¿CÓMO PROGRAMAR?

Usando una herramienta apropiada: Un computador

Muy sencillo: decirle qué queremos hacer.

# ¿CÓMO PROGRAMAR?

Usando una herramienta apropiada: Un computador

Muy sencillo: decirle qué queremos hacer.

Especificándole un algoritmo

# ¿CÓMO PROGRAMAR?

Usando una herramienta apropiada: Un computador

Muy sencillo: decirle qué queremos hacer.

Especificándole un algoritmo

¿Cómo le decimos qué queremos hacer?

# ¿CÓMO PROGRAMAR?

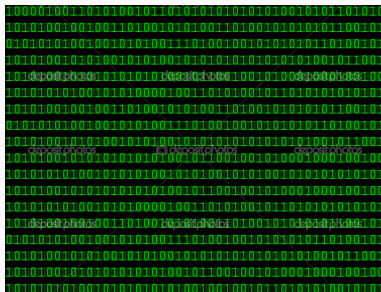
Usando una herramienta apropiada: Un computador

Muy sencillo: decirle qué queremos hacer.

Especificándole un algoritmo

¿Cómo le decimos qué queremos hacer?

Hablando en su lenguaje: Lenguaje binario



# ¿CÓMO PROGRAMAR?

En general no escribimos programas de esta manera.

- Código binario es un lenguaje de **bajo nivel**
- Es más fácil para un humano escribir en un lenguaje más cercano al lenguaje natural: **lenguaje de alto nivel**
- Lamentablemente los computadores aún no son capaces de entender lenguaje natural (aún).



# ¿CÓMO PROGRAMAR?

Pero podemos llegar a un punto intermedio:

## Lenguajes de programación

Comprensibles por el computador, y por el humano

```
File Edit Watcher MediaTools JES Functions
1 # Make a World and A Turtle
2 earth = makeWorld(400, 400)
3 turtle = makeTurtle(earth)
4
5 #Make a Square
6 def makeSquare(turtle, size):
7     for i in range(4):
8         turtle.forward(size)
9         turtle.turn(90)
10
11 makeSquare(turtle, 100)
12
13 #Make a Spiral Square
14 def makeSpiralSquare(turtle, size):
15     for i in range(18):
16         makeSquare(turtle, size)
17         turtle.turn(20)
18
19 makeSpiralSquare(turtle, 100)
20
21
```





## ¿CÓMO PROGRAMAR?

## ¿En qué lenguaje?



# ¿CÓMO PROGRAMAR?

¿En qué lenguaje?



# ¿PYTHON?

¿Por qué python?

C++:

```
1 #include <iostream>
2 int main() {
3     std::cout << "Hello World!" << std::endl;
4     std::cin.get();
5     return 0;
6 }
```

# ¿PYTHON?

¿Por qué python?

Java:

```
1 public class HelloWorld {  
2     public static void main(String[] args) {  
3         System.out.println("Hello World!");  
4     }  
5 }
```

# ¿PYTHON?

¿Por qué python?

Python:

```
1 print("Hello world!")
```

Presentación

Reglas

Algoritmo

**Cátedra:** Martes y Jueves, módulo 1 (B18)

**Laboratorio:** Lunes módulos 5 y 6 (pueden ir el día que quieran).

## Evaluaciones:

- 3 Tareas de Programación
- 2 Interrogaciones
  - I1: Lunes 24 de abril
  - I2: Martes 23 de mayo
- 1 Nota de Participación (Explicada más adelante)
- 1 Examen: Sábado 24 de junio

**Cátedra:** Martes y Jueves, módulo 1 (B18)

**Laboratorio:** Lunes módulos 5 y 6 (pueden ir el día que quieran).

## Evaluaciones:

- 3 Tareas de Programación
- 2 Interrogaciones
  - I1: Lunes 24 de abril
  - I2: Martes 23 de mayo
- 1 Nota de Participación (Explicada más adelante)
- 1 Examen: Sábado 24 de junio



**Cátedra:** Martes y Jueves, módulo 1 (B18)

**Laboratorio:** Lunes módulos 5 y 6 (pueden ir el día que quieran).

- **¡Este es un curso práctico!**
  - Aprender haciendo

## **Evaluaciones:**

- 3 Tareas de Programación
- 2 Interrogaciones
  - I1: Lunes 24 de abril
  - I2: Martes 23 de mayo
- 1 Nota de Participación (Explicada más adelante)
- 1 Examen: Sábado 24 de junio

## Evaluaciones:

- 3 Tareas de Programación (T1, T2, T3)
- 2 Interrogaciones
  - I1: Lunes 24 de abril
  - I2: Martes 23 de mayo
- 1 Nota de Participación (NP)
- 1 Examen: Sábado 24 de junio (E)

¿Y la nota final?

$$NF = 0.15 \cdot I1 + 0.15 \cdot I2 + 0.3 \cdot E + 0.1 \cdot T1 + 0.1 \cdot T2 + 0.1 \cdot T3 + 0.1 \cdot NP$$

No, el examen no es reprobatorio

¿Cómo apruebo?

