

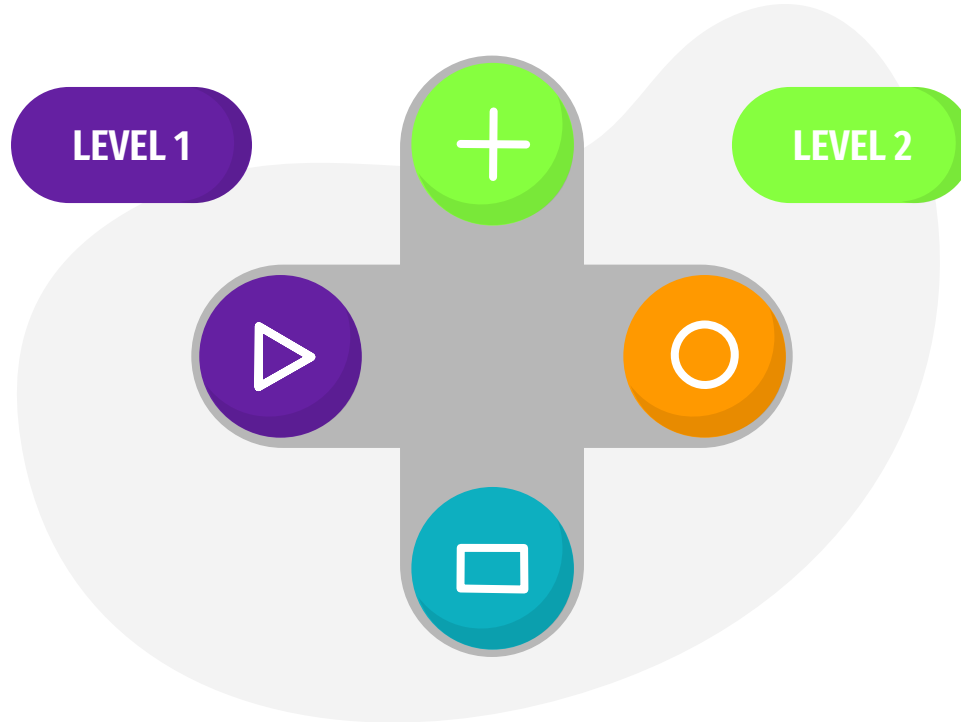


# Fundamentos de Programación en Videojuegos I

Clase 4

# Contenido de la clase

Programando en papel.  
Importancia de programar  
en papel. Sintaxis para  
escribir programas en  
papel: palabras clave.



EJERCITARIO 2 -  
PRIMEROS  
PROGRAMAS:  
PRIMITIVAS Y  
PROCEDIMIENTOS

**Programando en papel. Importancia de programar en papel.  
Sintaxis para escribir programas en papel: palabras clave.**



# Límites de la herramienta actual

- Pilas Bloques es una excelente herramienta para el aprendizaje de la programación inicial. Pero, la cantidad de ejercicios que hay es limitada. A veces queremos más ejercicios.
- Existen otras herramientas para aprendizaje, pero cambiar de entorno y acostumbrarse al mismo lleva un tiempo y esfuerzo que no es deseable invertir.
- La computadora está buena, pero su dependencia extrema es peligrosa. ¿Estás pensando los ejercicios o simplemente encajando bloques hasta que funcional

# Las ventajas de programar en papel

- No dependemos de ninguna herramienta, y podemos tener tantos ejercicios como nuestra imaginación (y la propuesta didáctica) lo permita.
- Nos permite trabajar comprensión de enunciados más complejos, a veces sin el componente gráfico que brinda la computadora, lo cual es muy importante que desarrollen como estudiantes universitarios.
- Elimina la dependencia de la computadora, y nos obliga a pensar sí o sí sobre lo que estamos haciendo, ya que no podemos sacar las cosas por “prueba y error”.

# La dualidad de la programación

- Cuando programamos, existe siempre una especie de paradoja que nos debe dominar.
- El programa debe poder ser ejecutado por una computadora, por lo que la computadora subyacente es muy importante. A su vez, la computadora subyacente no es relevante (caja negra) y queremos programas que puedan funcionar en cualquier computadora, por lo que lo más importante es que el programa cumpla otras características (plantee adecuadamente la estrategia de solución, sea claro y legible, utilice las herramientas del lenguaje y las primitivas de forma apropiada, etc.).
- Cuando trabajamos en máquina corremos el riesgo de enfocarnos mucho en que las cosas “funcionen” sin prestar atención a la parte verdaderamente importante, la de usar bien los conceptos que venimos trabajando.
- Al trabajar en papel, se elimina el factor de pensar en que funcione, y nos fuerza a pensar en usar bien los conceptos.

