

# APRENDIENDO A PROGRAMAR

---

Con Scratch  
(<https://scratch.mit.edu/>)

# Pensamiento computacional

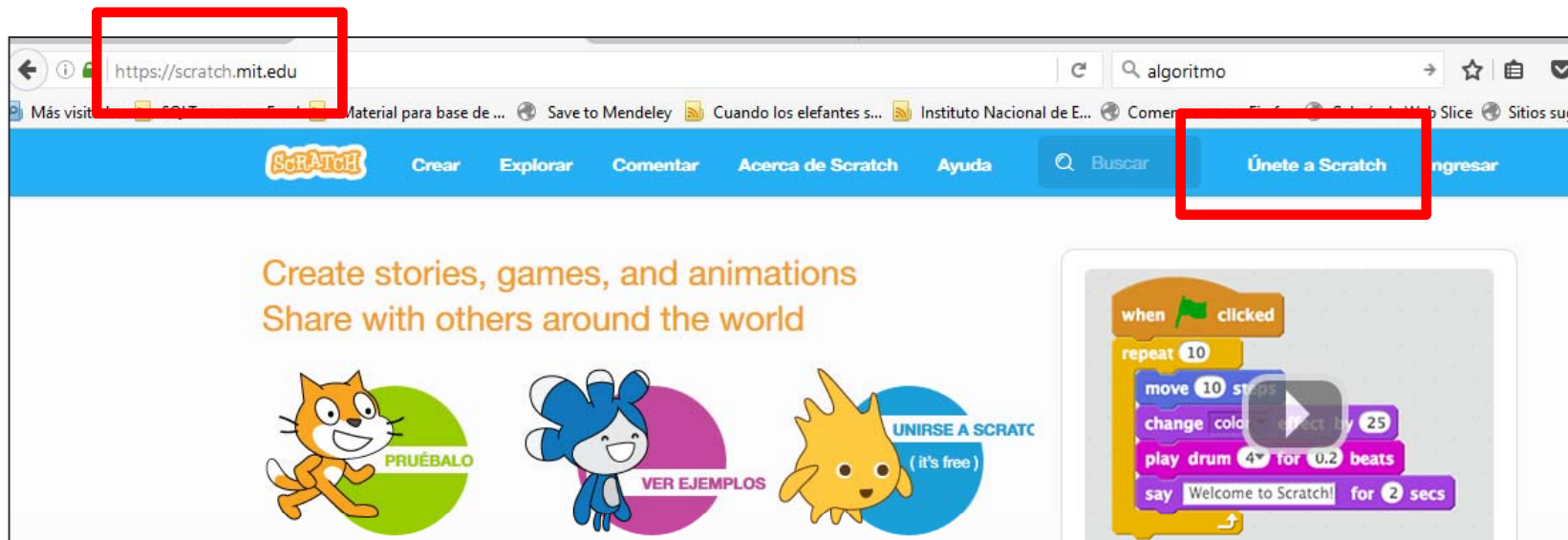
- Capacidad para la resolución algorítmica de los problemas.
- Imprescindible el conocimiento lógico-matemático.
- Si enseñamos a programar computadores en las escuelas, estaremos fomentando el pensamiento computacional y, por ende, el pensamiento lógico-matemático.

# Pensamiento computacional

- Basado en el concepto de “algoritmo”
  - Un conjunto ordenado y finito de operaciones que permite hallar la solución de un problema
  - Un conjunto prescrito de instrucciones o reglas bien definidas, ordenadas y finitas que permite realizar una actividad mediante pasos sucesivos que no generen dudas a quien deba realizar dicha actividad.

# ¿Por qué Scratch para iniciar a programar?

- **Scratch** es un entorno de programación creado en el *Massachusetts Institute of Technology*(MIT) como resultado de la investigación llevada a cabo en el *Lifelong Kindergarten Group*.



