

# Programación

Clase 1

¿Qué es la programación?  
Primeros autómatas

Universidad Nacional de Quilmes

# Comenzando el día

¿Qué cosas hicieron a la mañana/tarde/noche para conectarse a la reunión o clase virtual?

¿Importa el orden en el cual realizamos las actividades?

Por ejemplo, una persona podría bañarse primero, luego desayunar, y conectarse a la clase, y otra podría hacerlo al revés, primero desayunar, luego bañarse, y por último conectarse, lo que no implicaría un problema de orden.

Pero, ¿qué pasa si antes de conectarse, no prende la computadora o el celular?

Veamos otro ejemplo más completo.



**Cebar mate:** ¿en qué orden van las acciones?



## Cebar mate: listamos las acciones

Para lograr nuestro propósito debemos seguir una serie de pasos ordenados:

**Paso 1:** Encender la hornalla

**Paso 2:** Cargar la pava con agua

**Paso 3:** Poner la pava en el fuego

**Paso 4:** Colocar un poco de jugo a la pava

**Paso 5:** Verificar temperatura del agua. Si la temperatura está entre 70 y 80 grados, retirar la pava del fuego

**Paso 6:** Verter la yerba del mate hasta unas  $\frac{3}{4}$  partes del mismo

**Paso 7:** Tapar con una mano la boca del mate, invertir y agitar unos instantes

**Paso 8:** En el mate, formar un pequeño hueco a un costado dejando la yerba recostada sobre un lado del mate

**Paso 9:** Colocar agua tibia sobre el hueco

**Paso 10:** Usar lo que resta de agua de la pava y colocarla en una taza para café

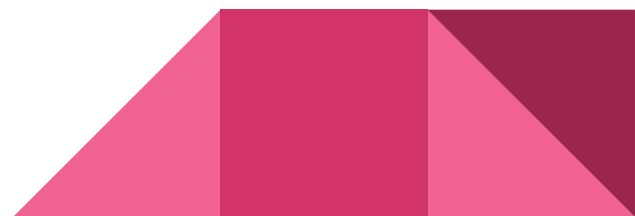
**Paso 11:** Colocar café en la taza

**Paso 12:** Ubicar la bombilla sobre el sector de la yerba húmeda

**Paso 13:** Esperar un minuto

**Paso 14:** Volcar agua en el mate hasta el borde

**¿ Notamos algo particular?**



## Cebiar mate: encontramos errores

**Paso 1:** Encender la hornalla

**Paso 2:** Cargar la pava con agua

**Paso 3:** Poner la pava en el fuego

**Paso 4:** Colocar un poco de jugo a la pava

**Paso 5:** Verificar temperatura del agua. Si la temperatura está entre 70 y 80 grados, retirar la pava del fuego

**Paso 6:** Verter la yerba del mate hasta unas  $\frac{3}{4}$  partes del mismo

**Paso 7:** Tapar con una mano la boca del mate, invertir y agitar unos instantes

**Paso 8:** En el mate, formar un pequeño hueco a un costado dejando la yerba recostada sobre un lado del mate

**Paso 9:** Colocar agua tibia sobre el hueco

**Paso 10:** Usar lo que resta de agua de la pava y colocarla en una taza para café

**Paso 11:** Colocar café en la taza

**Paso 12:** Ubicar la bombilla sobre el sector de la yerba húmeda

**Paso 13:** Esperar un minuto

**Paso 14:** Volcar agua en el mate hasta el borde

En **rojo** marcamos los **pasos erróneos**, y en **azul** los pasos **innecesarios** para cumplir con el propósito



# Algoritmo

Algoritmo es una secuencia ordenada de instrucciones para alcanzar un objetivo.

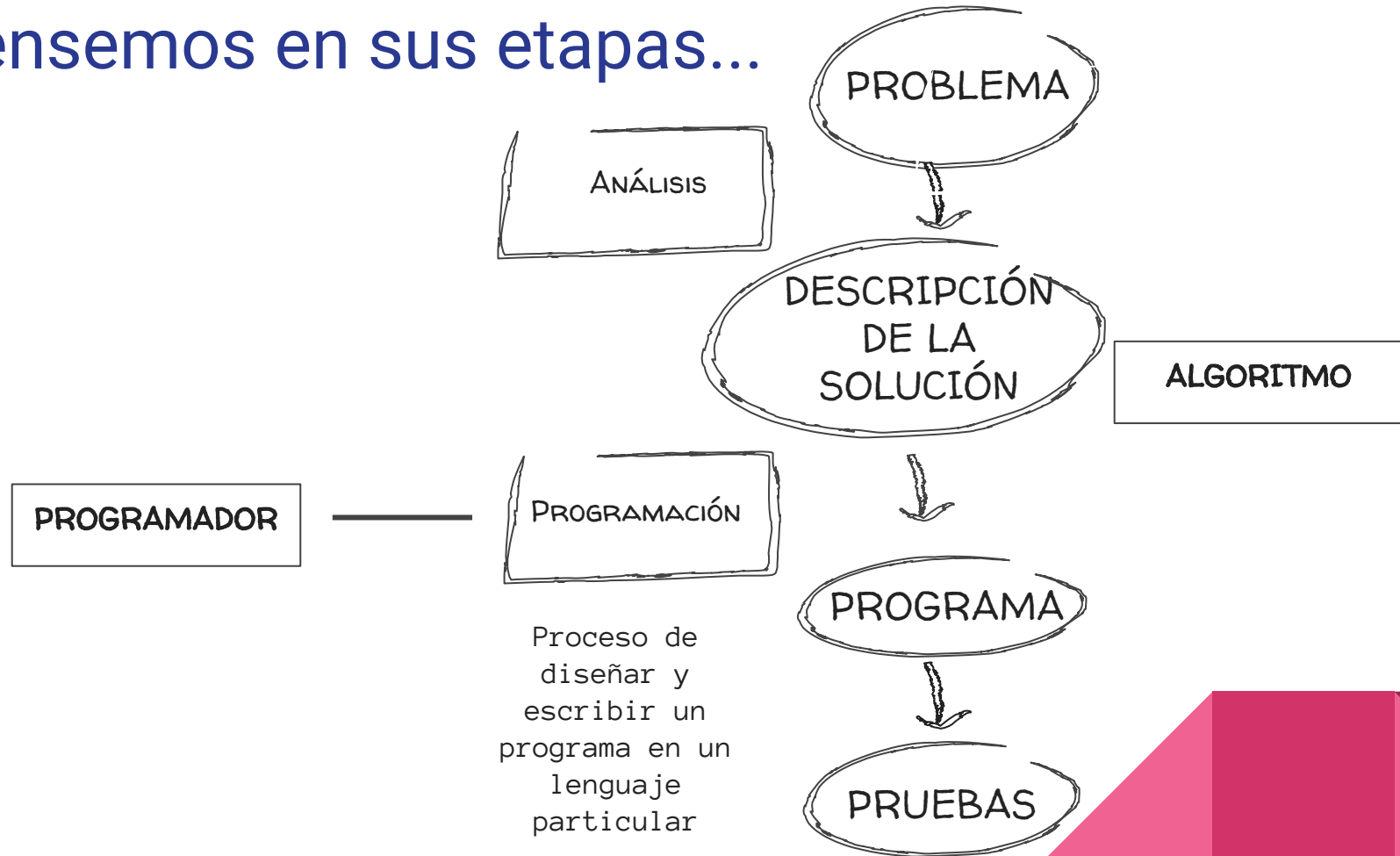
Las secuencias tienen un orden que puede cambiar sin que esto afecte el resultado, mientras que en otros casos esto no es posible.



