



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencias de la Computación

Clase 01: Introducción

Rodrigo Toro Icarte (rntoro@uc.cl)

IIC1103 Introducción a la Programación - Sección 5

04 de Marzo, 2015

Bienvenidos!

Bienvenidos!

¿De dónde son?

Bienvenidos!

¿De dónde son?

¿Sistema operativo?

Bienvenidos!

¿De dónde son?

¿Sistema operativo?

¿Alguien que quiera dedicarse a computación?

Un poco sobre mí...

Un poco sobre mí...



Un poco sobre ustedes...

Un poco sobre ustedes...



Encuesta IIC1103 sección 5

La idea de esta encuesta es conocerlos un poco antes de que comiencen las clases. Esto me ayudará a preparar mejores diapositivas y actividades para el curso. Respondan en forma honesta, prometo que ninguna de sus respuestas serán usadas en su contra.

* Required

Número de alumno *

¿Por qué tomaste esta sección? *

- Me acomodaba el horario
- Me recomendaron al profe :)
- Fue un llamado del destino
- De tñ, marin, dedo, pingüé (?)
- Other:

¿Es la primera vez que tomas el ramo? *

- Sí
- No, lo tomé al año pasado pero lo boté
- No, lo tomé al año pasado pero me lo eché
- No, lo he tomado varias veces pero siempre lo boto o me lo echo u.u

¿Cuáles son tus expectativas respecto al curso? *

- Romperla! pasarlo con el medio promedio
- Aprender a programar
- Luchar por pasar el ramo (y si me lo echo, por último vender cara la derrota)
- Esforzarme lo justo para pasar
- Cachar qué onda, y si se ve muy difícil lo boto
- Other:

Un poco sobre ustedes...

¿Por qué tomaste esta sección?

Un poco sobre ustedes...

¿Por qué tomaste esta sección?

- **43%** Me acomodaba el horario
- **33%** Me recomendaron al profe :)
- **15%** Fue un llamado del destino
- **3%** De tin, marín, dedo, pingüé
- **5%** Otro

Un poco sobre ustedes...

¿Es la primera vez que tomas el ramo?

Un poco sobre ustedes...

¿Es la primera vez que tomas el ramo?

- **78%** Sí
- **7%** No, lo tomé al año pasado pero lo boté
- **15%** No, lo tomé al año pasado pero me lo eché
- **0%** No, lo he tomado varias veces pero siempre lo boto o me lo echo u.u

¿Qué veremos?

Aprenderemos a programar.

¿Qué veremos?

Aprenderemos a programar.



¿Qué veremos?

Programar: Herramienta útil (y rentable) para solucionar un problema.

¿Qué veremos?

Programar: Herramienta útil (y rentable) para solucionar un problema.



La voz de la experiencia...

La voz de la experiencia...

“Si bien el ramo lo debería haber tomado un par de años atrás, ahora lo veo su real importancia, ya que otros ramos que tome ahora (Mecánica de Sólidos) me piden hacer tareas en Python, espero lograr aprender bien, y lograr un buen promedio en este curso.”

¿Qué veremos?

¿Cómo se programa?:

```
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
```



¿Qué veremos?

Lenguaje:

¿Qué veremos?

Lenguaje:

- **53%** Nunca he programado
- **15%** Matlab
- **27%** Python
- **3%** Java
- **3%** C/C++
- **0%** C#
- **7%** Otro

¿Qué veremos?

Lenguaje:



