

Programación

¿Qué es la programación?
Primeros autómatas

Universidad Nacional de Quilmes

Comenzando el día

¿Qué cosas hicieron a la mañana/tarde/noche para venir a la clase de EPyL?

¿Importa el orden en el cual realizamos las actividades?

Por ejemplo, una persona podría bañarse, luego desayunar, tomar el colectivo, entrar a la universidad y finalmente ir hasta el aula, y otra podría hacerlo en el orden inverso. Primero desayunar, luego bañarse, salir para la universidad, entrar y buscar el aula. Estas situaciones no implican un problema en el orden en que se realicen.

Pero, ¿qué pasa si antes de entrar a la universidad se quiere primero entrar al aula? ¿Tiene sentido? ¿Es posible?

Veamos otro ejemplo más completo.



Ejemplo sencillo

Tarea 'Cebiar mate': ¿en qué orden se deben realizar sus acciones?



Cebar mate: listamos las acciones

Para lograr nuestro propósito debemos seguir una serie de pasos **ordenados**, es decir una **secuencia**:

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

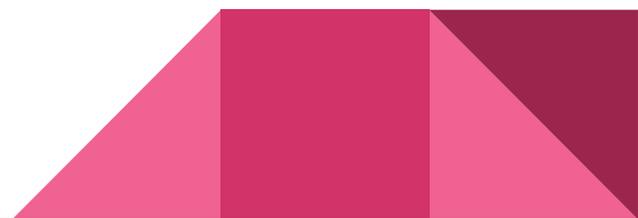
Paso 11: Colocar café en la taza

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

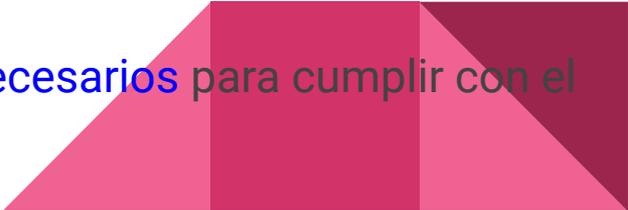
Paso 11: Colocar café en la taza

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

Un **algoritmo** es una serie ordenada de pasos, es decir una secuencia, para lograr un objetivo.

En algunos algoritmos, el orden de su secuencia puede variar sin que esto afecte el resultado, mientras que en otros casos esto no es posible.



¿Qué es la programación?



Etapas de la programación

La tarea de programar tiene, al menos 3 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

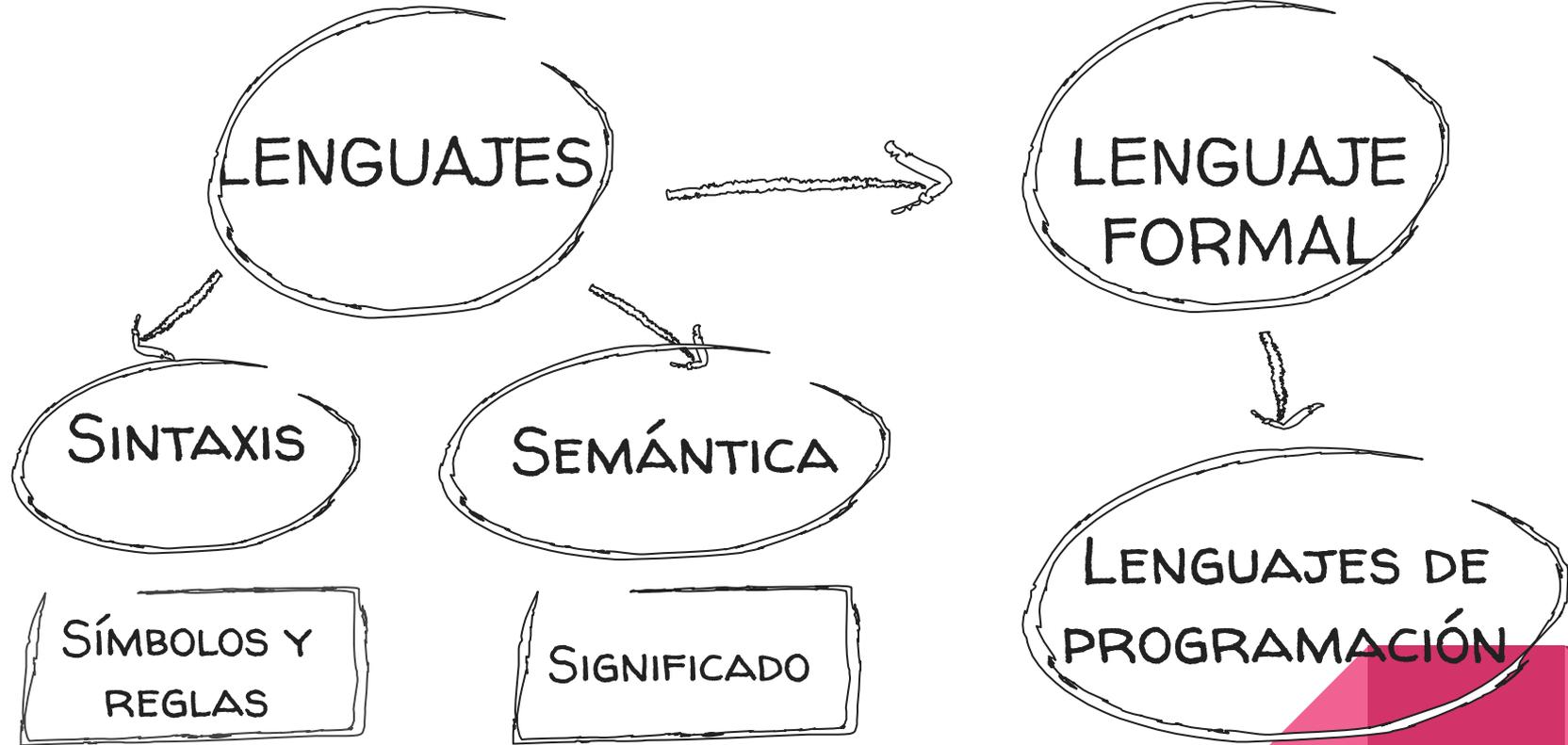
Se puede entender a la acción de programar, como el hecho de **comunicar** descripciones de soluciones a problemas en un **lenguaje** particular. Y para comunicar dichas descripciones utilizamos **algoritmos**.