# ¿Qué es la programación?




# Pensemos en sus etapas...





# Etapas de la programación

La tarea de programar tiene etapas:

- Analizar el propósito/problema: entender bien el propósito, buscar la posible solución mediante un algoritmo
  - Escribir el código del algoritmo (código fuente): especificarle a la computadora las instrucciones que debe realizar para resolver el propósito/problema
  - Probar que el código escrito en el programa haga lo requerido en el propósito (resolver el problema)
- 

# Programar

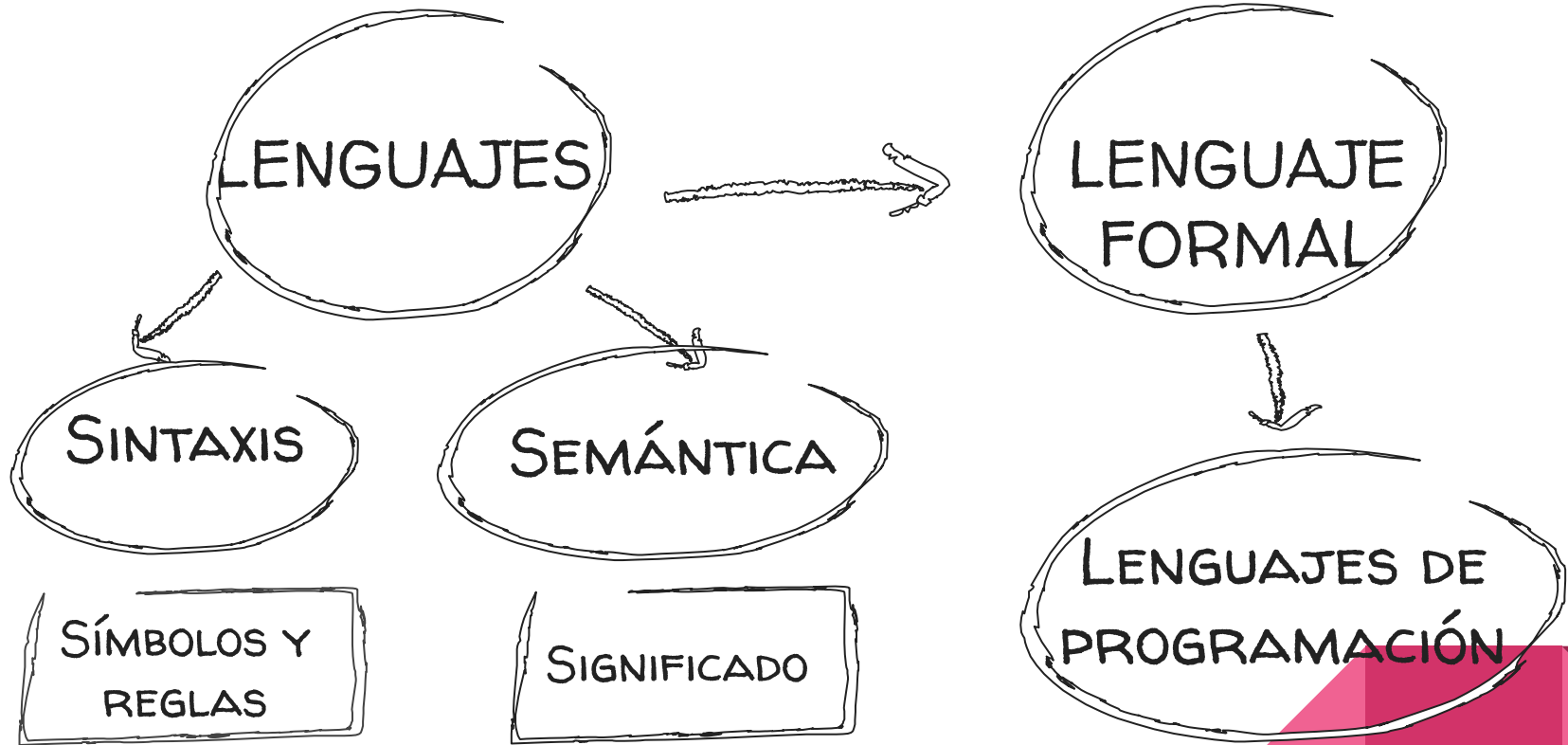
Programar es **comunicar** descripciones de soluciones a problemas en un **lenguaje** particular.

Un programa se puede ver como la **descripción de la solución a un problema.**

Un programa debe estar redactado en algún **lenguaje** (por ejemplo, en español).



# Repasemos entonces que es un lenguaje...



# Lenguaje de programación

Piensen que la solución que ustedes están escribiendo en su programa justamente está escrito en un lenguaje que la máquina/autómata puede interpretar.