¿Por qué 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 }
```

¿Por qué Python?

Java:

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

¿Por qué Python?

Python:

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

Reglas

Cátedra: Lunes y Miércoles módulo 1 (N7)

Laboratorios: Martes módulos 5 y 6

Reglas

Cátedra: Lunes y Miércoles módulo 1 (N7)

Laboratorios: Martes módulos 5 y 6

Evaluaciones:

- 1 3 Tareas.
- 2 3 Controles.
- 3 Midterm.
- 4 Examen.

Reglas

IIC1103 tiene un problema histórico...

Reglas

IIC1103 tiene un problema histórico... es un curso práctico.

Reglas

IIC1103 tiene un problema histórico... es un curso práctico.

La mejor evaluación es mediante tareas.

Reglas

IIC1103 tiene un problema histórico... es un curso práctico.

La mejor evaluación es mediante tareas.

Problema: La copia y compra de tareas.

Reglas

Consecuencias:

- Evaluar programar en papel.
- Examen reprobatorio.
- Controles asociados a tareas.
- Laboratorios evaluados.
- Sistemas detectores de copias.
- etc...

Reglas

Consecuencias:

- Evaluar programar en papel.
- Examen reprobatorio.
- Controles asociados a tareas.
- Laboratorios evaluados.
- Sistemas detectores de copias.
- etc...

Hemos probado todas las combinaciones...

Reglas

Consecuencias:

- Evaluar programar en papel.
- Examen reprobatorio.
- Controles asociados a tareas.
- Laboratorios evaluados.
- Sistemas detectores de copias.
- etc...

Hemos probado todas las combinaciones... pero ninguna ha funcionado u.u

Reglas

Sin embargo, existe una combinación que nadie se a atrevido a intentar...

Reglas

Sin embargo, existe una combinación que nadie se a atrevido a intentar...

$$NF = 0,3 \cdot C + 0,3 \cdot mt + 0,4 \cdot E$$

Donde:

- C : Promedio controles.
- mt : Nota midterm.
- E : Nota examen.

Reglas

Controles:

- Asociados a cada tarea.
- Serán en computador.

Reglas

Controles:

- Asociados a cada tarea.
- Serán en computador.

Examen y controles **reprobatorio**: Si $E < 4,0$ o $C < 4,0$:

$$NF = \min(0,3 \cdot C + 0,3 \cdot mt + 0,4 \cdot E; 3,9)$$

Reglas

Laboratorios: Bonus de asistencia.