# ¿Por qué Scratch para iniciar a programar?

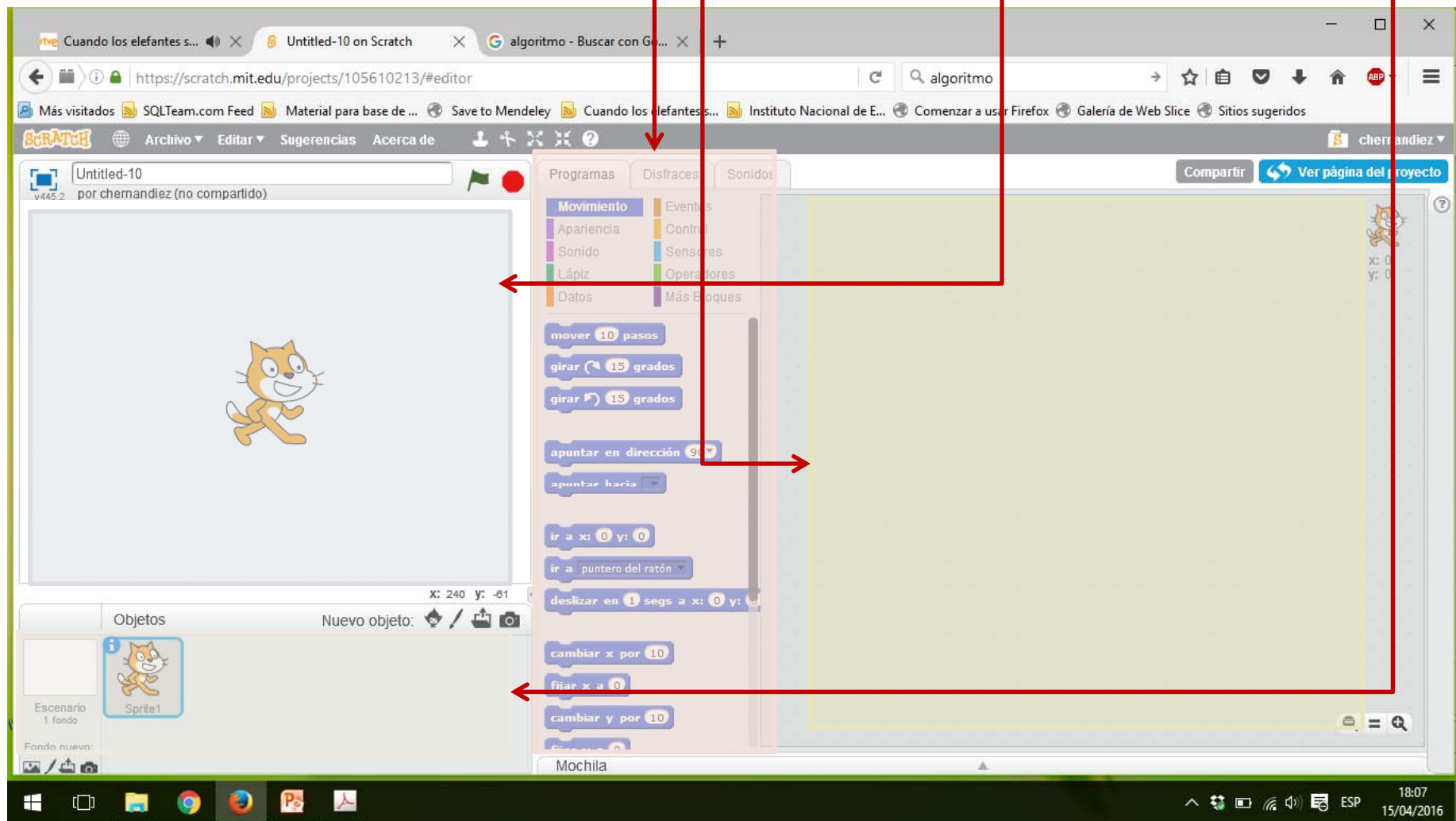
- Vamos a la página <https://scratch.mit.edu/>
- Nos unimos a la comunidad de Scratch

