



# **Ciencias de la Computación**

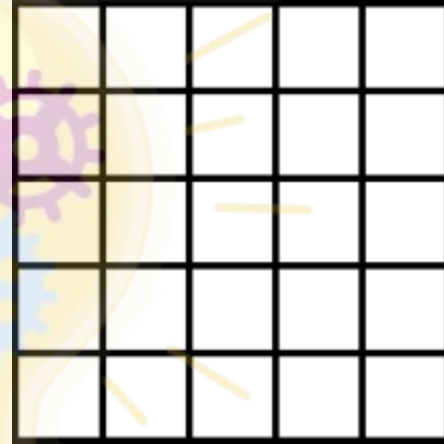
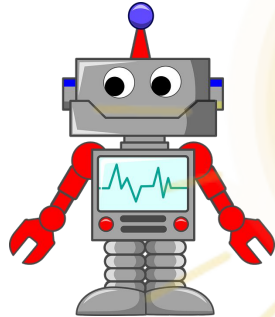
## **Introducción a la Programación**

### Dibujando en la Cuadrícula



**Departamento de Computación**  
**Facultad de Ciencias Exactas Físico-Químicas y Naturales**  
**Universidad Nacional de Río Cuarto**

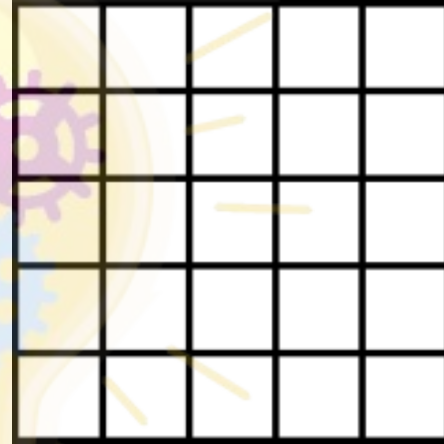
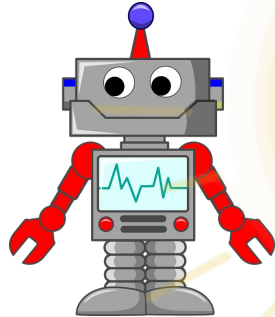
## Actividad: Dibujando en una cuadrícula



Debemos darle órdenes al Robot para que dibuje un cuadrado negro en la cuadrícula

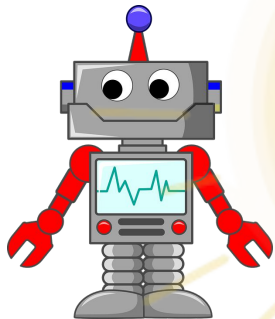
**Ejercicio 1: Escribir individualmente en papel las órdenes**