- 5 décimas en promedio de controles.
- Pueden faltar hasta a 2 laboratorios.
- Es su responsabilidad asegurar que queden presentes.

Reglas

Laboratorios: Bonus de asistencia.

- 5 décimas en promedio de controles.
- Pueden faltar hasta a 2 laboratorios.
- Es su responsabilidad asegurar que queden presentes.

Inasistencias:

- Deben ser justificadas por pregrado.
- Control recuperativo.
- Examen reemplaza al midterm.

Reglas

Fechas:

- Control 1 → Sábado 11 de abril
- Control 2 → Sábado 9 de mayo
- Midterm → Sábado 16 de mayo
- Control 3 → Sábado 6 de junio
- Examen → Sábado 20 de junio

Reglas

Bibliografía general:

- ① V. Ceder. *The quick python book*.
- ② A. B. Downey. *Think Python: How to think like a computer scientist*.
- ③ J. M. Zelle. *Python programming: An introduction to computer science*.

Reglas

Bibliografía general:

- 1 V. Ceder. *The quick python book*.
- 2 A. B. Downey. *Think Python: How to think like a computer scientist*.
- 3 J. M. Zelle. *Python programming: An introduction to computer science*.

Mi bibliografía:

- 1 Clases.
- 2 *Python v3 Documentation*, <http://docs.python.org/3/>.

Otros comentarios

Otros comentarios

“Quiero aprender y dar todo por sacar el ramo de la mejor manera, me han contado que es bastante difícil por lo que me esforzaré, espero no sea tan terrible cómo me han dicho”

Otros comentarios

“Quiero aprender y dar todo por sacar el ramo de la mejor manera, me han contado que es bastante difícil por lo que me esforzaré, espero no sea tan terrible cómo me han dicho”

“Personalmente, este curso no lo tomé antes porque me genera mucho miedo reprobalo, soy nula total con lo que respecta a los computadores por ende le tengo muuuucho respeto al ramo u.u . Espero poder dar lo mejor de mí y aprender :)”

Expectativas

¿Cuáles son tus expectativas respecto al curso?

- **20%** Romperla! pasarlo con el medio promedio
- **57%** Aprender a programar
- **18%** Luchar por pasar el ramo (y si me lo echo, por último vender cara la derrota)
- **0%** Esforzarme lo justo para pasar
- **0%** Cachar qué onda, y si se ve muy difícil lo boto