Acá es importante ver la diferencia entre código fuente y código objeto

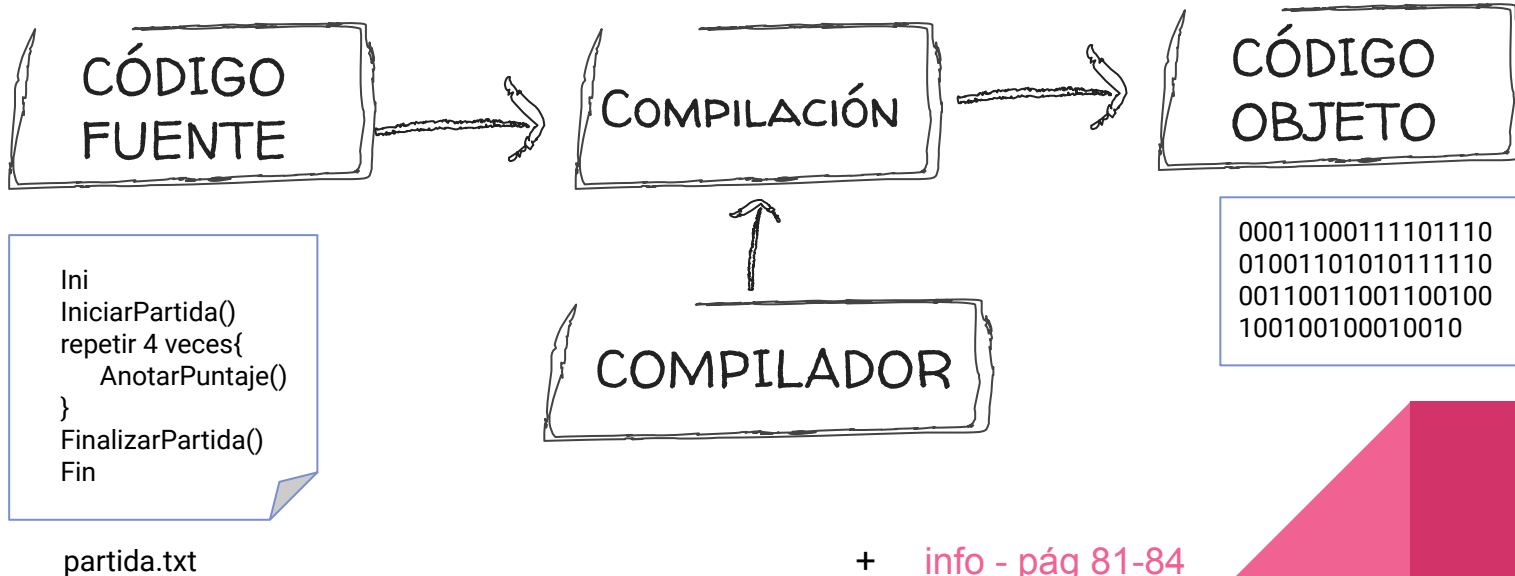
# Lenguaje de programación - diferencia códigos

L@s programadorxs utilizamos lenguajes de alto nivel para escribir la solución a un problema, es decir el **código fuente**, que se acercan más a nuestro lenguaje natural. ¡Por suerte!

Ya que los compiladores son los que se encargan de pasar ese código fuente a **código objeto**, capaz de ser ejecutado en una computadora.


# Código fuente - objeto (continuación)

En resumen y de forma muy acotada, podemos verlo de la siguiente manera:



# ¿Qué hace un programador/a?

Escribe, depura y mantiene el código fuente de **programas** informáticos, es decir, el conjunto de instrucciones que ejecuta el hardware de una computadora, para realizar una tarea determinada, y así solucionar un problema dado.




# ¿Nos interesan todos los problemas?

**NO.** Sólo aquellos que tienen una solución que puede ser calculada por una computadora (**Autómata**).

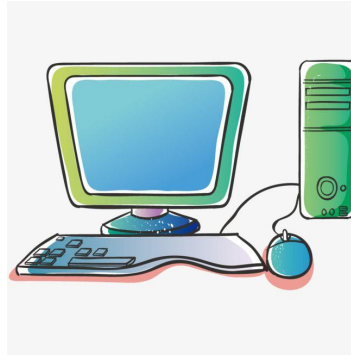
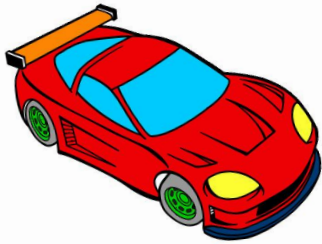
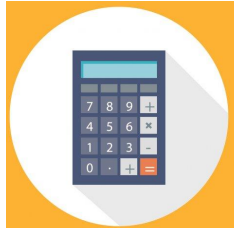
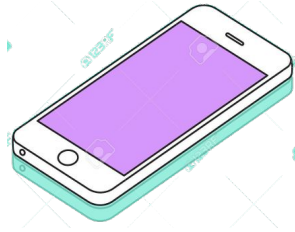




# Autómatas

- Entienden un lenguaje determinado, acotado de instrucciones.
  - Siguen al pie de la letra las instrucciones que se les dá.
  - La máquina no decide qué hacer, ejecuta lo que le pedimos.
  - Las instrucciones no pueden ser ambiguas.
- 

# Ejemplos de autómatas



# Entonces...

Programar es **comunicar** descripciones **ejecutables** de soluciones a problemas **computacionales** en un **lenguaje capaz de ser entendido por un autómata.**



# Programar es Comunicar

- Comunicamos al autómeta
- Comunicamos a otros/as programadores/as
- Comunicamos a nosotrxs mismxs (que luego nos olvidamos qué hicimos y por qué)



=




# Habilidades para la Programación

La programación es una **disciplina** que requiere simultáneamente del uso de cierto grado de **creatividad**, un conjunto de **conocimientos técnicos** asociados y la capacidad de operar constantemente con **abstracciones**.



