

Programación y pensamiento computacional

TP: 1. Repeticiones y programación sin computadoras

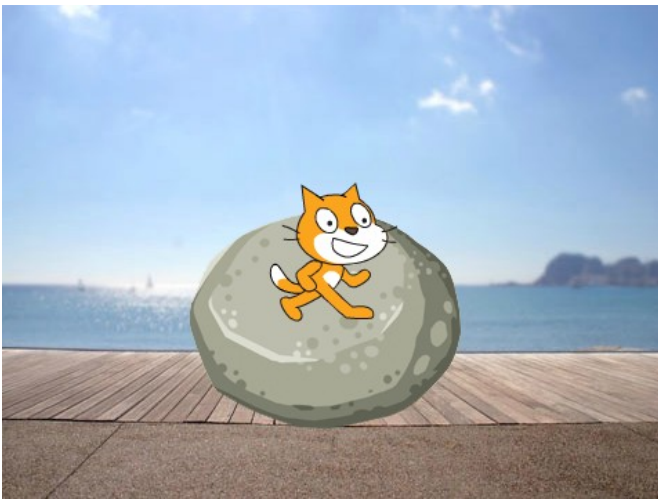
Problema 2: No me canso de saltar

Consigna: que el gato salte 30 veces para llegar a la última piedra

Observaciones

- *El gato avisa cuántos saltos le faltan*
- *Solo contamos con la primitiva Saltar...*

Escenario



Primitivas

[illegible]

Herramienta/concepto necesario

- **Repetición simple:**
 - *sirve para repetir una tarea*

Actividad:

- **Volver a intentar la solución usando repetición simple**



Reflexiones

¿Cuántos comandos fue necesario escribir para resolver el problema usando repetición simple?

- Las tareas repetitivas se pueden reducir al uso de una repetición
- La cantidad de veces, por el momento, es fija

Con buenas herramientas de programación es posible ser muy expresivo

Problema 3: programando en cuadriculado

Actividad sin computadoras

- **Comprenderemos**
 - Cómo formular instrucciones de manera adecuada, según un modelo de ejecución dado.
- **Ejercitaremos**
- Los principios de programación ya vistos, pero en otro modelo de ejecución.

CONSIGNA:

- Realizar distintas actividades con el modelo de programación en papel cuadriculado.
- Observaciones
 - Utilizaremos 2 hojas cuadriculadas una para los programas ("Programas") y
 - otra para la ejecución ("Dibujos").
- Nuestros programas representarán pasos para hacer el dibujo
- Los comandos primitivos son los siguientes



- Se puede repetir un grupo de comandos
 - Para ello se lo pone entre paréntesis, seguido de un número
 - Ejemplo: "Pintar hacia la derecha dos cuadraditos"

(⚡→) 2

- **Podemos crear procedimientos**
 - Se le da nombre a un grupo de comandos
- **Ejemplo:** “Dibujar ventana”

```
(  ↗  ↑ ) 3
(  ↘  → ) 3
(  ↙  ↓ ) 3
(  ↖  ← ) 3
```

- Los procedimientos definidos se pueden usar en la definición de otros procedimientos.
 - Se pone su nombre entre corchetes
- Ejemplo:

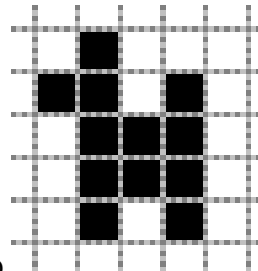
“Dibujar ventana doble”

[Dibujar ventana]

→ 3

[Dibujar ventana]

- **Actividad:**
 - **Escribir un programa que realice este dibujo**
 - **Ayuda:**
 - Pensar una estrategia
 - Dividir en subtarear y representarlas como procedimientos (ej. “Dibujar cabeza”)



Reflexiones

¿Usaron subtareas para los posicionamientos?

¿Buscaron aprovechar la repetición al descomponer?

Actividad:

Revisar el programa para ver si puede mejorarse

Sugerencia:

Cada procedimiento podría dejar la posición en el mismo lugar en que empezó

Una posible solución

“Dibujar perro”

[Dibujar cabeza]

[Ir a cuerpo desde cabeza]

[Dibujar cuerpo]

[Ir a pata delantera desde cuerpo]

[Dibujar pata]

[Ir a pata trasera desde delantera]

[Dibujar pata]

[Ir a cola desde pata trasera]

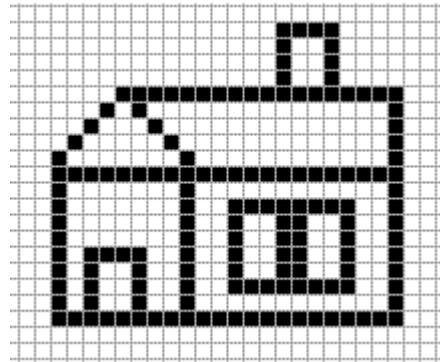
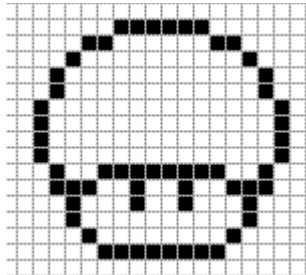
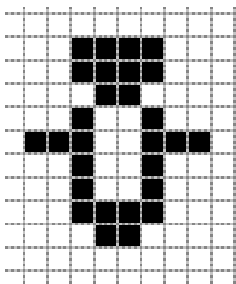
[Ir a dibujar cola]

Actividad:

- Modificar el programa para que el perro tenga las patas más largas
- Modificar el programa para que dibuje un perro del doble de tamaño
- **Observaciones:**
 - ¿Qué partes hay que modificar?
 - ¿El programa principal tiene que cambiar?

Actividad:

- Escribir algunos programas más



Actividad:

- Inventar un dibujo simple
- Escribir un programa que lo dibuje
- Intercambiar el programa con un compañero y tratar de hacer el dibujo usando solo con el programa (sin ayuda)
- Sugerencia:
 - Intentar que las subtareas sean significativas

Reflexiones

¡No solo las computadoras ejecutan programas!

- Al ejecutar un programa de otro
 - ¿ayudó la división en subtareas a entender lo que hacía?
 - ¿los nombres de los procedimientos ayudaron?
- **Legibilidad**
- ¡Los programas tienen que ser leídos por máquinas y también por personas!