## Pero debería funcionar...

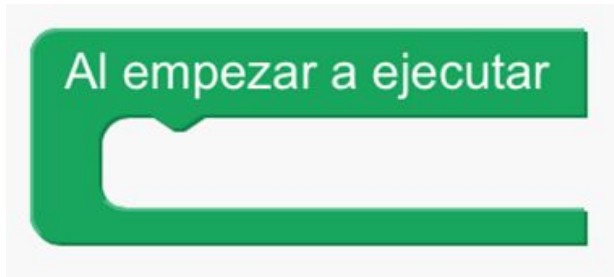
- El hecho de que no vaya a ser ejecutado no significa que podamos hacer cualquier cosa.
- La idea es escribir programas de forma tal que si estuviéramos ejecutando en una computadora, ese programa debería funcionar.
- Es decir, la idea de ejecución mediante secuencia y repetición, cuándo un programa va a fallar, etc. se debe tener en cuenta, a pesar de que no haya una computadora que vaya a ejecutar las cosas.

# ¿Cómo escribimos un programa en papel?

- Dibujar bloques no es práctico, por lo que necesitamos una forma de escribir las cosas de forma sencilla, marcando claramente la jerarquía de los elementos.
- Por jerarquía nos referimos a “qué bloques están dentro de qué otros bloques”, es decir, como están anidados.
- Además debemos dejar en claro los bloques que son parte del lenguaje, como la repetición simple, el punto de entrada, o las definiciones de procedimientos.
- Vamos a usar la idea de “palabras clave”, cosas que queremos escribir todos de la misma forma, porque tienen una semántica puntual bien clara, ej “al empezar a ejecutar”

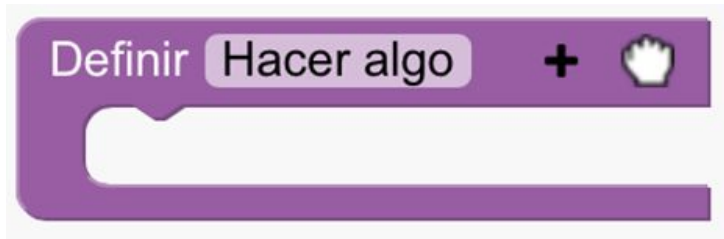


# Palabras claves



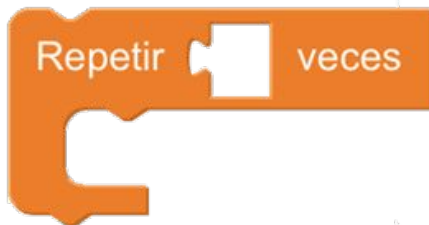
`Al empezar a ejecutar`

`| <cuerpo>`



`Definir <nombre del procedimiento>`

`| <cuerpo>`



`Repetir <expresión numérica> veces`

`| <cuerpo>`

# Expresiones

7



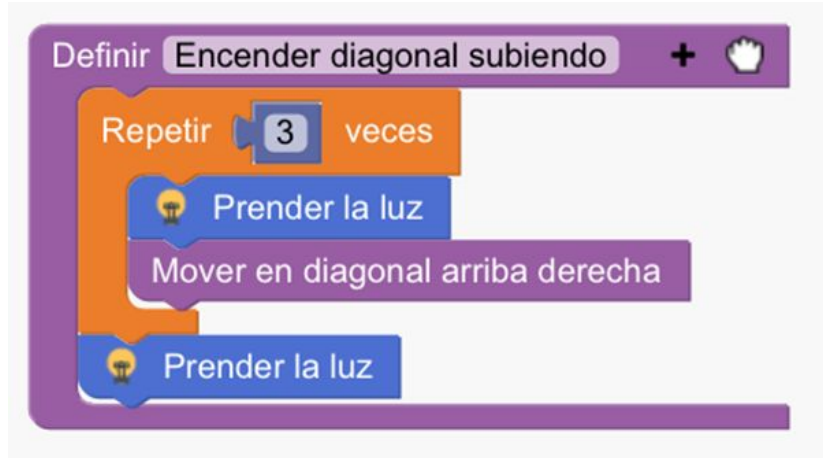
es decir, solamente escribimos el número y listo

# Comandos y primitivas



- **Mover a la derecha**
- es decir, escribimos el nombre, tal cual fue dado en las primitivas o pusimos en la definición de un procedimiento que realizamos.