# Veamos...

- **Creatividad:** se puede crear algo de la nada. Los problemas a veces requieren pensar “fuera de la caja”
  - **Conocimientos técnicos:** hay que saber como darle las instrucciones de forma precisa y clara a la máquina. Muchos problemas requieren conocimientos lógicos y matemáticos complejos para poder solucionarlos.
  - **Abstracción:** Ver estructuras y elementos en la computadora es abstraer la realidad a modelos matemáticos o mentales para poder operarlos.
- 

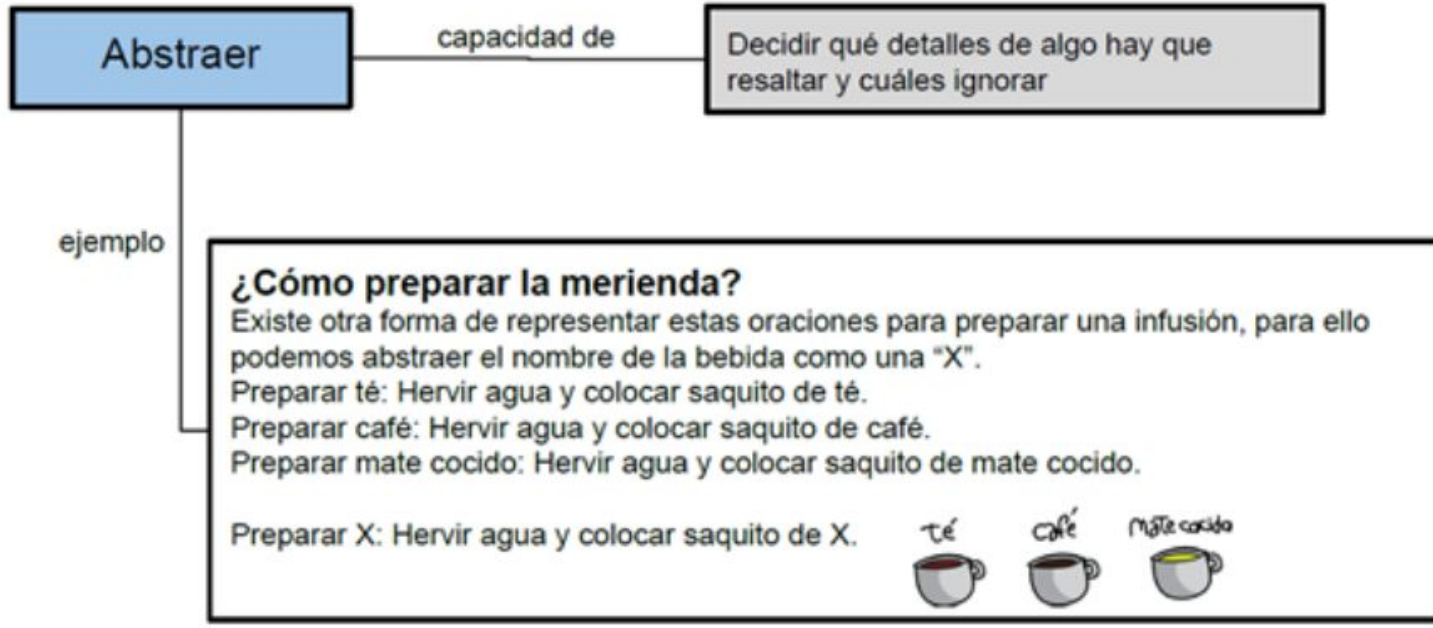
# Conceptos esenciales de programación

Conceptos claves que constituyen el Pensamiento computacional:

- Capacidad de pensar en términos **abstractos**, seleccionando buenas representaciones
- Capacidad de pensar en **generalizaciones**, identificando y utilizando patrones
- Capacidad de pensar en términos de **descomposición**
- Capacidad de pensar de forma **algorítmica**
- Capacidad de pensar en términos de **evaluación** (las pruebas)

# Abstracción

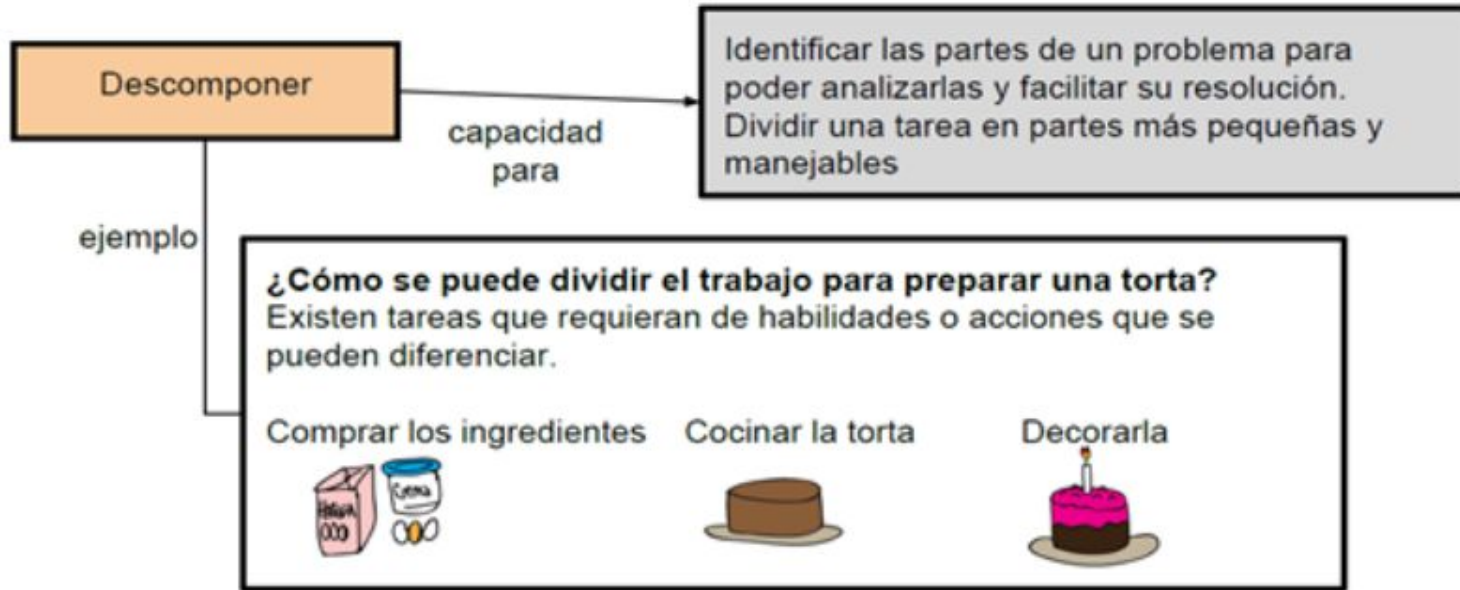
Se refiere a la capacidad de elegir **representaciones** que permitan simplificar un problema y así poder realizar conceptualizaciones sobre los mismos.