Zona de bloques

Zona de programación

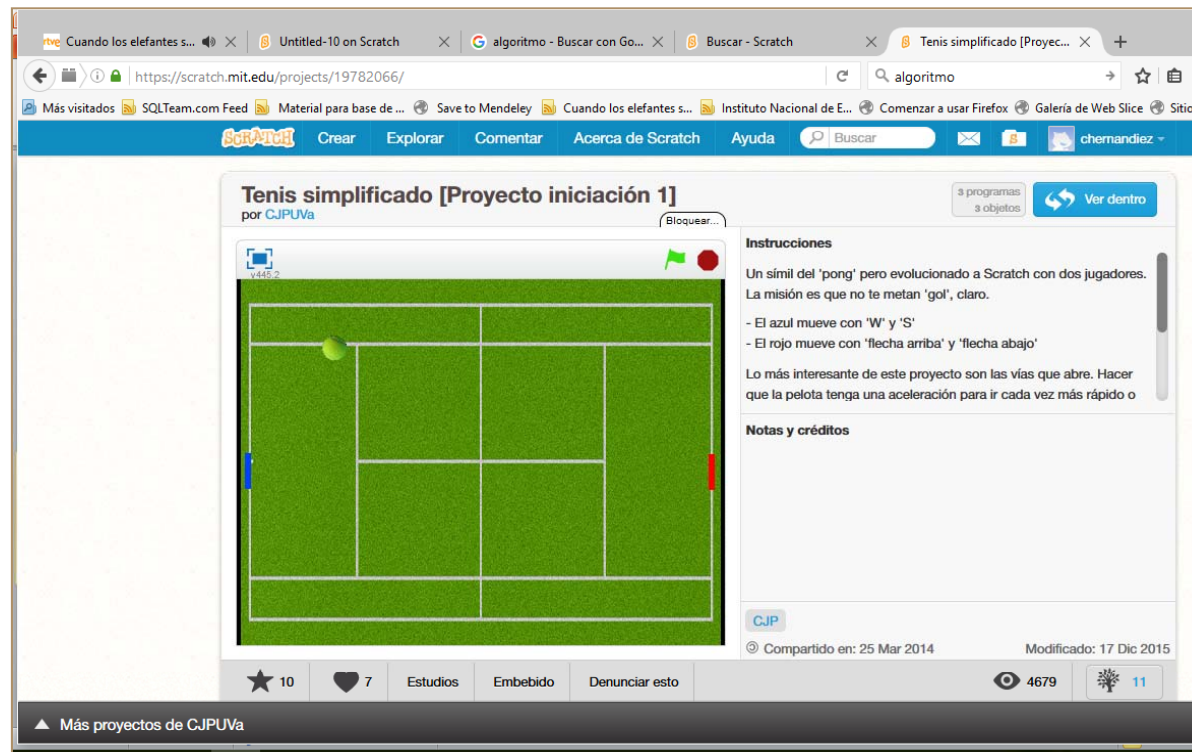
Zona de objetos “programables”