# Jerarquía



## Definir Encender diagonal subiendo

Repetir 3 veces

Prender la luz

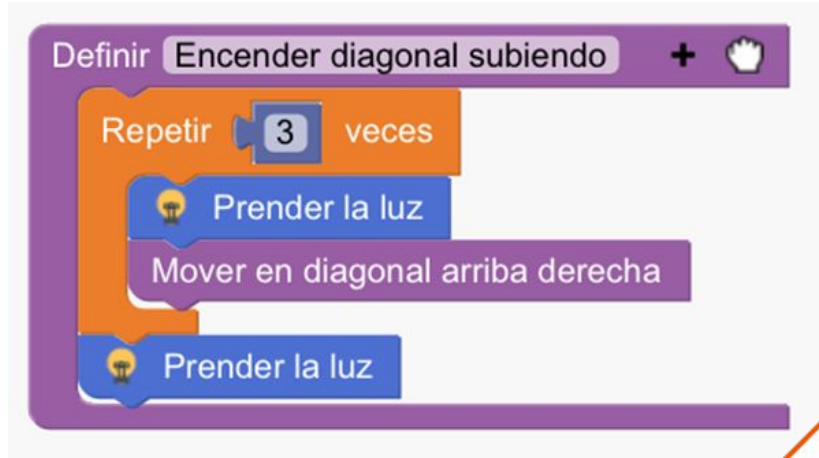
Mover en diagonal arriba derecha

Prender la luz

Usamos líneas verticales para dejar en claro la jerarquía, y que quede bien marcado qué cosa está dentro de otra.

La línea vertical termina cuando terminaría el bloque. La escribimos justo debajo del elemento con cuerpo, y no pegamos a la misma los elementos que están dentro de ese cuerpo, para que sea bien visible.

# Jerarquía



Esta línea es la del bloque del procedimiento, y deja en claro que todos los elementos están dentro de la definición.

## Definir Encender diagonal subiendo

Repetir 3 veces

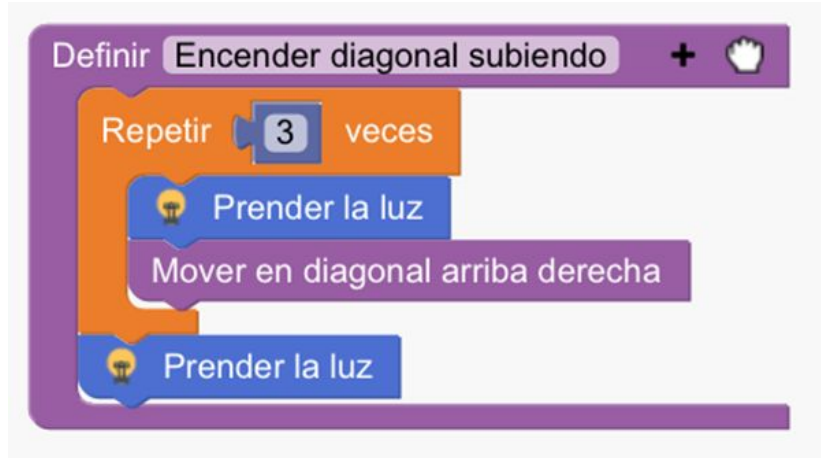
Prender la luz

Mover en diagonal arriba derecha

Prender la luz

Esta es la línea del bloque de la repetición, y deja en claro que los textos "Prender la luz" y "Mover en diagonal arriba derecha" están dentro de esa repetición.

# Jerarquía



Definir Encender diagonal subiendo

Repetir 3 veces

Prender la luz

Mover en diagonal arriba derecha

Prender la luz

A partir de las líneas, queda claro que este último “Prender la luz” está dentro del procedimiento, pero fuera de la repetición.