## Actividad: Dibujando en una cuadrícula



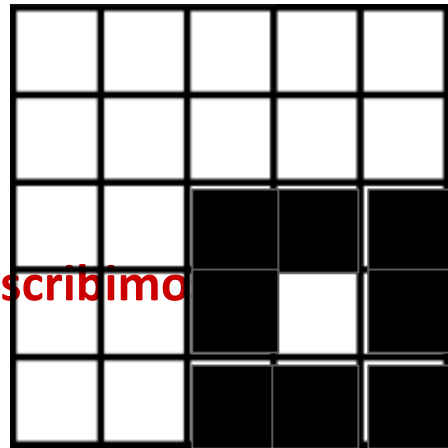
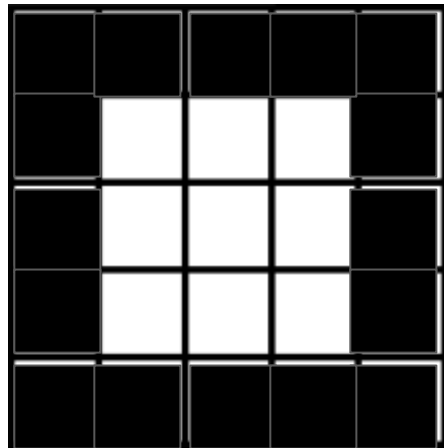
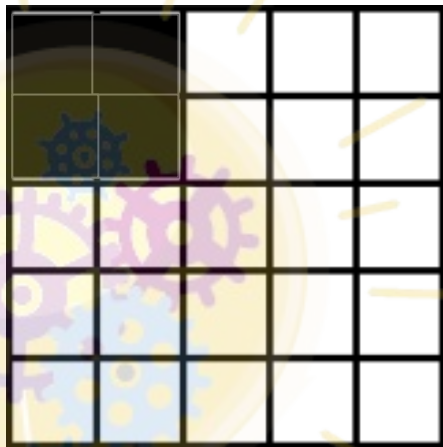
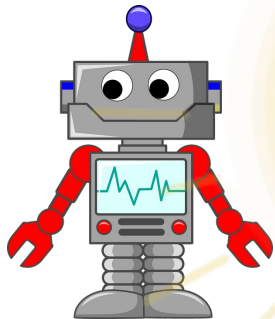
**Veamos que hace el Robot con las órdenes que escribimos...**

## Actividad: Dibujando en una cuadrícula



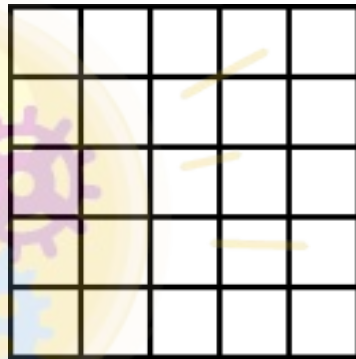
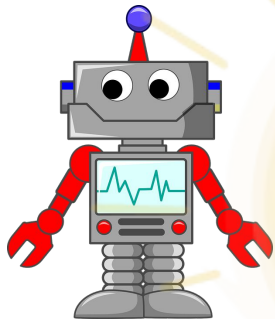
**Veamos que hace el Robot con las órdenes que escribimos...**

## Actividad: Dibujando en una cuadrícula



Veamos que hace el Robot con las órdenes que escribimos

## Actividad: Dibujando en una cuadrícula

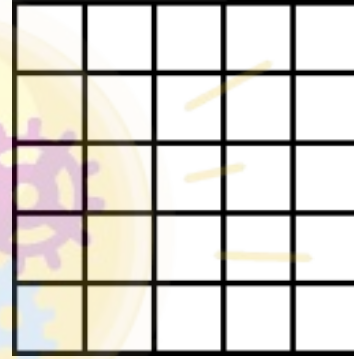
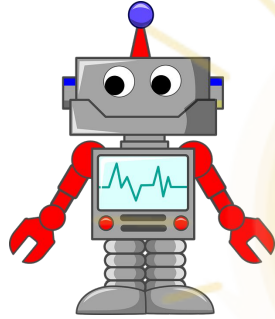


Si el enunciado se reformulara de la siguiente manera:

Debemos darle las órdenes al Robot para que pinte de negro el borde del cuadrado

Qué órdenes le daría al Robot? son las mismas que pensó inicialmente

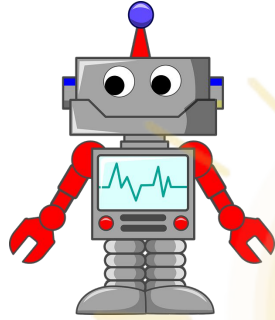
## Actividad: Dibujando en una cuadrícula