- **5%** Otro

Autoeficacia

¿Crees que tienes talento para programar?

- **10%** Sí
- **22%** No
- **47%** No lo sabré hasta que lo intente
- **22%** No estoy seguro

Otro comentario

Otro comentario

“Bueno, yo soy nula para la computación, además no me gusta mucho, se usar Word y PowerPoint jajajaj yo llegaba el semestre pasado a las clases de intro a la progra y no me eran difíciles los ejemplos que hacíamos en clases... el profe los explicaba bien, yo los entendía y se me podían ocurrir solos.. pero en los laboratorios y en tareas para mí era IMPOSIBLE hacerlos por mi cuenta, ya que la complejidad de las clases era mucho menor.. Si bien la ingeniería viene de “ingenio”, y la universidad me da las herramientas para que pueda resolver problemas sola en distintos ámbitos, sentí que realmente en clases no me dieron las herramientas para poder hacer por mi cuenta una tarea como el del juego Threes o el Whatsapp... Si algún compañero ‘seco’ no me ayudaba, podía entregar la tarea en blanco... A varios de mis compañeros les pasó esto.. Vale decir que iba a todos los laboratorios y que aún así seguía teniendo los mismos problemas...”

Volviendo al curso...

Este es un curso fácil y entretenido.

Volviendo al curso...

Este es un curso fácil y entretenido.

... entonces ¿Por qué tanta gente reprueba?

Metáfora

En clases: Les daré herramientas (las justas y necesarias).



Metáfora

En laboratorios: Verán usos prácticos para las herramientas.



Metáfora

Tarea 1: Construya una casa.



Metáfora

Tarea 2: Construya un rascacielos.



Metáfora

Tipos de alumnos:

- ① El despreocupado.

Metáfora

Tipos de alumnos:

- 1 El despreocupado. PPR: 0%

Metáfora

Tipos de alumnos:

- 1 El despreocupado. PPR: 0%
- 2 El ayudantado.

Metáfora

Tipos de alumnos:

- 1 El despreocupado. PPR: 0%
- 2 El ayudantado. PPR: 5%

Metáfora

Tipos de alumnos:

- 1 El despreocupado. PPR: **0%**
- 2 El ayudantado. PPR: **5%**
- 3 El ilegal.

Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile deben mantener un comportamiento acorde a la Declaración de Principios de la Universidad. (...) Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. (...) Si un alumno copia un trabajo, obtendrá nota final 1.1 en el curso (...) Por copia se entiende incluir en el trabajo presentado como propio partes hechas por otra persona. (...)

Metáfora

Tipos de alumnos:

- ① El despreocupado. PPR: **0%**
- ② El ayudantado. PPR: **5%**
- ③ El ilegal. PPR: **20%**

Metáfora

Tipos de alumnos:

- 1 El despreocupado. PPR: **0%**
- 2 El ayudantado. PPR: **5%**
- 3 El ilegal. PPR: **20%**
- 4 El analista.

Metáfora

Desglose: Si quiero una casa, primero necesito una pared.

Metáfora

Desglose: Si quiero una casa, primero necesito una pared.



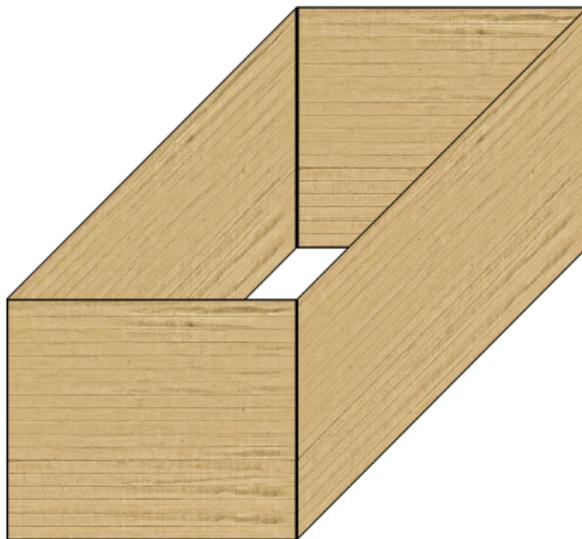
Metáfora

Desglose: Si quiero una casa, primero necesito una pared.



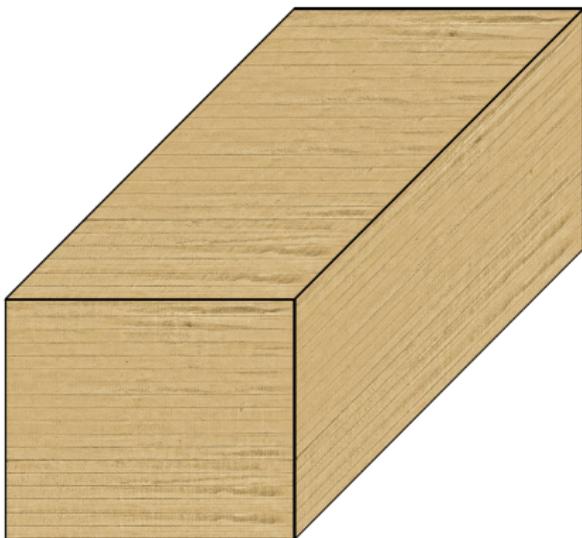
Metáfora

Desglose: Unir 4 paredes.



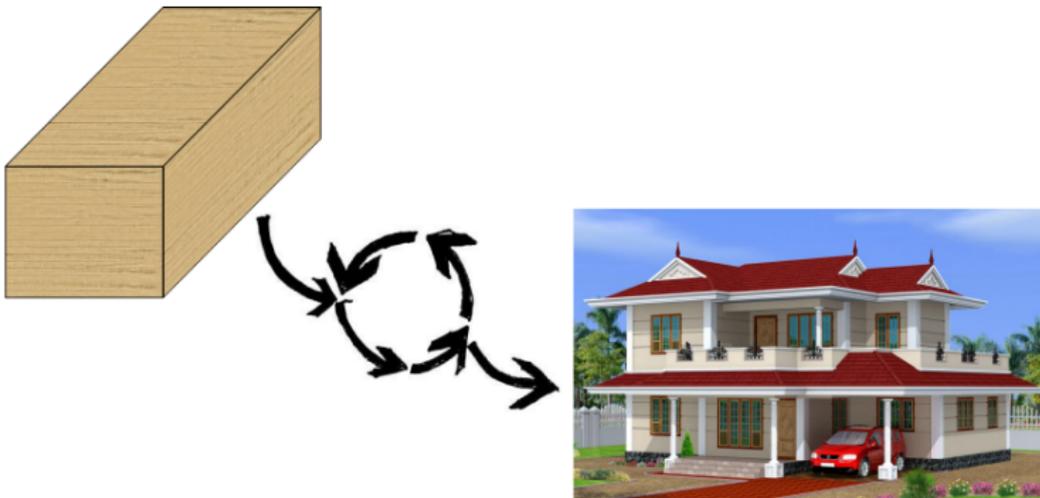
Metáfora

Desglose: Otra pared como techo.



Metáfora

Iterar:



Metáfora

Tipos de alumnos:

- ① El despreocupado. PPR: **0%**
- ② El ayudantado. PPR: **5%**
- ③ El ilegal. PPR: **20%**
- ④ El analista. PPR: **99%**

Metáfora

¿Por qué tanta gente reprueba?

Metáfora

¿Por qué tanta gente reprueba?

- IIC1103 es engañoso (*poca materia, mucha práctica*).

Metáfora

¿Por qué tanta gente reprueba?

- IIC1103 es engañoso (*poca materia, mucha práctica*).
- IIC1103 es acumulativo (*sin casa no hay rascacielos*).

Metáfora

¿Por qué tanta gente reprueba?

- IIC1103 es engañoso (*poca materia, mucha práctica*).
- IIC1103 es acumulativo (*sin casa no hay rascacielos*).
- Programar parece fácil (*Otra cosa es con guitarra*).

Metáfora

¿Por qué tanta gente reprueba?

- IIC1103 es engañoso (*poca materia, mucha práctica*).
- IIC1103 es acumulativo (*sin casa no hay rascacielos*).
- Programar parece fácil (*Otra cosa es con guitarra*).
- Hay que programar en papel.

Metáfora

¿Por qué tanta gente reprueba?

- IIC1103 es engañoso (*poca materia, mucha práctica*).
- IIC1103 es acumulativo (*sin casa no hay rascacielos*).
- Programar parece fácil (*Otra cosa es con guitarra*).
- Hay que programar en papel.