Escenario

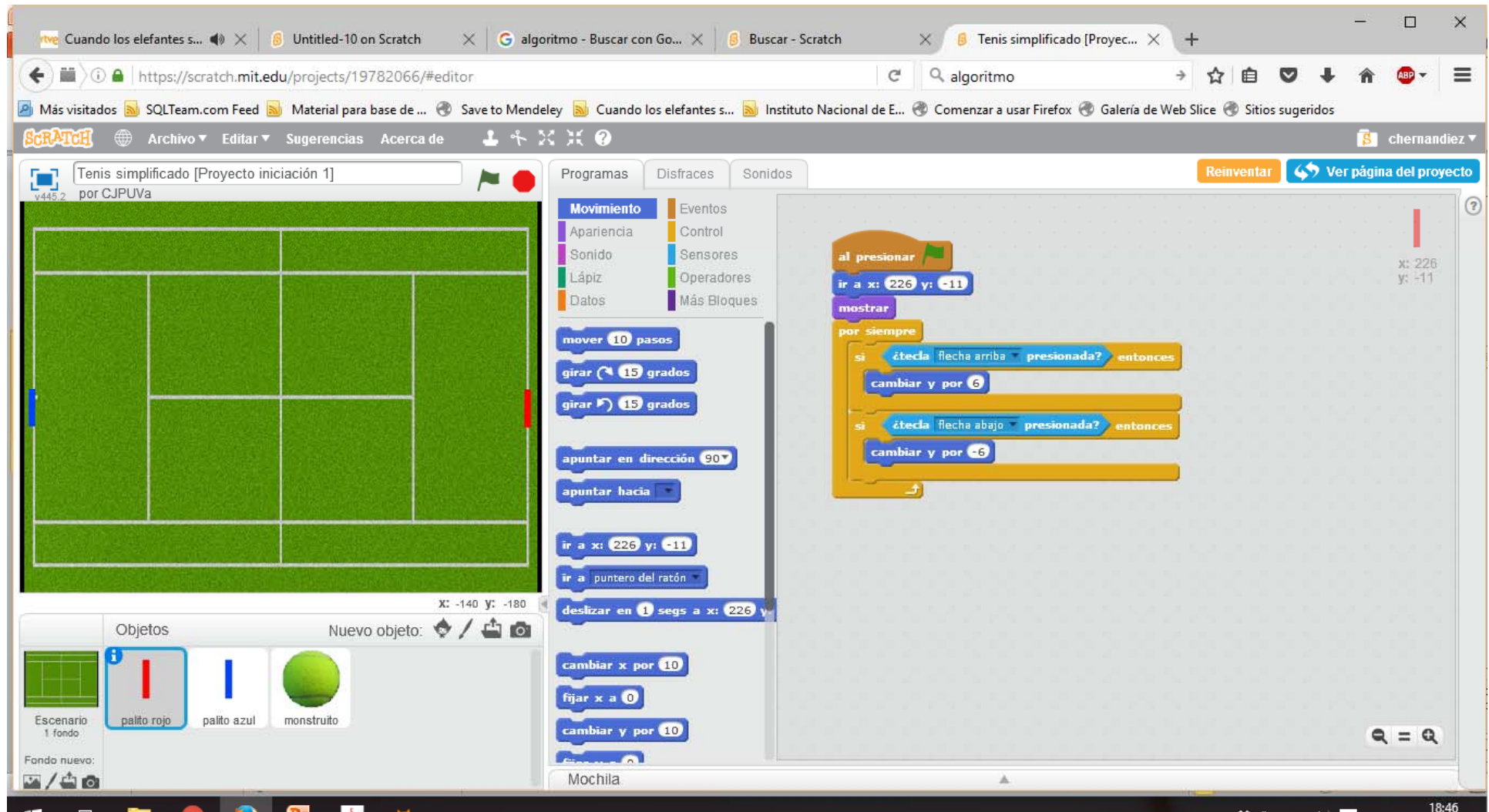


# Primer programa

- Juego de tenis
  - Buscar CJP@UVa
- <https://scratch.mit.edu/projects/19782066/>