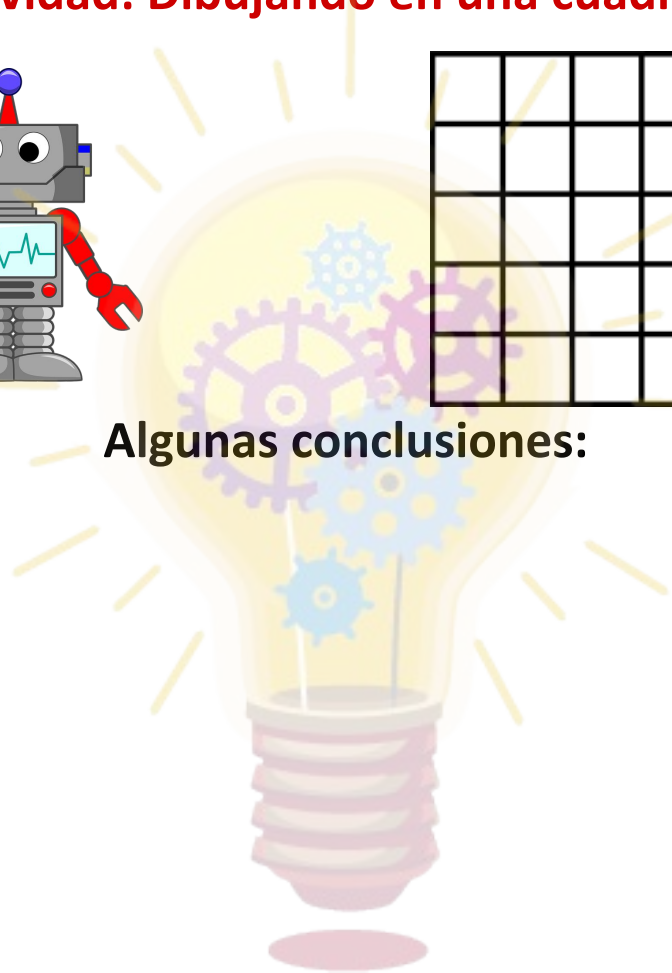
Reflexionemos:

- ¿Un robot realmente puede “entender” lo que la gente dice?
- ¿El Robot puede entender todas las órdenes que le damos?
- ¿Cómo las entiende? ¿Qué pasa si la orden es ambigua?
- ¿Hay una sola secuencia de órdenes posible?
- ...

## Actividad: Dibujando en una cuadrícula

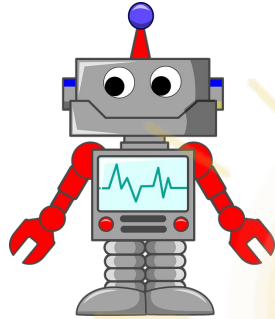


Algunas conclusiones:





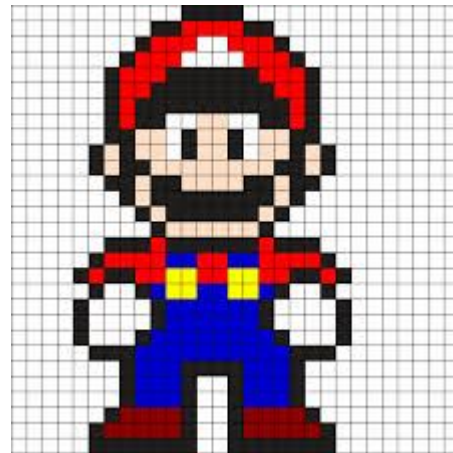
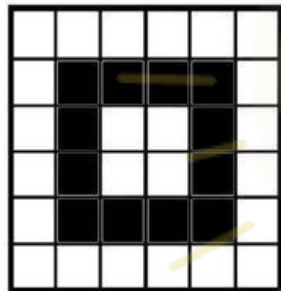
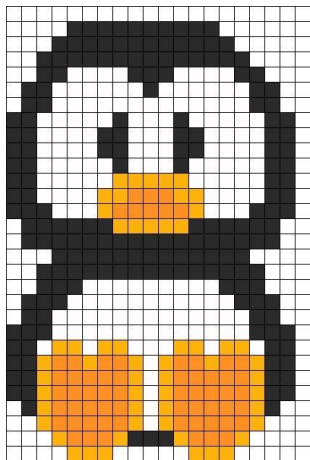
## Actividad: Dibujando en una cuadrícula