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

Un programa debe estar redactado en un **lenguaje formal**.



Pero entonces, ¿qué es un lenguaje?



Lenguaje de programación

Se deberá entender que la solución escrita en un programa, justamente está escrito en un lenguaje que la máquina/autómata puede interpretar.

Es importante ver la diferencia entre código fuente y código máquina (código objeto).

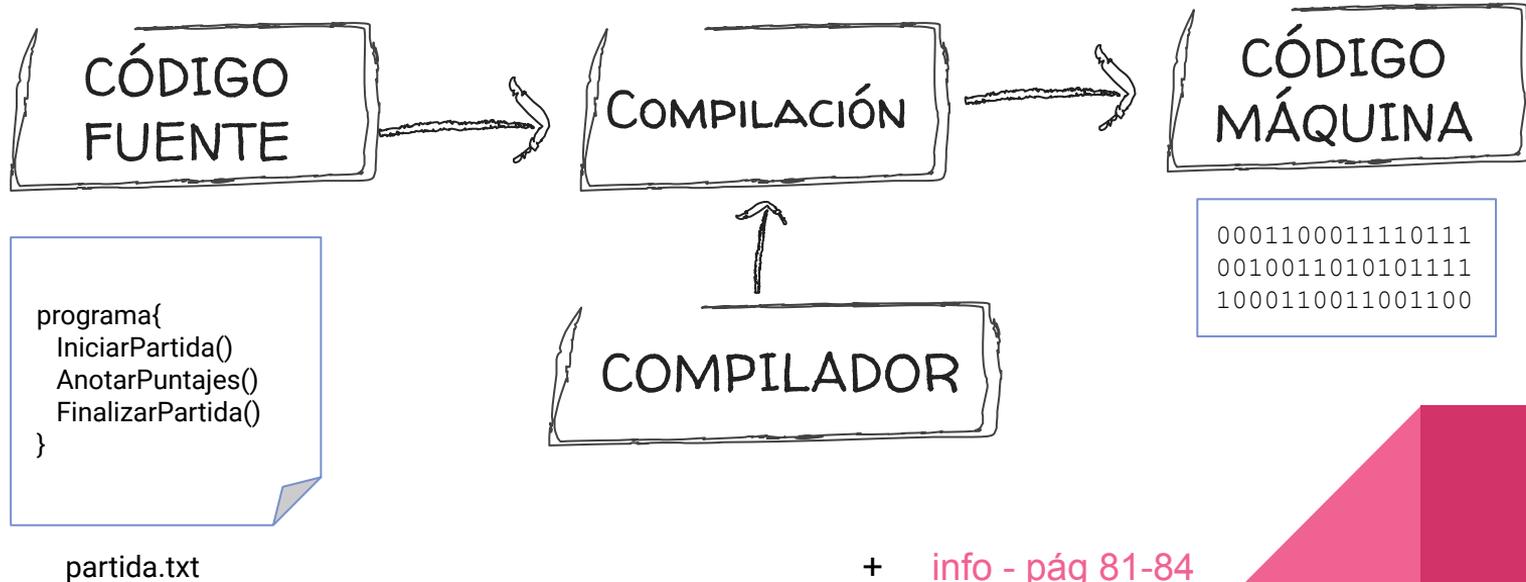
Lenguaje de programación y sus códigos

Quienes programamos utilizamos lenguajes de alto nivel (de abstracción) para escribir la solución a un problema, es decir que se genera un **código fuente**, el cual se acerca más a nuestro lenguaje natural.

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

Código fuente - Código máquina

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



¿Qué hace un/a programador/a?

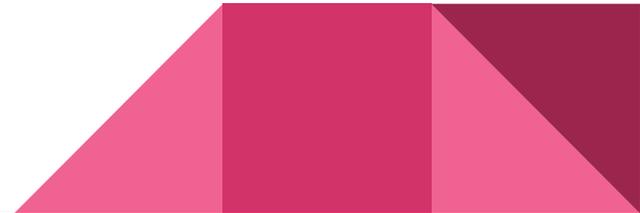
Escribe, depura y mantiene el código fuente de **programas** informáticos, es decir, el conjunto de instrucciones que ejecuta el hardware de una computadora, para realizar cumplir con un propósito, y así solucionar un problema dado.



¿Nos interesan todos los problemas?

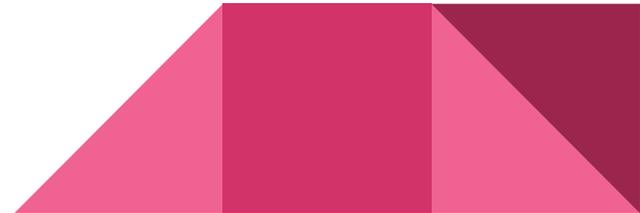
NO. Sólo aquellos que tienen una solución que puede ser calculada por una computadora (**Autómata**).