# Un ejemplo más complejo



**Al empezar a ejecutar**

**Repetir 3 veces**

**Tomar fila de estrellas**

**Ir al inicio de siguiente fila**

**Tomar fila de estrellas**

**Definir Tomar fila de estrellas**

**Repetir 4 veces**

**Mover a la derecha**

**Tomar estrella**

**Definir Ir al inicio de la siguiente fila**

**Ir al borde izquierdo**

**Mover arriba**

# Algunas consideraciones adicionales

- Escribimos una definición abajo de otra (no usamos varias columnas)
- El orden en el que definimos los procedimientos no importa, aunque preferimos que primero esté el “al comenzar a ejecutar” y luego aparezcan los procedimientos que este utiliza, seguidos de los procedimientos que usan los anteriores y así siguiendo.
- Lo ideal es hoja cuadriculada, pero no pasa nada si se usa rayada o lisa. Sí es importante ser prolijos.
- No es problema si tenés fea letra, escribí en imprenta y listo.
- No hace falta usar regla para las líneas, suelen ser cortitas, (y sí no, sospecha que te están faltando procedimientos) salvo que realmente no tengas pulso en absoluto, en cuyo caso opta por regla.



# ¿Qué pinta tiene un ejercicio en papel?

Un ejercicio en papel va a plantearnos varias partes:

- Va a darnos un enunciado, en el cual nos va a describir los elementos involucrados en el problema. Por ej. el enunciado podría contarnos que hay un marciano, al que se le rompió la nave, y que debe conseguir tuercas para arreglarla.
- Luego se nos va a indicar qué problema se desea resolver en el programa, por ej. “escriba un programa que haga que el marciano junte todas las tuercas del escenario”.
- Puede que además se nos presente el escenario en un dibujo, mostrando la disposición inicial de los elementos. Incluso podrían haber más de un escenario de ejemplo si el ejercicio lo requiere. Pueden además haber una muestra del escenario luego de ejecutar (para mayor claridad en lo que se espera que el programa realice). Podría darse el caso de que no hayan dibujos, y el escenario esté descrito en texto.
- Se va a proveer el conjunto de primitivas que pueden utilizarse para resolver el problema (las primitivas son propias de cada ejercicio, a diferencia de los elementos del lenguaje, como los procedimientos o la repetición, que siempre los vamos a poder usar). No podemos usar primitivas de un ejercicio en otro, tenemos que ver cuáles son las primitivas disponibles. Además se nos va a dar una descripción de la semántica de cada una de las primitivas, para evitar ambigüedades.

## Ejemplo 1: Charlie en su circo

Charlie es el dueño de un circo, y se encuentra practicando un peligroso y nuevo acto, en el que caminará por una alfombra verde esquivando pelotas gigantes que giran velozmente hacia él. Su objetivo es llegar a un podio que se encuentra a varios lugares hacia la derecha de su ubicación inicial. A continuación se muestra el escenario inicial.

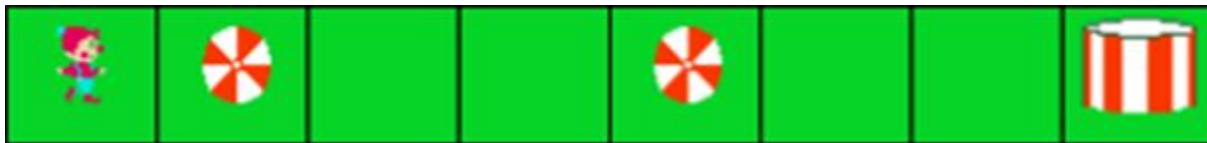


Se pide que escriba un programa que lleve a Charlie hasta el podio, esquivando todas las pelotas en el camino.

# Ejemplo 1: Charlie en su circo

Para resolver el problema, existen las siguientes primitivas:

- **Mover a la derecha:** Mueve a Charlie un lugar a la derecha. El lugar al que se mueve debe estar vacío, sin pelota o podio. Si no hubiera más lugar, o a la derecha hubiera una pelota o el podio, entonces el programa falla.
- **Saltar la pelota que viene:** Hace que Charlie salte la pelota que se encuentra en el lugar inmediatamente a la derecha de él. Charlie terminará en la ubicación a la derecha de la pelota. Si no hubiera una pelota a la derecha, entonces el programa falla.
- **Subir al podio:** Hace que Charlie suba al podio que está a la derecha de él. Si no hubiera un podio a la derecha, entonces el programa falla.