La clave: Práctica.

Metáfora

¿Por qué tanta gente reprueba?

- IIC1103 es engañoso (*poca materia, mucha práctica*).
- IIC1103 es acumulativo (*sin casa no hay rascacielos*).
- Programar parece fácil (*Otra cosa es con guitarra*).
- Hay que programar en papel.

La clave: Práctica.

El objetivo: Aprender a desglosar el problema (importante para la vida).

Metáfora

Consejos:

Metáfora

Consejos:

- Vengan a clases **siempre**.

Metáfora

Consejos:

- Vengan a clases **siempre**.
- Veán rápido la materia y comiencen a programar.

Metáfora

Consejos:

- Vengan a clases **siempre**.
- Veán rápido la materia y comiencen a programar.
- Hagan las tareas con tiempo.

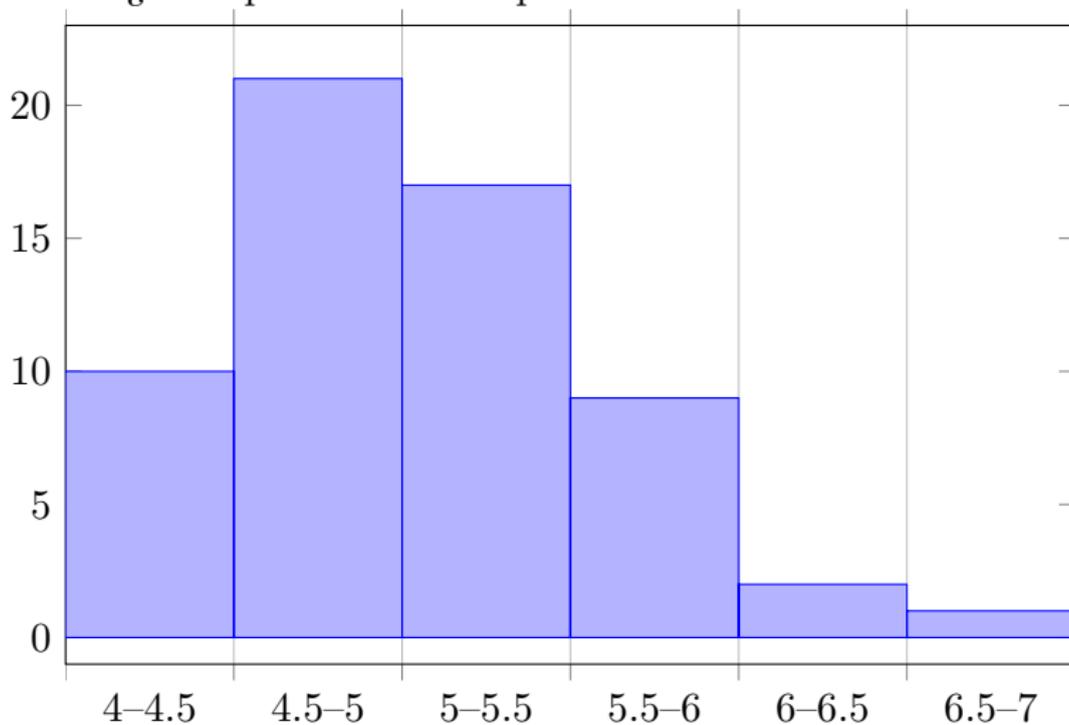
Metáfora

Consejos:

- Vengan a clases **siempre**.
- Veán rápido la materia y comiencen a programar.
- Hagan las tareas con tiempo.
- No apunten al 4.

Metáfora

¿Con qué nota crees que terminarás el curso?



Metáfora

Consejos:

- Vengan a clases **siempre**.
- Veán rápido la materia y comiencen a programar.
- Hagan las tareas con tiempo.
- No apunten al 4.
- Ensayen programar en papel.

Metáfora

Consejos:

- Vengan a clases **siempre**.
- Veán rápido la materia y comiencen a programar.
- Hagan las tareas con tiempo.
- No apunten al 4.
- Ensayen programar en papel.
- **No** usen Google.

Metáfora

Consejos:

- Vengan a clases **siempre**.
- Veán rápido la materia y comiencen a programar.
- Hagan las tareas con tiempo.
- No apunten al 4.
- Ensayen programar en papel.
- **No** usen Google.

Mi rol:

- Ayudarles a que aprendan a programar.

Metáfora

¿Qué factores crees que influyeron en que no lograras pasar el ramo?

Metáfora

¿Qué factores crees que influyeron en que no lograras pasar el ramo?

- **12%** No programé lo suficiente
- **10%** Un semestre muy pesado
- **10%** Faltaban ayudantías
- **8%** Las tareas eran imposibles
- **8%** Faltaba discutir más ejemplos en clase
- **7%** No tengo talento para programar :/
- **7%** No estudié lo suficiente
- **7%** Debí haber aprovechado mejor los laboratorios
- (...)

Estilo Sección 5