Es por eso que se dice que resuelve problemas **computables**.

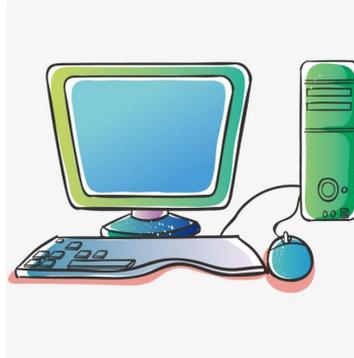
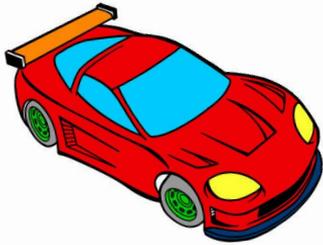


Autómatas

- Entienden un lenguaje determinado, acotado de instrucciones.
- La máquina no decide qué hacer, ejecuta el algoritmo siguiendo al pie de la letra sus instrucciones.
- Las instrucciones deben ser específicas, concretas y denotar una orden.



Ejemplos de autómatas



Entonces...

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



Programar es Comunicar

- Comunicamos al autómeta
- Comunicamos a otras personas
- Comunicamos a nosotras/os mismas/os para su mantenimiento futuro



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 conocer el lenguaje en que programamos y las buenas prácticas de programación. Además, muchos problemas requieren conocimientos lógicos y matemáticos complejos.
 - **Abstracción:** Reconocer estructuras y elementos en la computadora implica poder abstraer la realidad a modelos matemáticos o mentales para poder operarlos.
- 

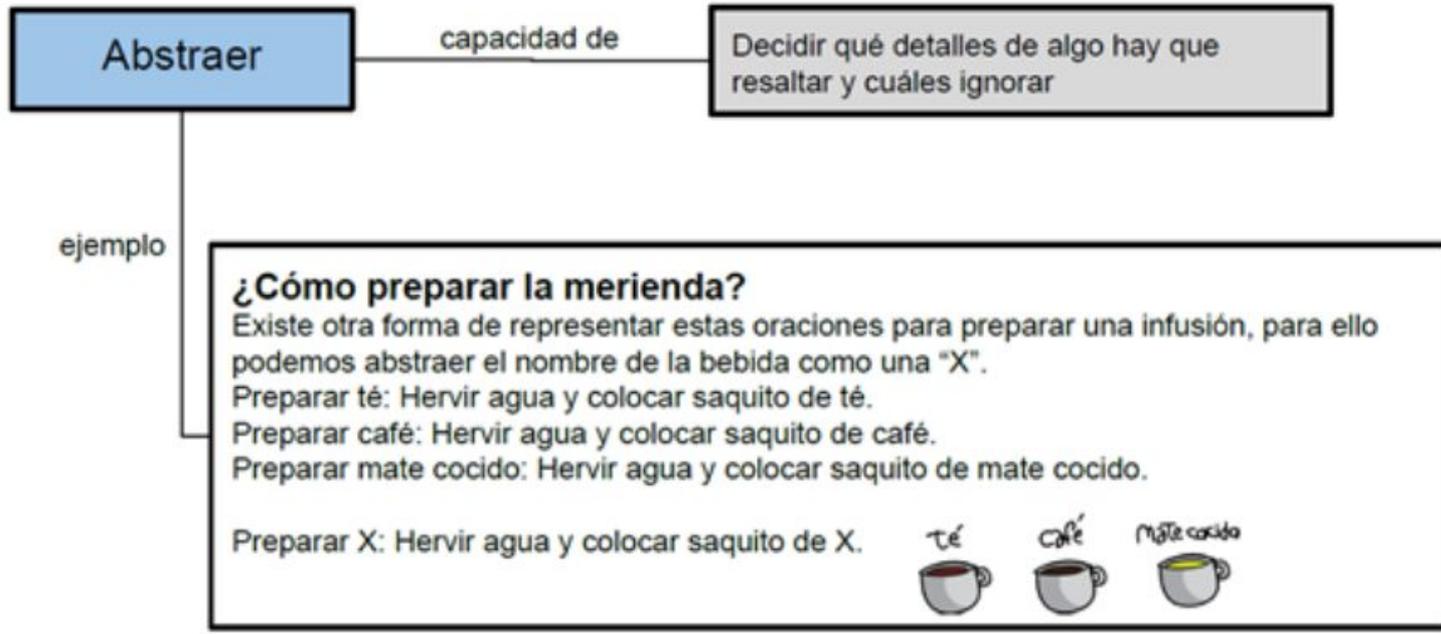
Conceptos esenciales de programación

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)