$NF = 0.15 \cdot I1 + 0.15 \cdot I2 + 0.3 \cdot EX + 0.1 \cdot T1 + 0.1 \cdot T2 + 0.1 \cdot T3 + 0.1 \cdot NP$   
Promedio ponderado de evaluaciones escritas mayor o igual a 4.0

$$(0.15 \cdot I1 + 0.15 \cdot I2 + 0.3 \cdot EX) / 0.6 \geq 4.0$$

Y

Promedio de tareas mayor o igual a 4.0

$$(T1 + T2 + T3) / 3 \geq 4.0$$

## Bibliografía general:

1. Python software foundation, Python v3 Documentation, <http://docs.python.org/3/>.
2. V. Ceder. The quick python book. Manning Publications Co., 2010.
3. A. B. Downey. Think Python: How to think like a computer scientist. Green Tea Press, 2013
4. J. M. Zelle. Python programming: An introduction to computer science. Franklin, Beedle & Associates, Inc., 2<sup>nd</sup> edition, 2010.
5. Py-Libre, Apunte interactivo para el curso Introducción a la Programación, <http://runest.ing.puc.cl>

No cometan faltas

Presentación

Reglas

Algoritmo

## Definición

Un **algoritmo** es una **secuencia** ordenada de **pasos** que sirven para **resolver** un problema.

### **Problema:**

Resolver el laberinto.

### **Herramientas:**

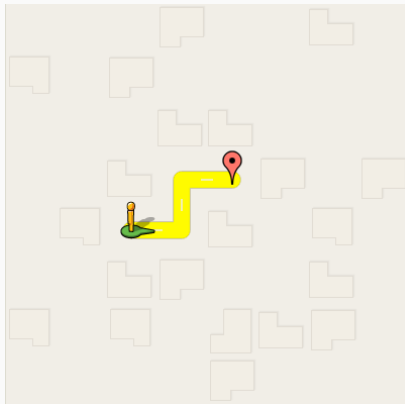
Avanzar 1 paso

Girar izquierda

Girar derecha



# EJEMPLO

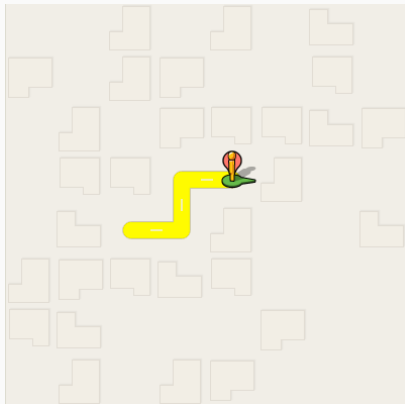


move forward

turn left ↶

turn right ↷

# EJEMPLO



move forward

turn left ↶

turn right ↷

move forward

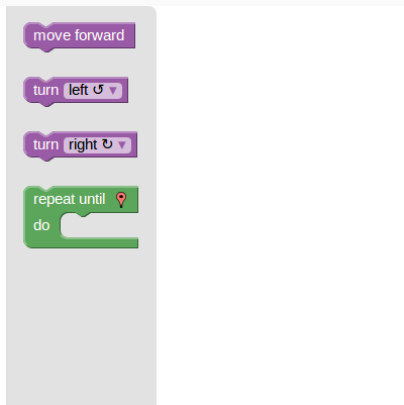
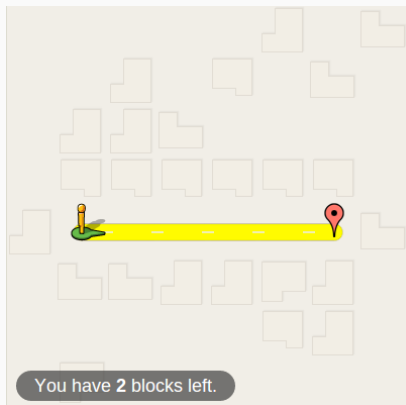
turn left ↶