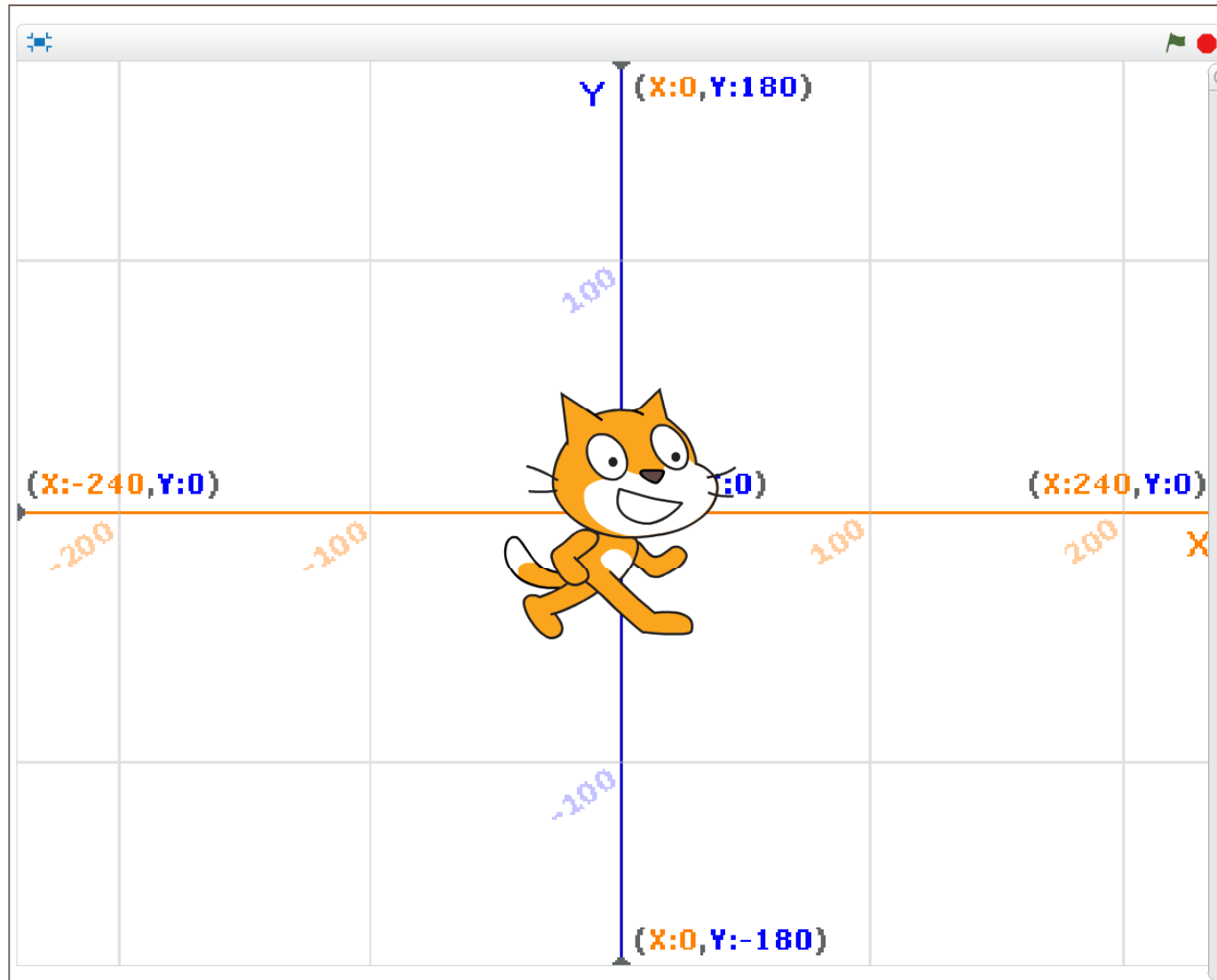




The screenshot shows the Scratch editor interface for a project titled "Tenis simplificado [Proyecto iniciación 1]". The stage displays a green tennis court with a net. The left sidebar shows the "Programas" (Scripts) category selected, with a list of blocks including "mover 10 pasos", "girar 15 grados", "apuntar en dirección 90", "apuntar hacia", "ir a x: 226 y: -11", "deslizar en 1 segs a x: 226 y: -11", "cambiar x por 10", "fijar x a 0", and "cambiar y por 10". The right sidebar shows the "Eventos" category selected, with a list of blocks including "al presionar", "ir a x: 226 y: -11", "mostrar", "por siempre", "si ¿tecla flecha arriba presionada? entonces", "cambiar y por 6", "si ¿tecla flecha abajo presionada? entonces", and "cambiar y por -6". The bottom left shows the "Objetos" (Sprites) area with a "palito rojo" (red stick) selected. The bottom right shows the "Mochila" (Inventory) area.