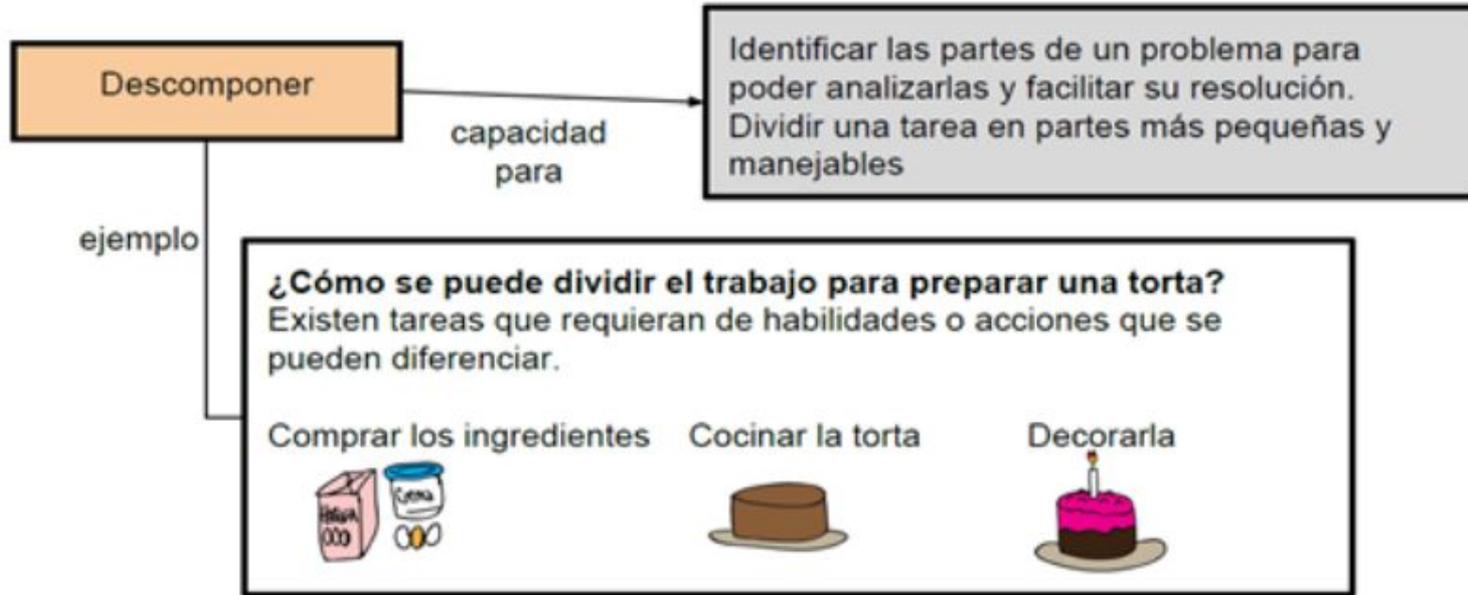
Abstracción

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



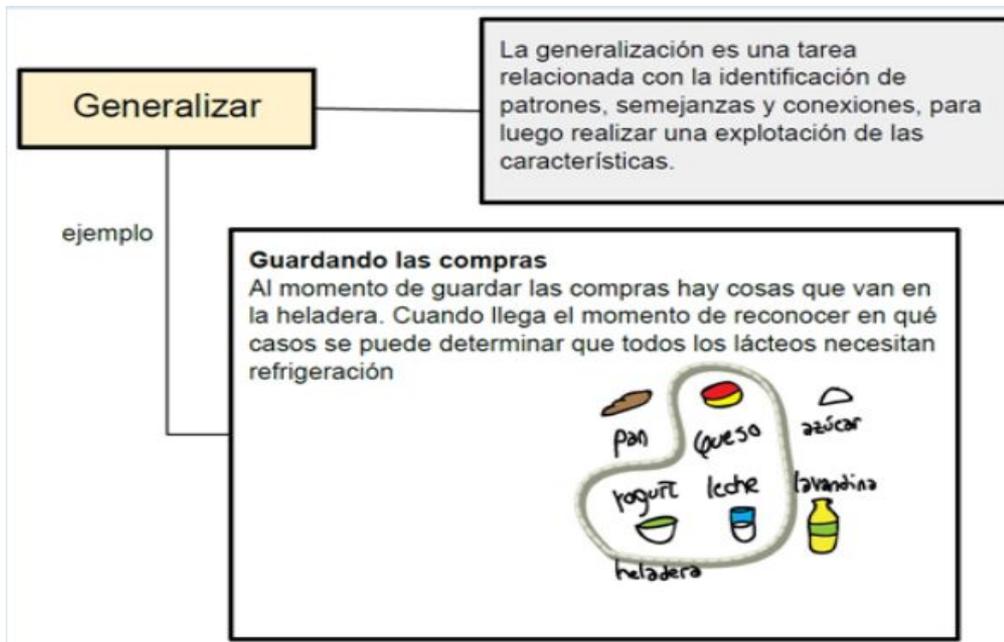
Descomposición

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.



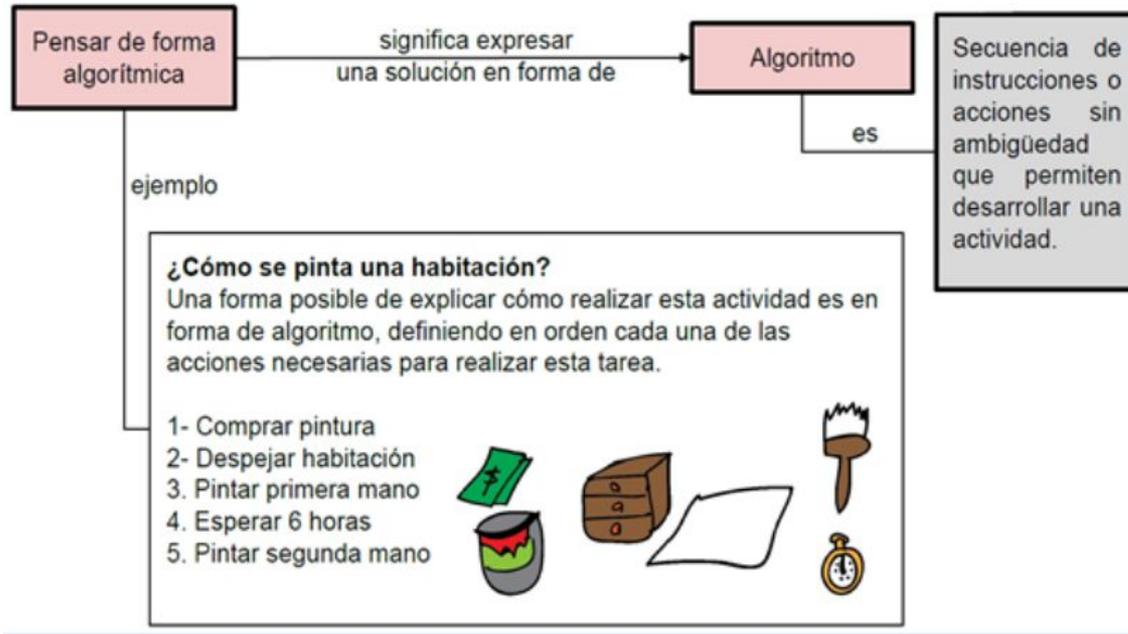
Generalización

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.