“ojalá en las clases veamos muchos ejemplos de situaciones para programar, porque lo que me pasó el semestre pasado era que en clases prestaba atención pero después veía las tareas y no entendía nada en verdad...”

Estilo Sección 5

“ojalá en las clases veamos muchos ejemplos de situaciones para programar, porque lo que me pasó el semestre pasado era que en clases prestaba atención pero después veía las tareas y no entendía nada en verdad...”

“Espero que no sean tan exigentes, en especial por la gente que no tiene conocimientos previos. Que las evaluaciones sean de dificultad similar a lo visto en clases, y que estas clases permitan a todos aprender correctamente.”

Estilo sección 5

Nivel de exigencia: **Alto.**

Estilo sección 5

Nivel de exigencia: **Alto**.

... pero no los dejaré solos:

Estilo sección 5

Nivel de exigencia: **Alto**.

... pero no los dejaré solos:

- Ir de lo simple a lo complejo.

Estilo sección 5

Nivel de exigencia: **Alto**.

... pero no los dejaré solos:

- Ir de lo simple a lo complejo.
- Muchos ejemplos (de buena dificultad).
- Ejercicios propuestos.

Estilo sección 5

Nivel de exigencia: **Alto**.

... pero no los dejaré solos:

- Ir de lo simple a lo complejo.
- Muchos ejemplos (de buena dificultad).
- Ejercicios propuestos.
- Responder dudas.
- Conversaciones sinceras.

Estilo sección 5

OJO: Esta sección es para quienes estén dispuestos a trabajar desde la clase 1.

Estilo sección 5

OJO: Esta sección es para quienes estén dispuestos a trabajar desde la clase 1.

“Ya tome el curso el semestre pasado y siento que fue netamente mi culpa el no pasarlo, me costo tomarle el ritmo y no aproveche bien todas las ayudas que tu como profesor ofrecías.”

Algoritmo

Definición

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

Ejemplo

Problema:

Resolver el laberinto.

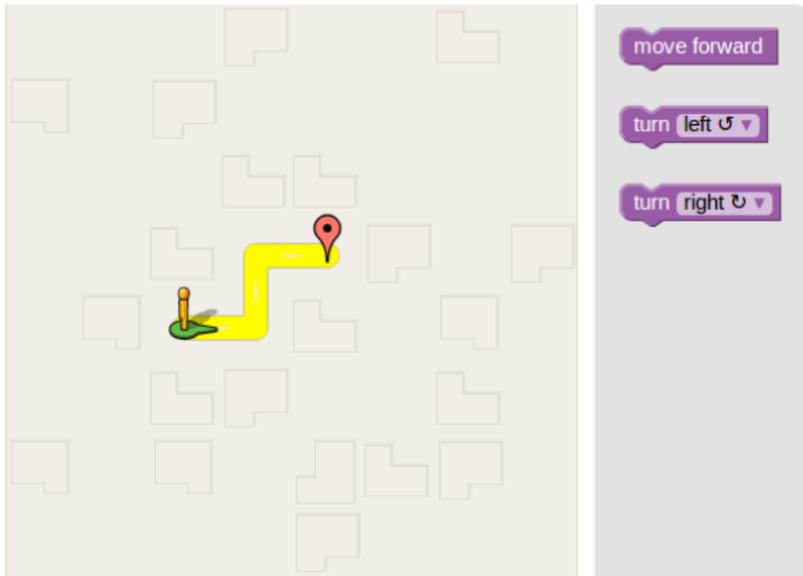
Herramientas:

Avanzar 1 paso

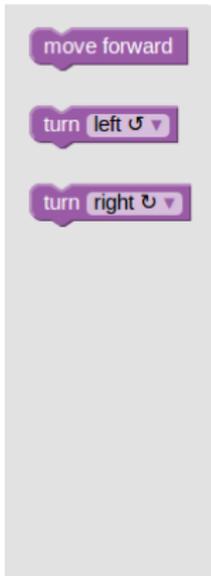
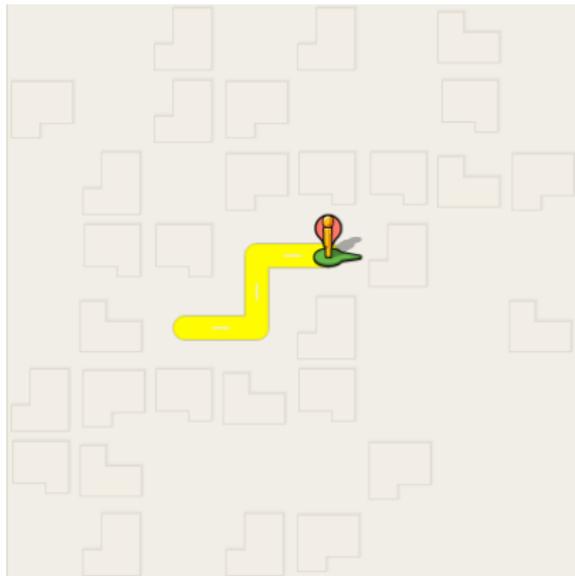
Girar izquierda

Girar derecha

Ejemplo



Ejemplo



Ejemplo