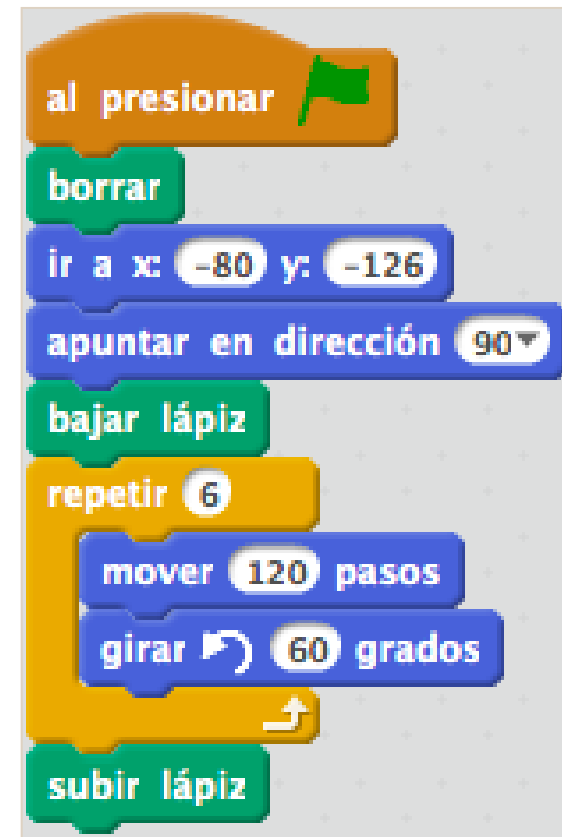
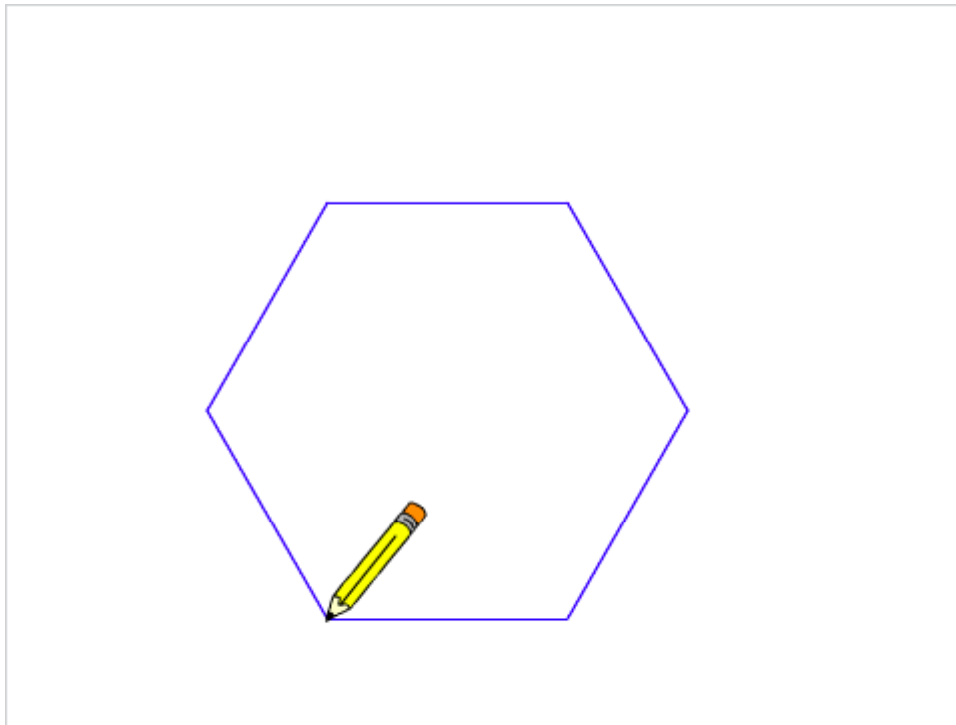