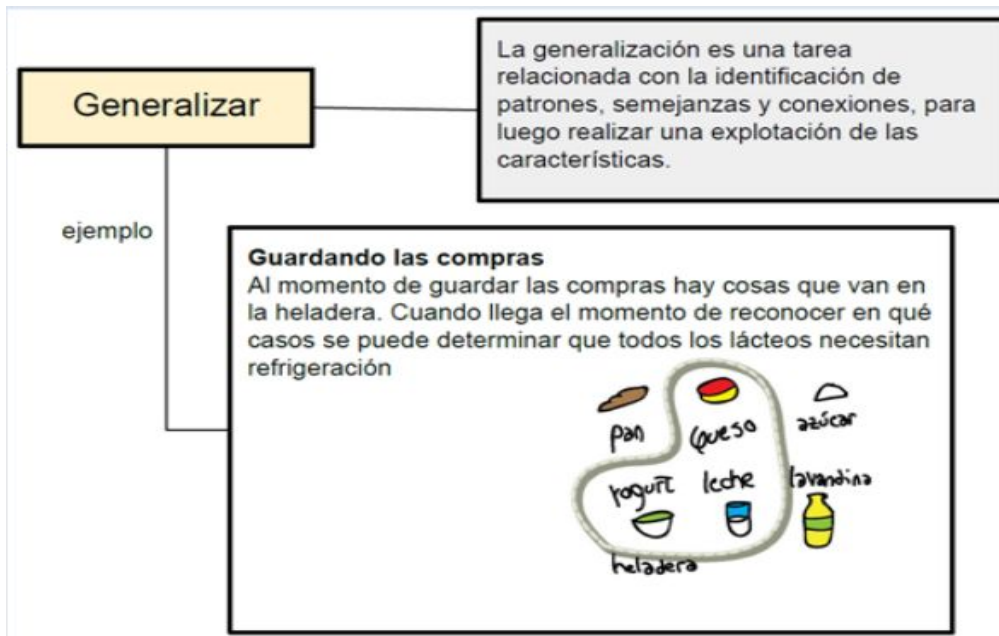
# Descomposición

Se refiere a la capacidad de poder **dividir** e identificar las partes más pequeñas que componen a un problema, haciéndolo más fácil de analizar y solucionar.



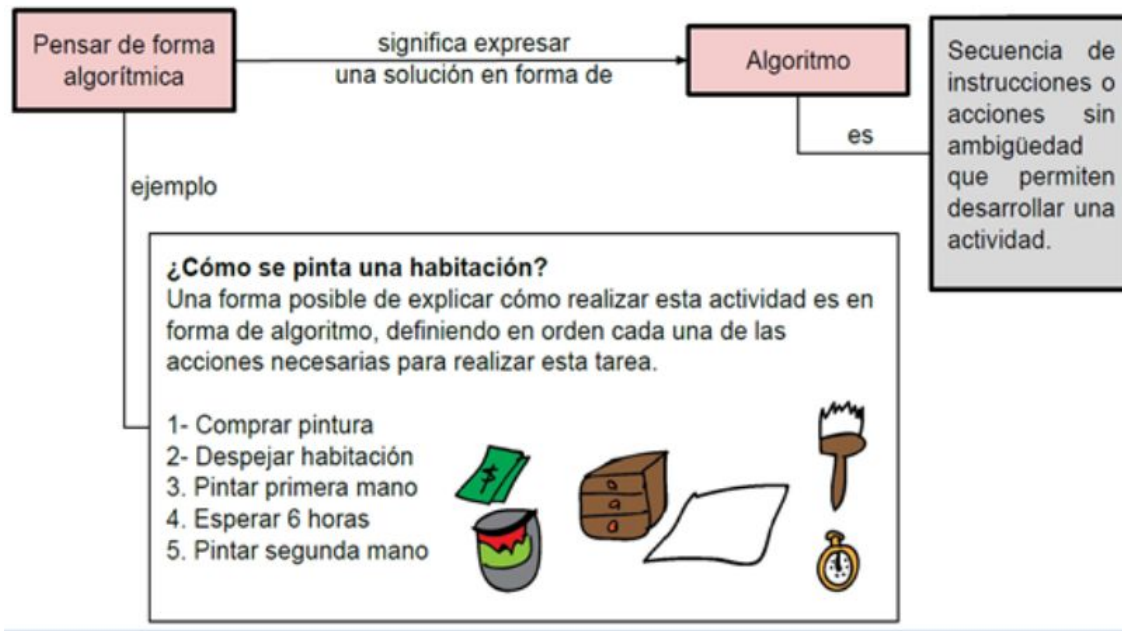
# Generalización

Se refiere a la capacidad de **descubrir patrones** en los elementos que componen un problema o en las soluciones que son aplicables a ellas. Esta capacidad guarda una estrecha relación con la abstracción.



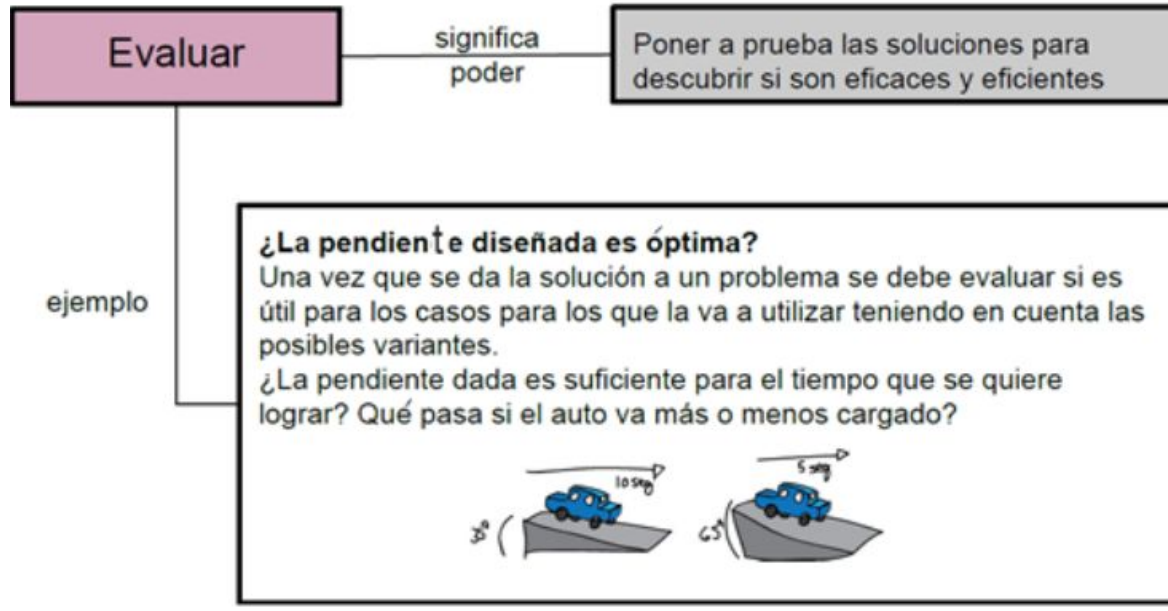
# Algoritmia

Es la capacidad para expresar soluciones de forma tal que se componga de una serie de **pasos finitos, no ambiguos y ordenados** que permitan que una persona o un autómatas pueda llevarlo adelante.