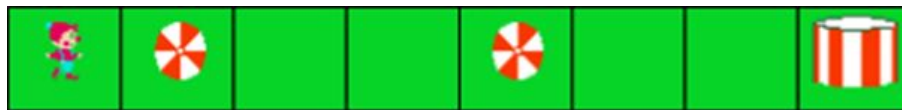
# Ejemplo 1: Charlie en su circo

Resumen:

Se pide que escriba un programa que lleve a Charlie hasta el podio, esquivando todas las pelotas en el camino.

- Primitivas

- **Mover a la derecha:**
- **Saltar la pelota que viene**
- **Subir al podio**



Solución:

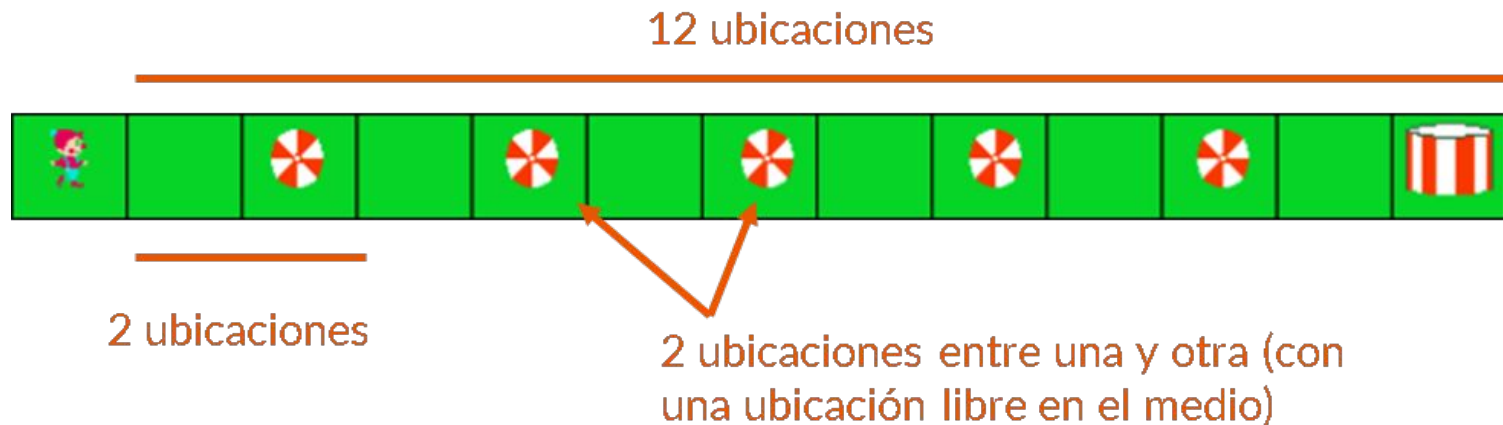
```
Al empezar a ejecutar
|
| Repetir 2 veces
| |
| | Saltar la pelota que viene
| | Mover a la derecha
| Subir al podio
```

## Ejemplo 2: Charlie en su circo 2

El acto de Charlie empieza a tomar forma, y se vuelve cada vez más complicado. Ahora las pelotas aparecen más seguido, y el podio se encuentra más lejos.

El podio está a exactamente 12 ubicaciones de distancia a la derecha. Las pelotas están a una distancia de dos ubicaciones entre ellas (con una ubicación libre en el medio entre una y otra), y la primera se encuentra a dos lugares a la derecha de Charlie.

Usando las mismas primitivas del ejercicio anterior, se pide que escriba un programa que lleve a Charlie hasta el podio, esquivando todas las pelotas del camino.



## Ejemplo 2: Charlie en su circo 2

Habiendo pensado el escenario, podemos empezar a resolver usando los elementos del lenguaje y las primitivas que tenemos.



Solución:

Al empezar a ejecutar

Mover a la derecha

Repetir 5 veces

Saltar la pelota que viene

Subir al podio

# ACTIVIDAD Y TAREA DE LA CLASE 4

**RESOLVER EL EJERCITARIO 2 - PRIMEROS PROGRAMAS: PRIMITIVAS Y  
PROCEDIMIENTOS**