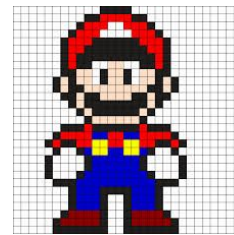
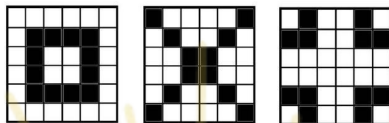
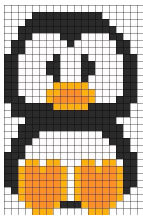
### Algunas conclusiones:

- El Robot ejecuta exactamente lo que dicen las instrucciones. No tiene en cuenta, por ejemplo, el tono de voz, señas, sentido común, ...
- Surge la necesidad de definir/conocer el conjunto de instrucciones que el Robot “entiende” y cómo las “entiende”
- Los robots operan con “instrucciones”, porque fueron programados para poder hacerlas
- Para poder llevar a cabo una tarea, un robot necesita tener una lista de instrucciones que puedan ejecutar

## Actividad: Dibujando en una cuadrícula



Daremos instrucciones para colorear cuadrados  
en una hoja cuadriculada, con el fin de reproducir una  
imagen en la hoja



## Actividad: Dibujando en una cuadrícula

¿Cómo dibujamos en una cuadrícula?



Mover un cuadrado adelante



Mover un cuadrado atrás



Mover un cuadrado arriba



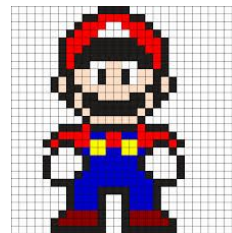
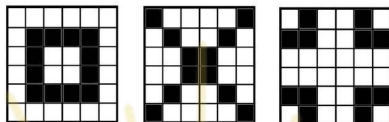
Mover un cuadrado abajo



Cambiar al siguiente color



Pintar cuadrado con el color seleccionado



## Actividad: Dibujando en una cuadrícula

### ¿Cómo dibujamos en una cuadrícula?

→ — Mover un cuadrado adelante

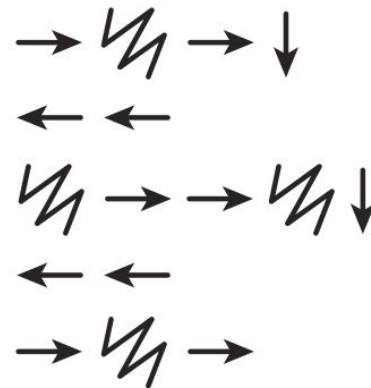
← — Mover un cuadrado atrás

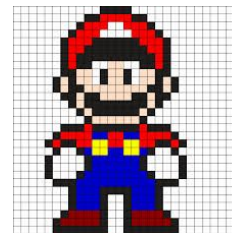
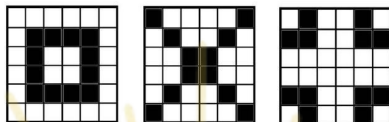
↑ — Mover un cuadrado arriba

↓ — Mover un cuadrado abajo

↻ — Cambiar al siguiente color

⚡ — Pintar cuadrado con el color seleccionado

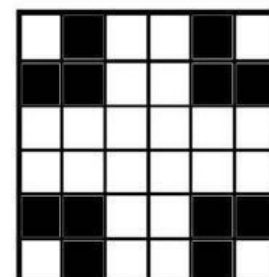


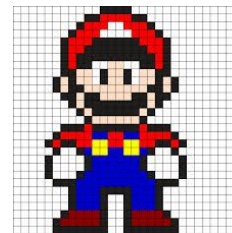
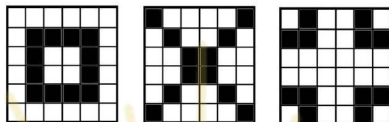


## Actividad: Dibujando en una cuadrícula

Selecciona una imagen y da la secuencia de instrucciones para dibujarla.