move forward

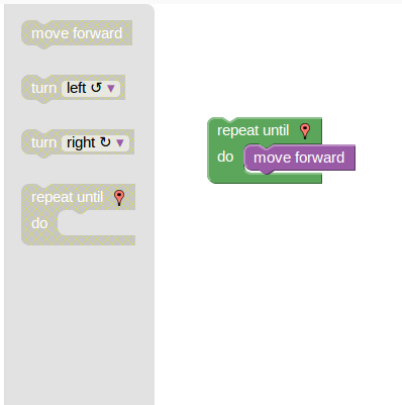
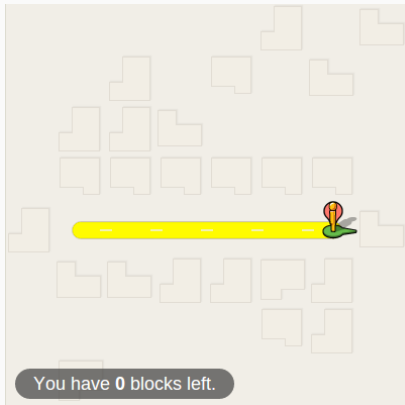
turn right ↷

move forward

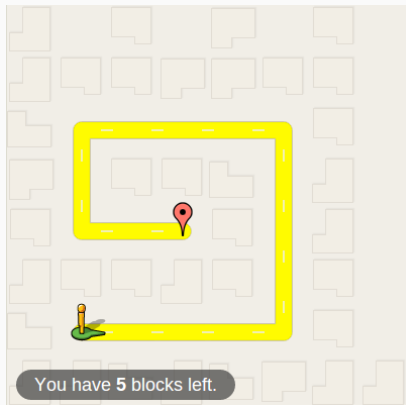
# EJEMPLO



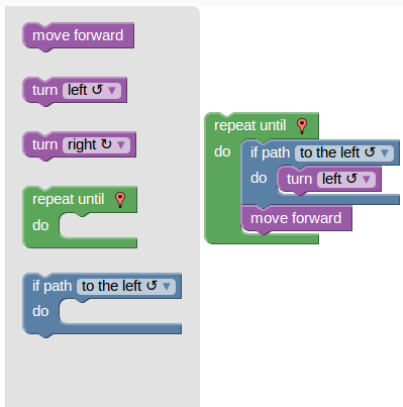
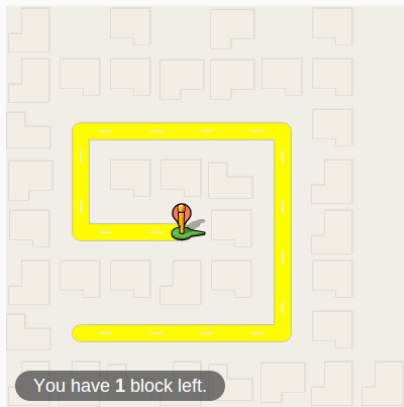
# EJEMPLO



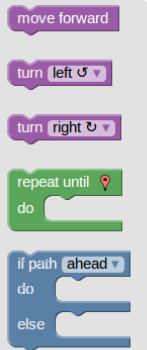
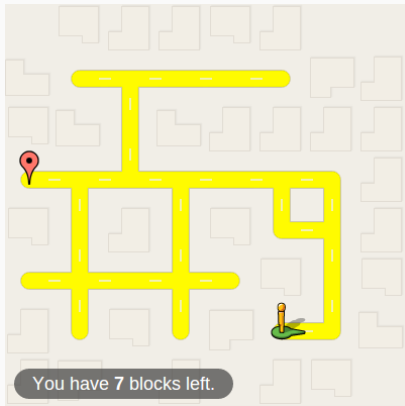
# EJEMPLO



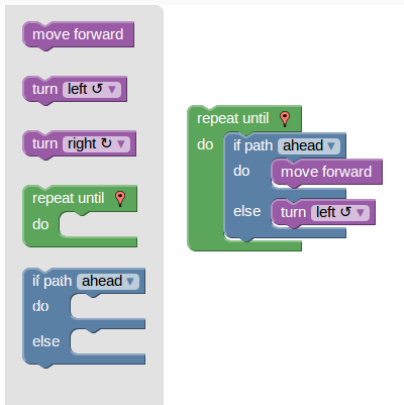
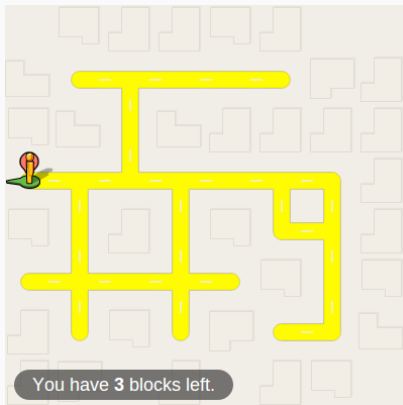
# EJEMPLO



## EJEMPLO



# EJEMPLO





## ¡ACTIVIDAD!

Ingresa a Blockly **aquí**.

<https://blockly-games.appspot.com/>

Resuelve los juegos en orden:

1. Puzzle.
2. Maze.
3. Bird.
4. Turtle.
5. Movie.
6. Pond Tutor.
7. Pond.
8. Pond Javascript.