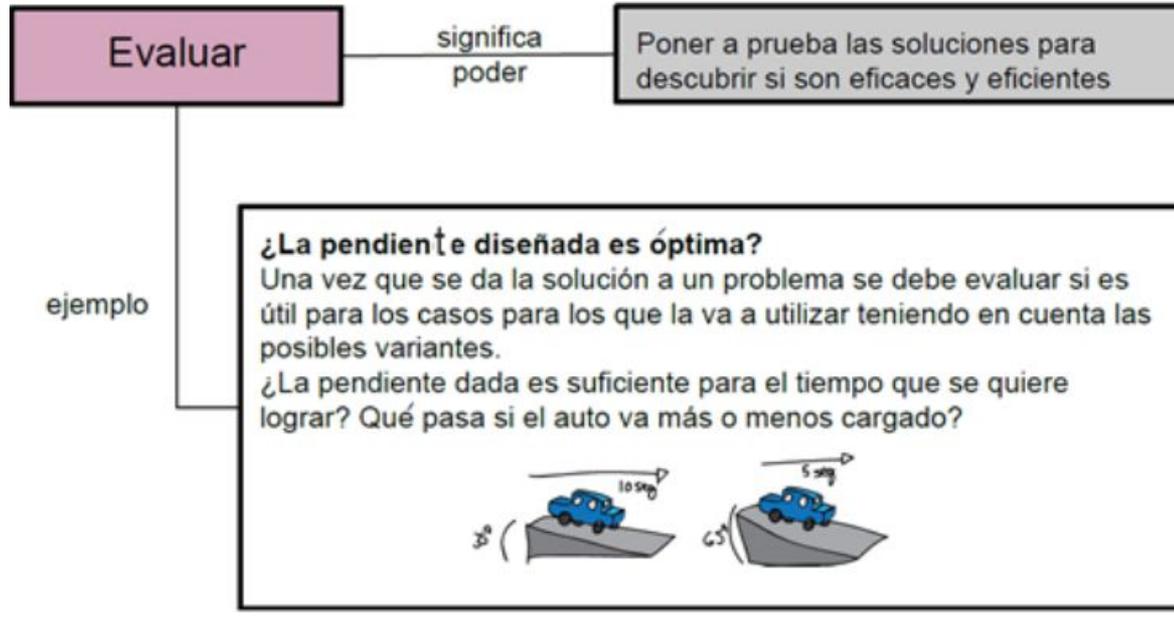
Algoritmia

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 autómata pueda llevarlo adelante.

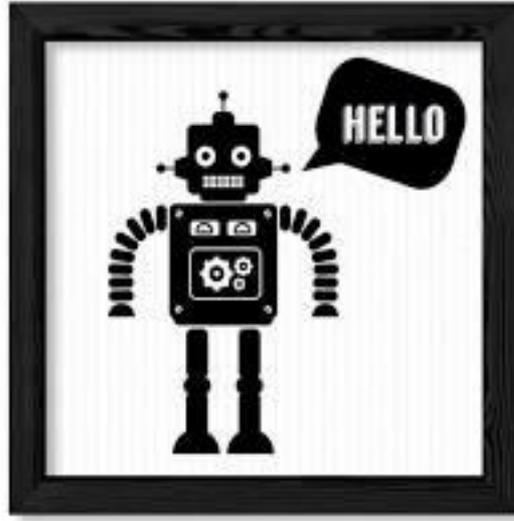


Evaluación (Probar)

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.

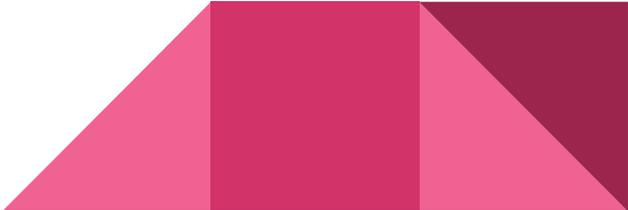


Robot Humano



Instrucciones

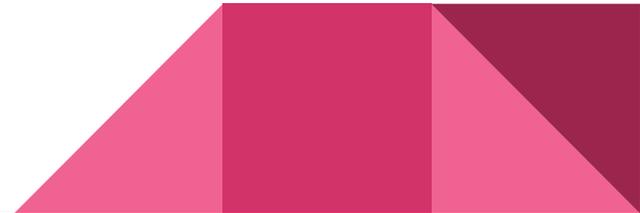
Supongamos que el robot contiene este set limitado de instrucciones:

- Avanzar un paso
 - Girar a la derecha
 - Girar a la izquierda
 - Abrir la ventana
 - Sentarse en una silla
 - Levantar el brazo derecho
 - Levantar el brazo izquierdo
- 

Ejemplo de algoritmo

Por ejemplo, le pedimos al robot que cumpla con el siguiente algoritmo:

1. Avanzar un paso
2. Levantar el brazo izquierdo
3. Abrir la ventana
4. Sentarse en una silla



Robot = Autómata

Ahora ¿Qué ocurriría en los siguientes casos?