- — Mover un cuadrado adelante
- ← — Mover un cuadrado atrás
- ↑ — Mover un cuadrado arriba
- ↓ — Mover un cuadrado abajo
- ↻ — Cambiar al siguiente color
- ⚡ — Pintar cuadrado con el color seleccionado

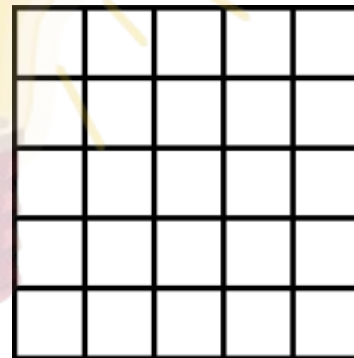




## Actividad: Dibujando en una cuadrícula

Dada la secuencia de instrucciones dibuja la imagen

↓ → → → ↘ → → → ↘ → ↓ ↓ ↘ ← ← ← ← ← ↘  
→ ↓ ↘ → → → ↘ ← ↓ ↘ ← ↘ →

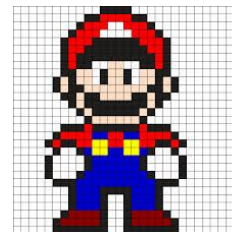
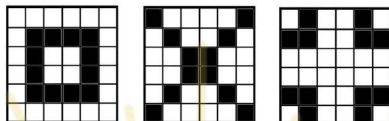