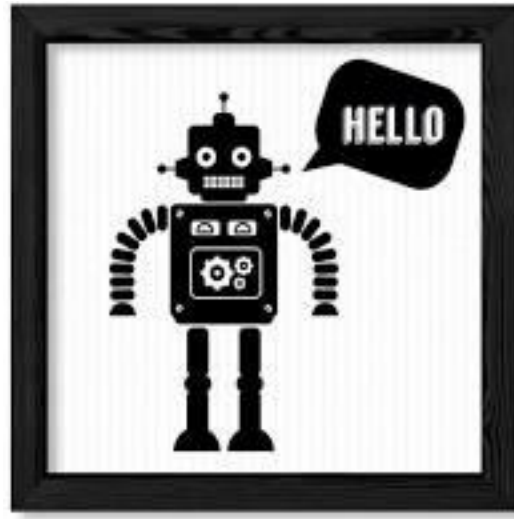


# Evaluación (Probar)

Se refiere a la capacidad de poder analizar críticamente las soluciones creadas para detectar y corregir errores, como así también verificar si una solución es eficiente en término de uso de recursos.



# Robot Humano



# Instrucciones

Supongamos que el robot contiene este set limitado de instrucciones:

- Avanzar un paso
- Girar a la derecha
- Girar a la izquierda
- Abrir
- Sentarse



# Ejemplo

Por ejemplo, le pedimos al robot que realice las siguientes tareas:

- Abrir la ventana
- Sentarse en una silla
- Cruzar la calle



# Robot = Autómata

Ahora ¿Qué ocurriría en los siguientes casos?

- Se le ordena al autómata que se siente en una silla, pero no hay ninguna silla.
- Se le dice al autómata que levante el brazo derecho y ya lo tiene levantado.
- Se le pide al autómata que levante el brazo derecho mientras tiene levantado el brazo izquierdo, y no se especifica si antes debe bajar este brazo o no.



# Volviendo al ejemplo

Reformulamos el pedido de la primera tarea, de la siguiente manera:

1. Caminar hasta la ventana
2. Detenerse
3. Abrir ventana

**Precondición:** Debe existir una ventana en dirección al robot!

**Efecto:** La ventana de la habitación donde se encuentra el robot está abierta

# Lightbot



# Lightbot

Lightbot es un videojuego que permite aprender algunas de las bases de la programación jugando con un autómata.

Sitio web oficial: <https://lightbot.com>

Disponible para

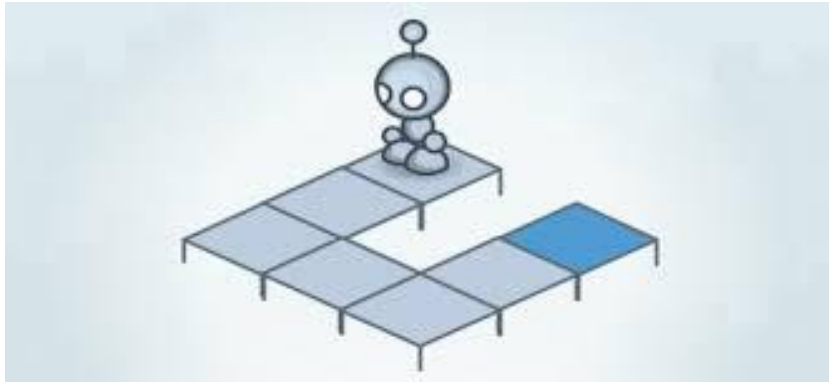
- Web
- iOS
- Android
- Windows Phone
- Kindle
- Windows
- macOS



# Lightbot

Se trata de un robot que trabaja en un fábrica y tiene como misión pender todas las baldosas de color azul.

Para ello hay que indicarle al robot que trayecto debe seguir y cuándo debe prender una baldosa.



# Lightbot

Conjunto limitado de instrucciones:



**Avanzar**



**Girar a la izquierda**



**Girar a la derecha**



**Saltar**



**Encender o apagar baldosa**

# Pantalla Lightbot

Reiniciar el nivel

Ejecutar el programa

Volver a la selección de nivel

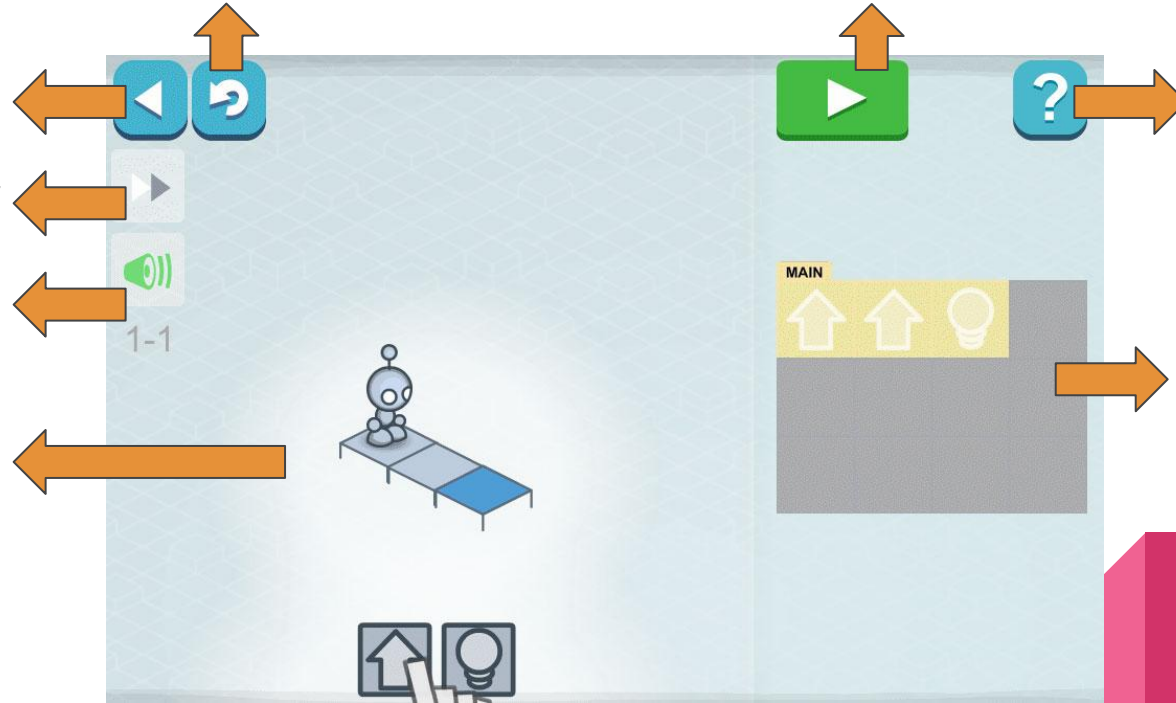
Activar/Desactivar cámara rápida

Activar/Desactivar música

Plataforma, estado de las luces y posición actual del robot

Ayuda


Descripción del programa Lightbot



# LightBot: Programa

Definimos un programa LightBot como la descripción de las acciones que el robot realiza cuando se presiona el botón ejecutar.

Un programa Lightbot tiene una cantidad limitada de acciones que el robot puede realizar (dependientes del nivel).



# Estados

Definimos el **estado actual** como el estado de las luces en la plataforma, la posición del robot y la dirección a la que mira.

Definimos el **estado final** como el estado en el que termina la plataforma y la posición del robot tras terminar de ejecutar completamente el programa.





# Propósito

El propósito de un programa es aquello que el programa busca resolver o intenta lograr. Es decir, es el **estado final esperado**.

En el caso de Lightbot, el propósito es siempre “prender todas las luces de la plataforma”.