- Se le pide al robot que levante el brazo izquierdo, y ya lo tiene levantado
 - Se le ordena al robot abrir una ventana y no hay ventana, o la misma ya se encuentra abierta
 - Se le ordena al robot que se siente en una silla, pero no hay ninguna silla
- 

Algoritmo completo

Reformulamos el algoritmo, agregando la siguiente información:

1. Avanzar un paso
2. Levantar el brazo izquierdo
3. Abrir la ventana
4. Sentarse en una silla

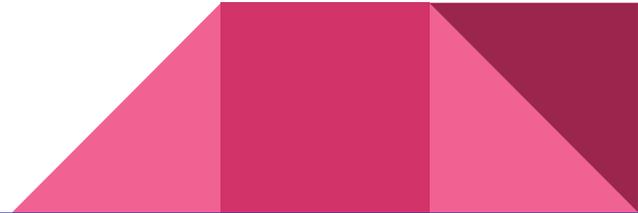
Propósito: Abrir una ventana

Precondición:

- Debe existir una ventana a un paso del robot
- La ventana debe estar cerrada
- Debe haber una silla
- El robot tiene su brazo izquierdo abajo



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

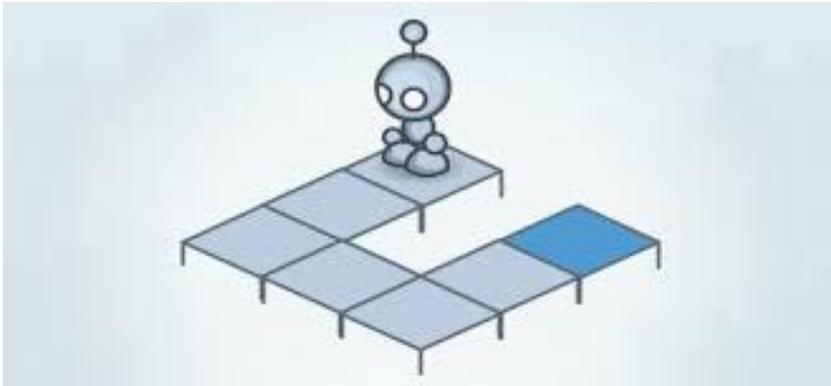
- Web
- iOS
- Android
- Windows Phone
- Kindle
- Windows
- macOS



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

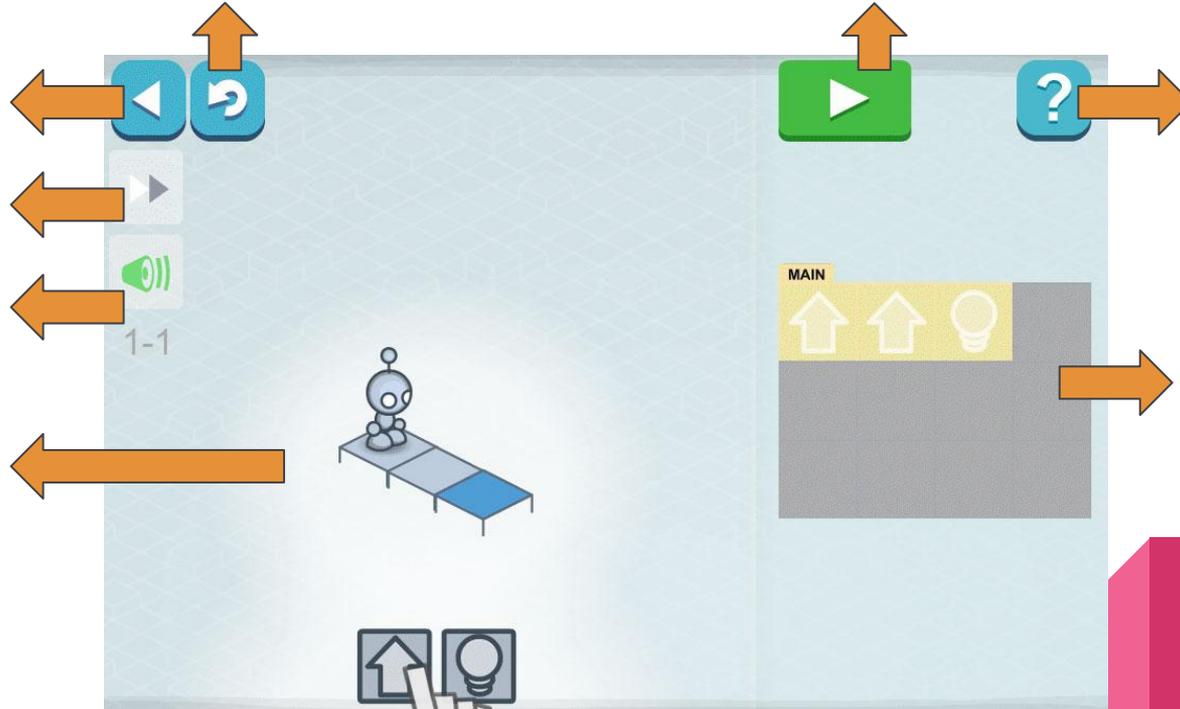
Activar/Desactivar cámara rápida

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.