## Actividad: ¿como hacer un sandwich?

<https://www.youtube.com/watch?v=ajkgIMnByFM>



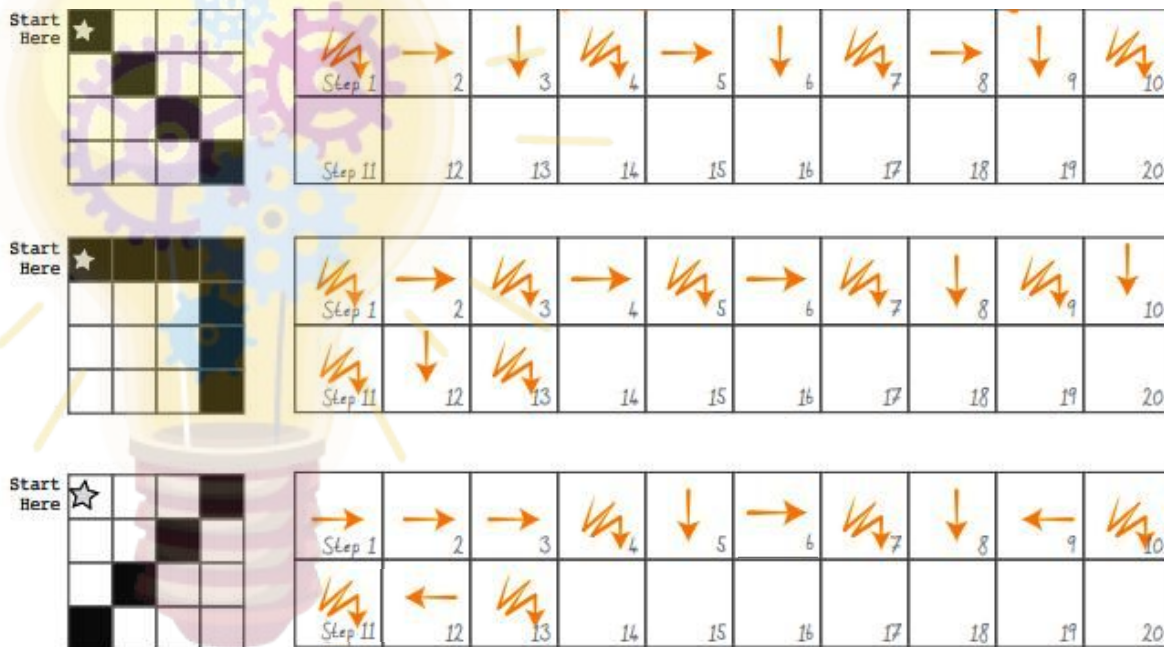




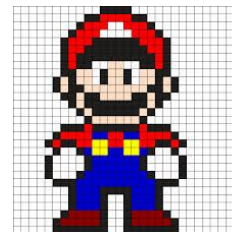
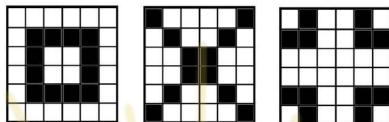
## Actividad: Dibujando en una cuadrícula

¿Son correctas las  
secuencias de  
instrucciones?

En caso de que alguna sea  
incorrecta,  
modifique la secuencia de  
instrucciones para  
que dibuje lo esperado







## Actividad: Dibujando en una cuadrícula

¿Son correctas las secuencias de instrucciones?

En caso de que alguna sea incorrecta, modifique la secuencia de instrucciones para que dibuje lo esperado



## **Reflexionamos...**

**¿Qué errores tuvieron a la hora de encontrar las soluciones?**

**¿Se repitieron esos errores?**

**¿Cómo detectaron esos errores? ¿Cómo encontraron en qué lugar se produce el error?**

**¿Consideran que utilizó algún método sistemático para corroborar que su solución era correcta?**

## Prueba y Depuración

**Prueba (validación) de las soluciones:** revisión de nuestras soluciones para corroborar que efectivamente realizan lo esperado.  
Responde a la pregunta ¿es una solución al problema?

**Depuración (debugging) de las soluciones:** proceso de identificar y corregir errores.

Son actividades muy importantes cuando programamos.

## Algoritmos

**Algoritmo:** secuencia de instrucciones que realizadas en orden conducen a obtener la solución de un problema.

**Programa:** secuencia de instrucciones, escritas para realizar una tarea específica en una computadora.

**Lenguaje de Programación:** lenguaje formal que especifica una serie de instrucciones para que una computadora las ejecute

Los lenguajes de programación pueden usarse para crear programas que pongan en práctica algoritmos y se ejecuten en una computadora

## Algoritmos

*En las Ciencias de la Computación, y en la programación, los algoritmos son más importantes que los lenguajes de programación o las computadoras.*

*Un lenguaje de programación es sólo un medio para expresar un algoritmo y una computadora es sólo un procesador para ejecutarlo*

# Programación y Ciencias de la Computación

Encarar un problema a resolver puede ser aterrador, pero con estas simples herramientas, nada es imposible:

**Paso 1) Descomponer** — partir un gran problema en algo mucho más simple. Muchas veces, los grandes problemas consisten de muchos pequeños problemas, todos juntos en el problema mayor.

**Paso 2) Patrones** — A veces, cuando un problema tiene muchas pequeñas partes, notarás que esas partes tienen algo en común. Si no lo tienen, entonces tal vez se parecen en mayor o menor medida a algo que ya fue resuelto con anterioridad. Si evidencias estos patrones, se vuelve más simple entender las distintas piezas que forman el problema.

**Paso 3) Abstracción** — La abstracción permite reducir los detalles para centrarse en la información relevante para resolver un problema.