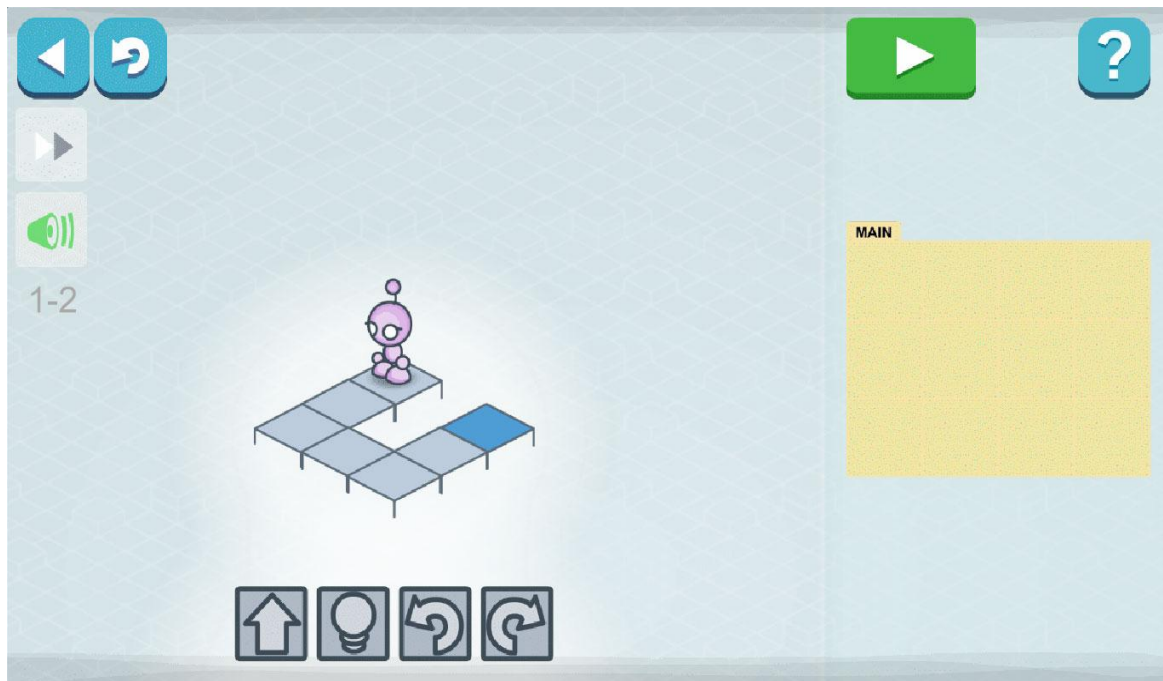
# Ejercicio para precalentar



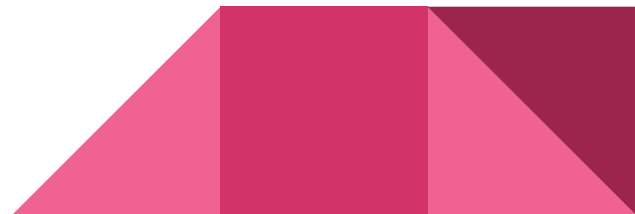
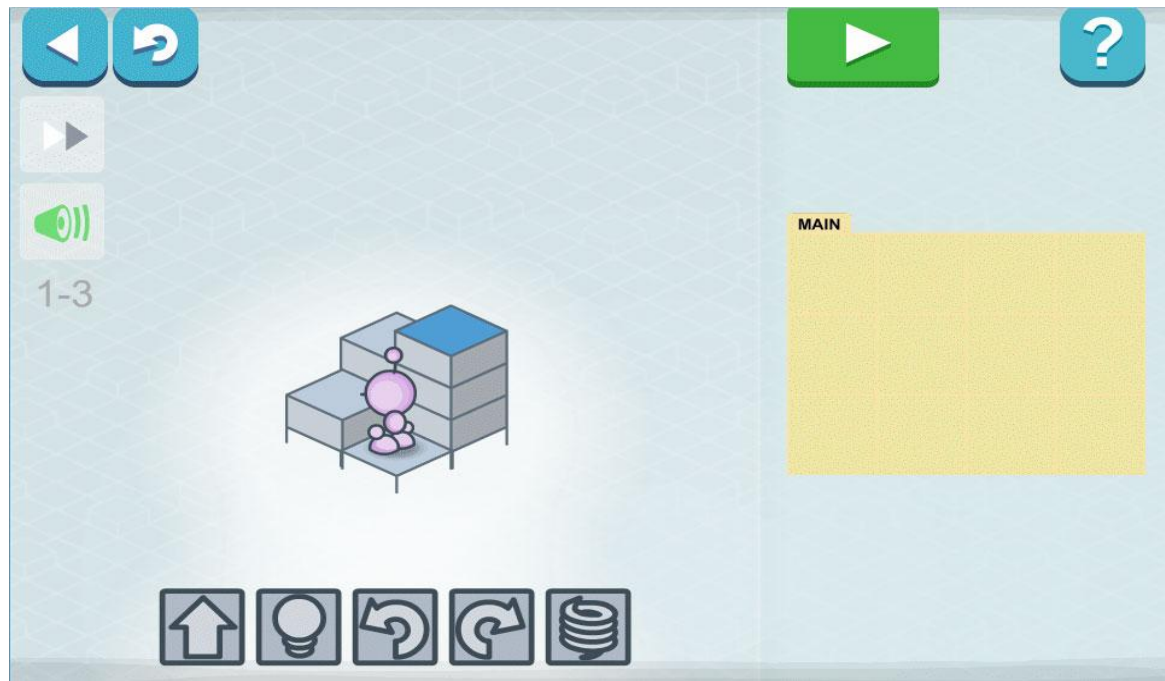
Resolvemos entre todxs algunos niveles de Lightbot...



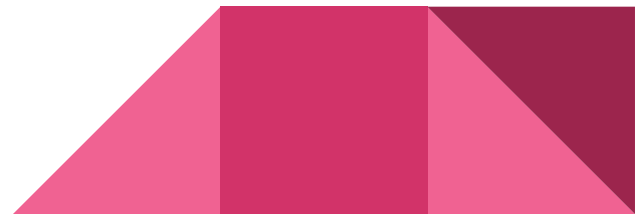
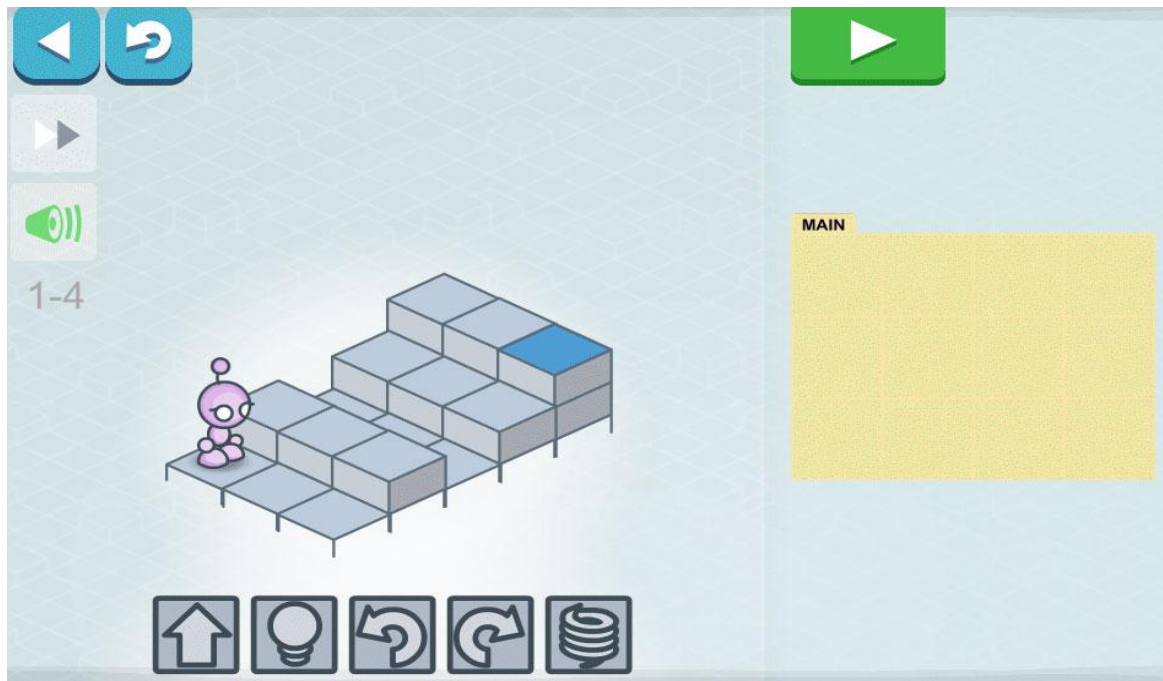
# Nivel 2



# Nivel 3



# Nivel 4



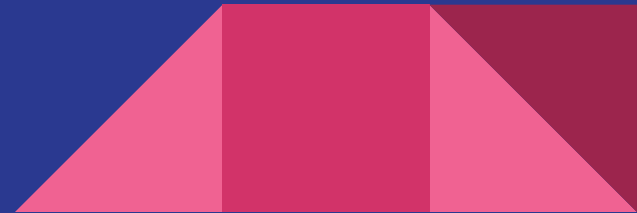
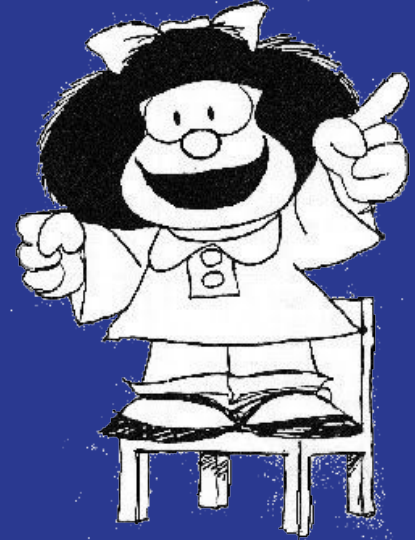
# Ejercitamos un poco más

El resto de los niveles para resolver individualmente.



Para reflexionar...

"La vida es un 10% lo  
que nos ocurre, y un  
90% cómo reaccionamos  
a ello"



# Programación

Clase 1

Universidad Nacional de Quilmes