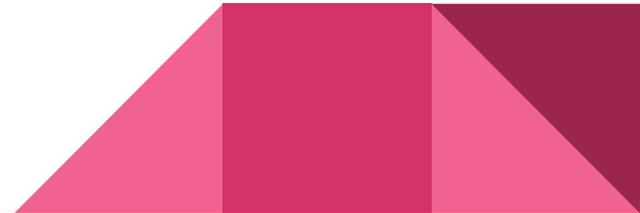
Un programa Lightbot tiene una cantidad limitada de acciones que el robot puede realizar (dependientes del nivel).



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 el objetivo final.

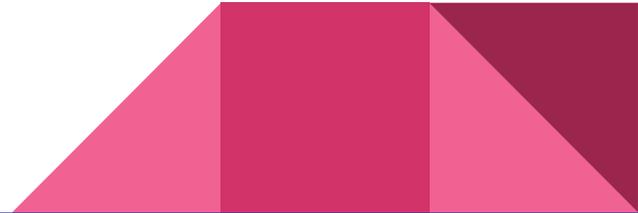
En el caso de lightbot es el estado final esperado. Y puntualmente siempre es “prender todas las luces de la plataforma”.



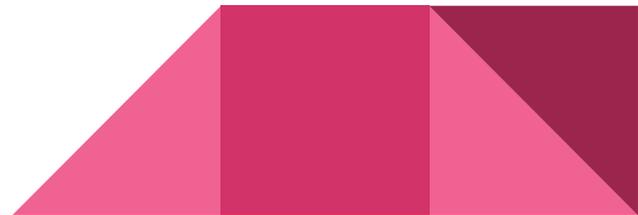
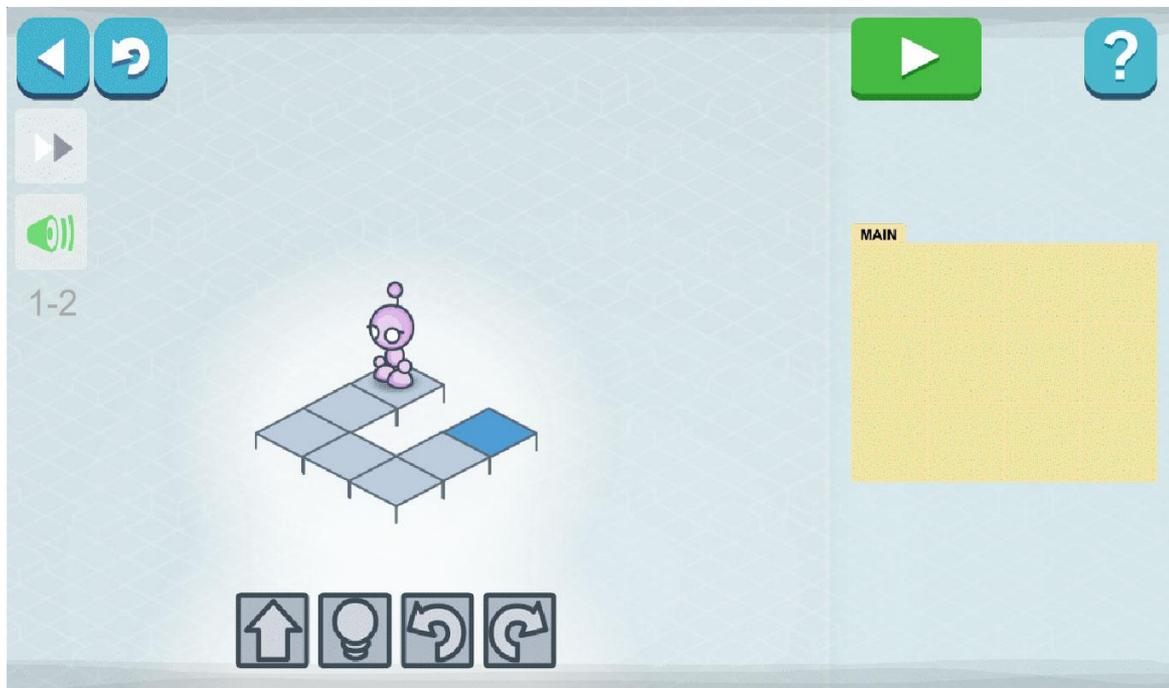
Ejercicio para precalentar



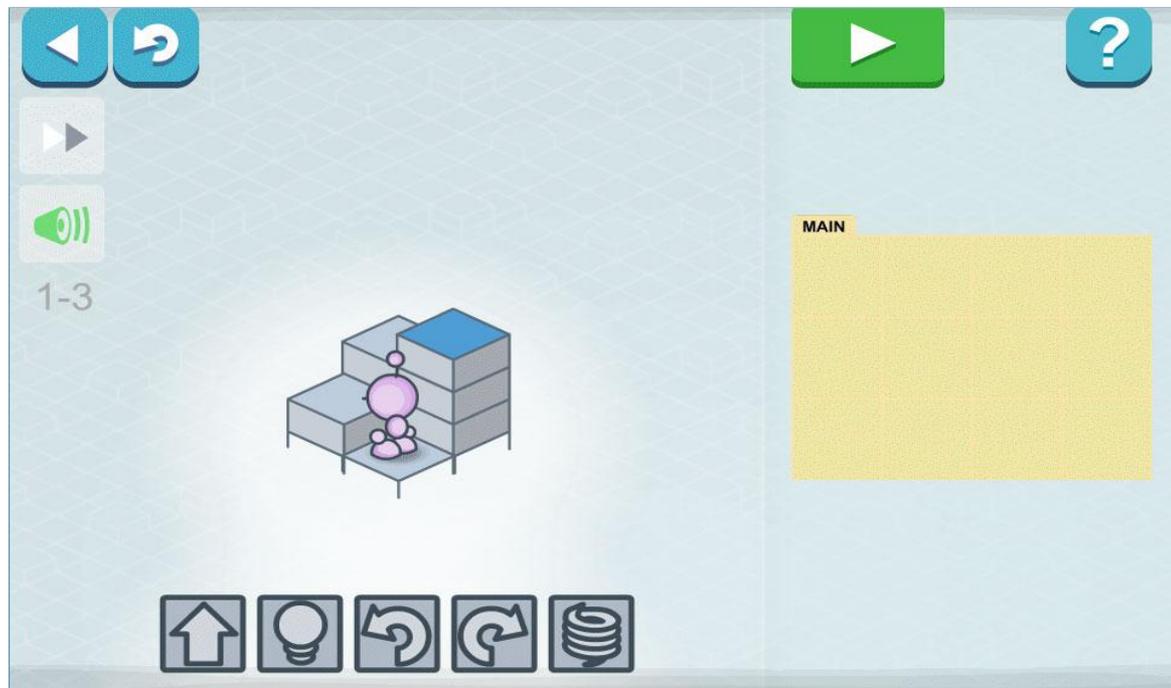
Resolvemos entre todes algunos niveles de Lightbot...