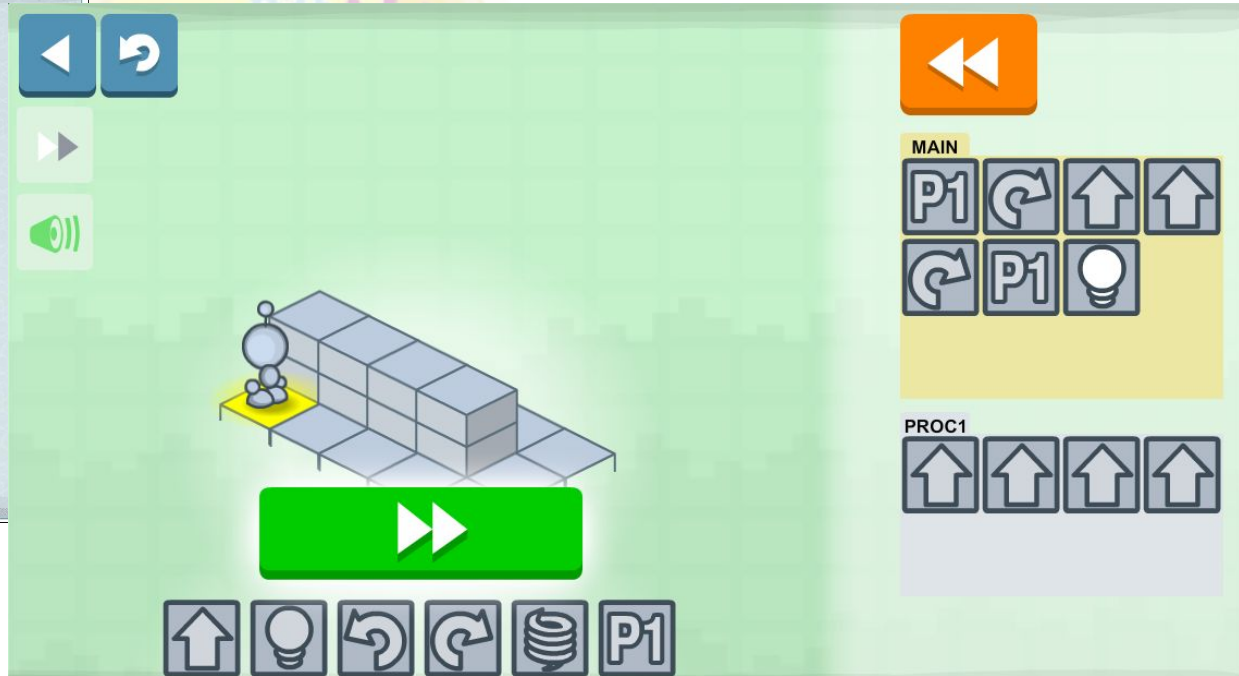
**Paso 4) Algoritmos** — Cuando una solución está completa, puedes realizar una descripción que permita procesarla paso a paso.

Y si queremos un programa que implemente nuestra solución

**Paso 5) Implementar** — traducir el algoritmo a un lenguaje de programación.

## Actividad: LightBot

Ingresa a <https://www.minijuegos.com/juego/light-bot>  
y realiza las actividades







### ¿Qué estudia esta disciplina?

- Análisis y desarrollo de *algoritmos*
- Representación de problemas e información (*estructuras de datos*)
- Estudio y desarrollo de *modelos de computación*, es decir de la *maquinaria* necesaria para computar
- Áreas prácticas: Gráficos, animación, simulación, redes y telecomunicaciones, aprendizaje automático (inteligencia artificial), ingeniería de software, ...

### Problema fundamental

*¿Qué tipos de problemas se pueden resolver automáticamente (por una máquina)?*





## Programas y Algoritmos

### ¿Qué es un algoritmo?

Una *secuencia de instrucciones bien definidas* que *resuelve un problema concreto* luego de una *secuencia de pasos finita (termina)*.