The image shows a programming environment with a maze on the left and a script on the right. The maze is a grid of light gray blocks with a green robot at the start and a red pin at the end. A yellow path with a dashed line leads from the robot to the pin. A speech bubble at the bottom left says "You have 2 blocks left." The script on the right consists of four blocks: "move forward", "turn left", "turn right", and a "repeat until" loop containing a "do" block.

move forward

turn left

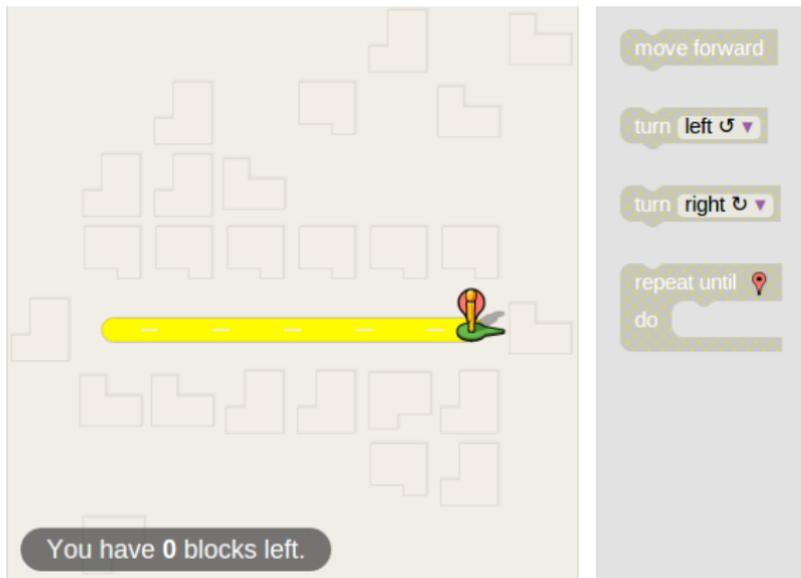
turn right

repeat until

do

You have 2 blocks left.

Ejemplo



The image shows a screenshot of a block-based programming environment. On the left, a maze is displayed with a green robot character at the bottom center. A yellow path with dashed lines leads from the robot to the left. A dark grey box at the bottom left of the maze area contains the text "You have 0 blocks left." On the right, a code editor shows a sequence of blocks: "move forward", "turn left", "turn right", and a "repeat until" loop containing a "do" block.

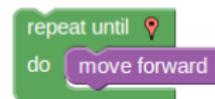
move forward

turn left

turn right

repeat until

do



A close-up of a "repeat until" loop block. The block is green and contains a "do" block with a purple "move forward" block inside it. A red location pin icon is visible on the "repeat until" block.

repeat until

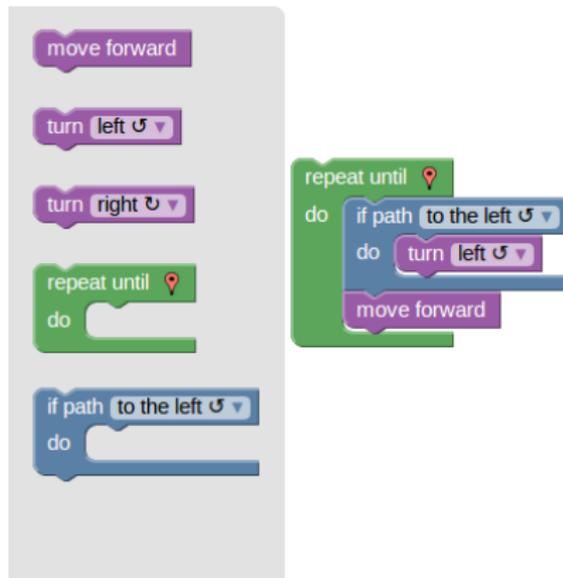
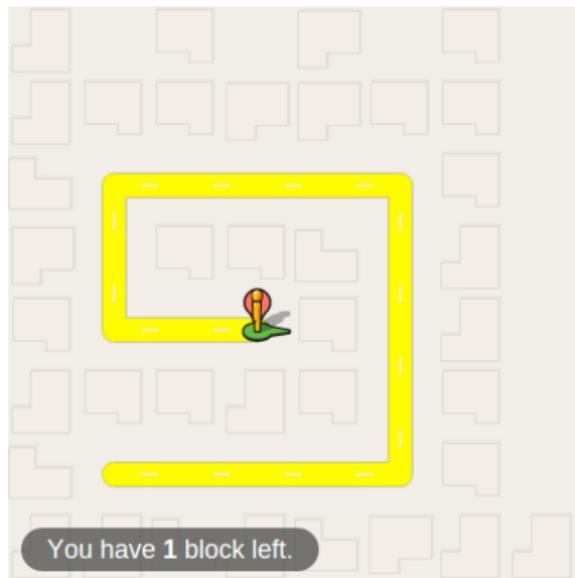
do

move forward

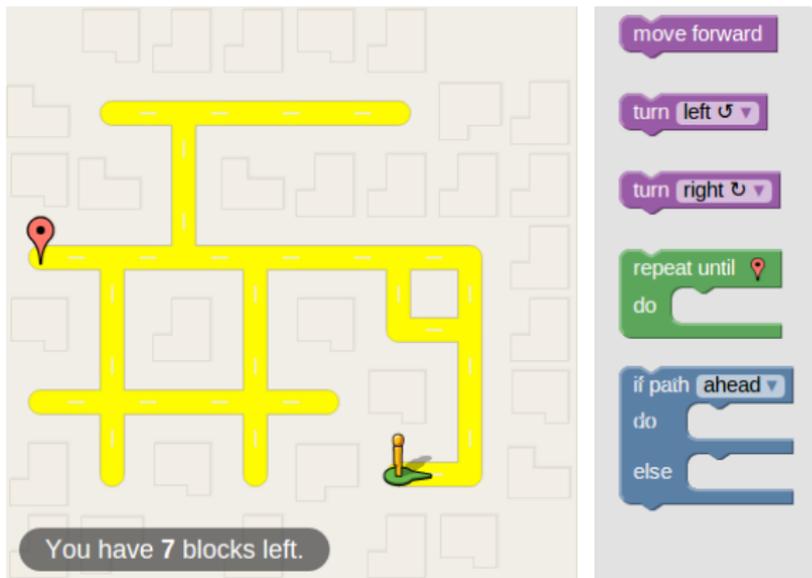
Ejemplo



Ejemplo



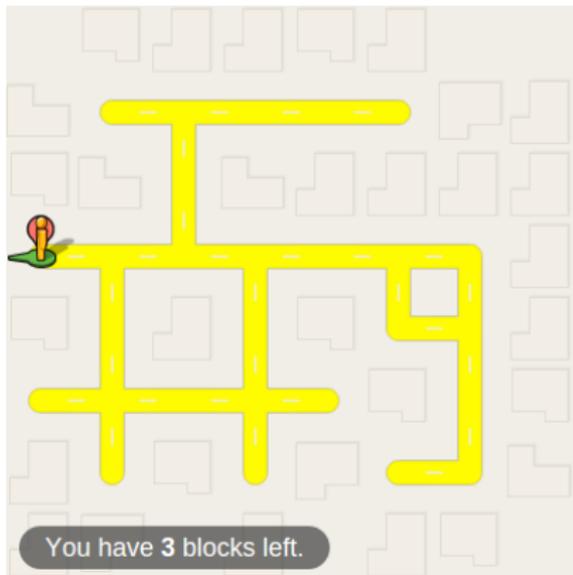
Ejemplo



The image shows a maze-solving example. On the left, a maze is displayed with a yellow path starting from a red location pin on the left and ending at a green location pin on the right. A grey speech bubble at the bottom left of the maze says "You have 7 blocks left." On the right, a code block contains the following instructions:

```
move forward
turn left ↶
turn right ↷
repeat until [red pin]
do
if path ahead
do
else
```

Ejemplo



Tarea!

Ingrese a Blockly pinchando **aquí**.

Resuelva los juegos en orden:

- 1 Puzzle.
- 2 Maze.
- 3 Bird.
- 4 Turtle.
- 5 Movie.
- 6 Pond.
- 7 JS Pond.