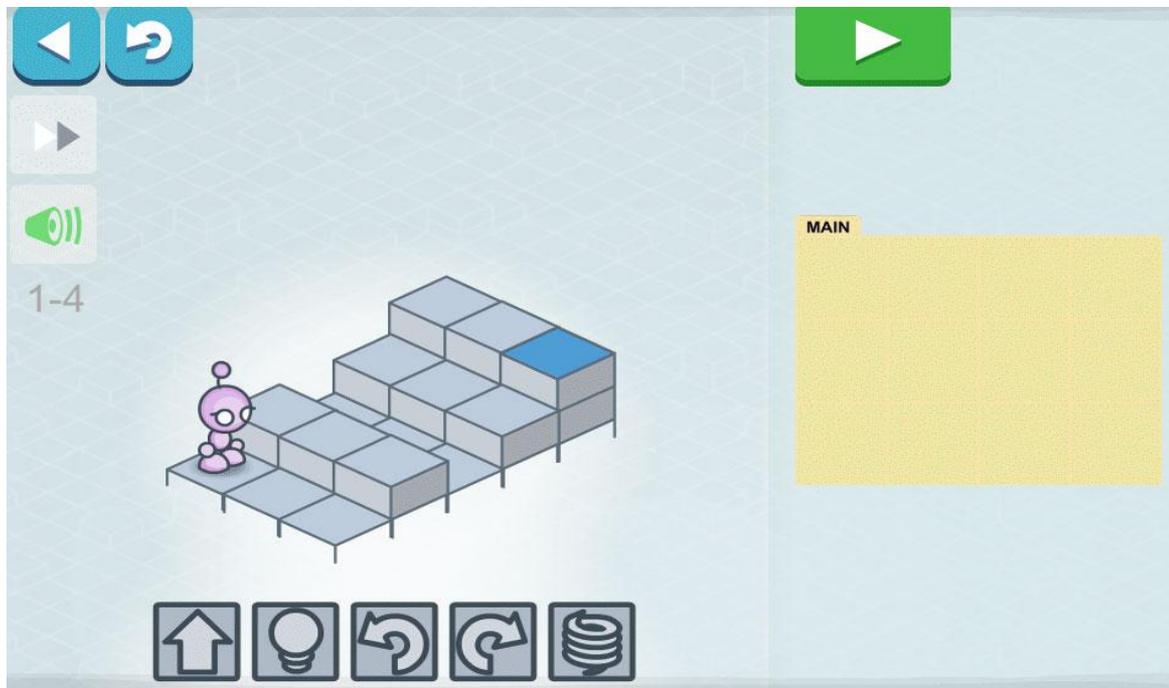
Nivel 2



Nivel 3

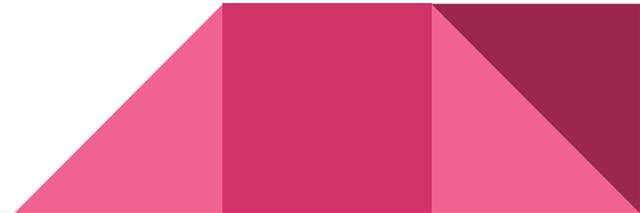


Nivel 4



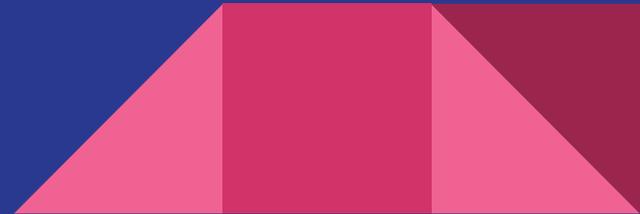
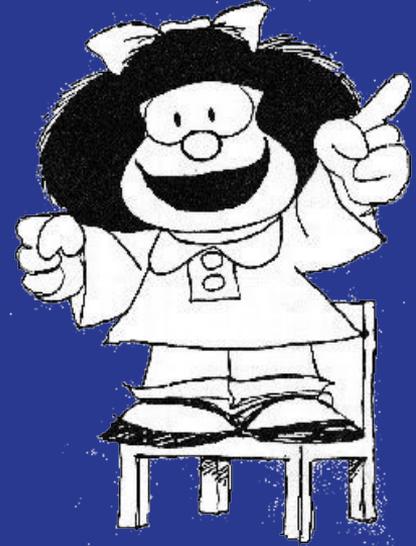
Ejercitamos un poco más

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

Introducción

Universidad Nacional de Quilmes