### ¿Cómo se describe un programa?

Por medio de un *lenguaje* que permite *describir* las instrucciones en cada paso

- **Lenguaje natural:** Problemas de *falta de precisión* y posibles *ambigüedades*
- **Código de máquina:** Representación de las instrucciones y datos directamente en la forma que *entiende la máquina*
- **Lenguaje de programación:** Provee una *máquina abstracta* (de más alto nivel), haciendo los *programas* más fáciles de escribir y entender



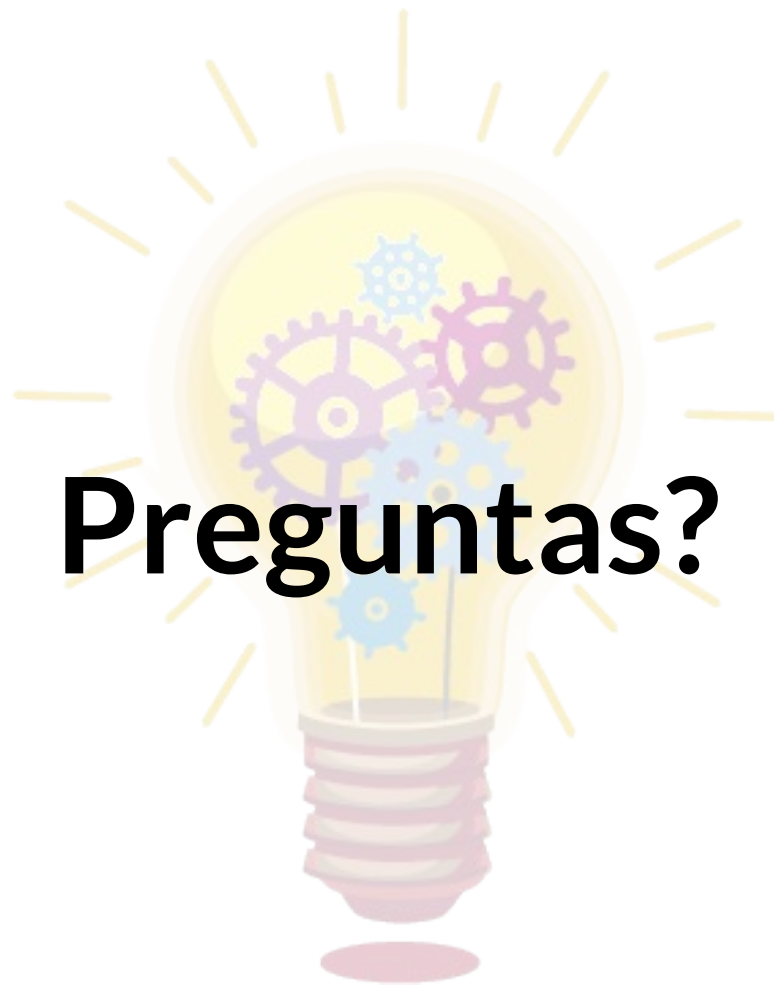
## Áreas de estudio

- Programación
  - Estructuras de datos y algoritmos
  - Lenguajes y *paradigmas* de programación
- Arquitecturas de computadoras (hardware)
- Bases de datos
- Ingeniería de software (gestión de proyectos, diseño, ...)
- Redes, telecomunicaciones y aplicaciones en red
- Software de base: Compiladores, sistemas operativos, ...
- Teoría: Autómatas y lenguajes, modelos de computación, computabilidad, complejidad computacional, ...
- Áreas básicas: Análisis matemático (cálculo), álgebra, geometría, lógica, estadística, ...
- Otras: Inteligencia artificial, simulación, gráficos, ...



## Para programar, ¿Sólo necesito aprender un lenguaje?

- ❶ **NO:** *Programar es resolver problemas mediante algoritmos*
- ❷ Un *lenguaje de programación* sólo permite describir programas
- ❸ Para *diseñar y escribir* una solución (programa) se requiere:
  - ❶ Definir cómo *representar* el problema.  
Uso de *estructuras de datos*:
  - ❷ Definir operaciones y manipulaciones (algoritmos) sobre esas estructuras de datos
- ❹ Otros problemas:
  - Almacenamiento y procesamiento de grandes volúmenes de datos (*bases de datos*)
  - Lidar con la *complejidad* del problema para lograr programas *eficientes y usables*



# Preguntas?