# XY-GRID

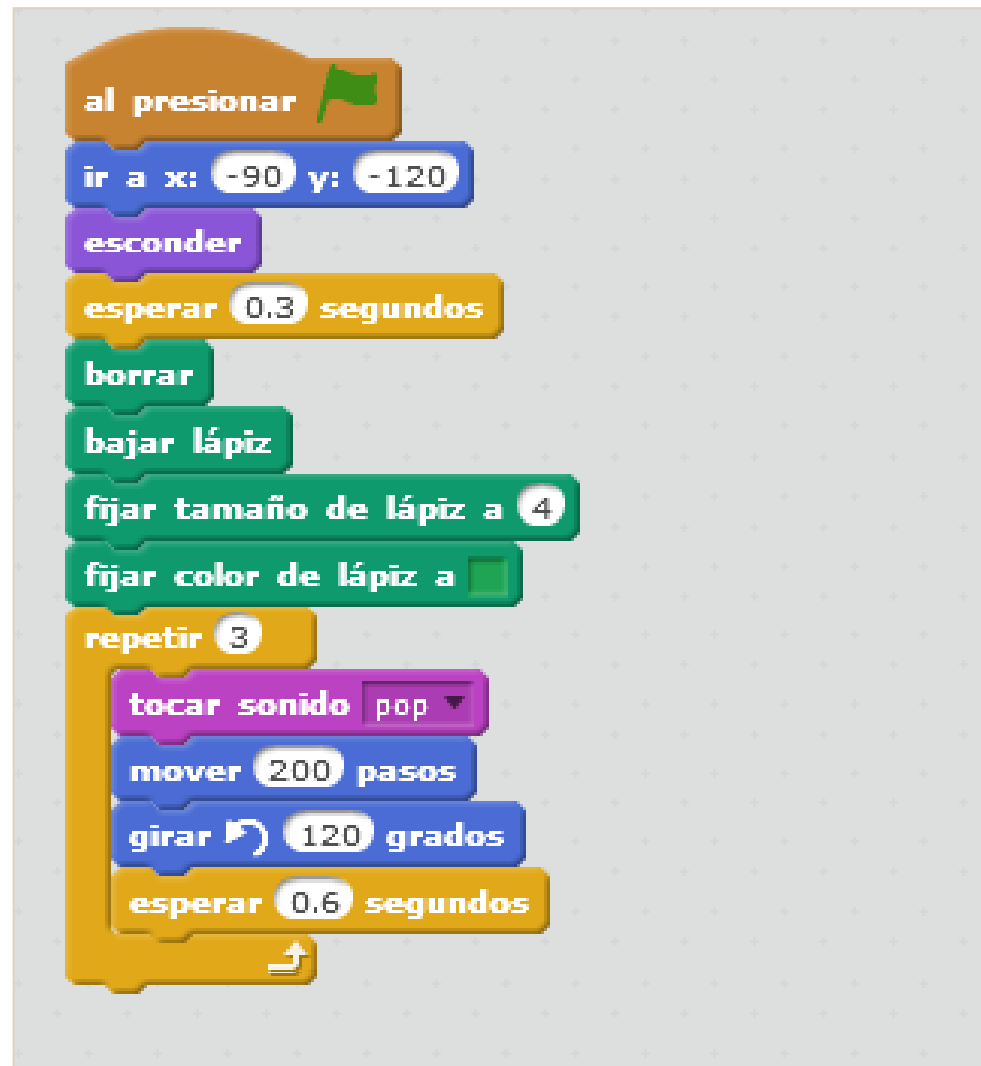


# Segundo programa

- Dibujar un hexágono
- <https://scratch.mit.edu/projects/24533963/>



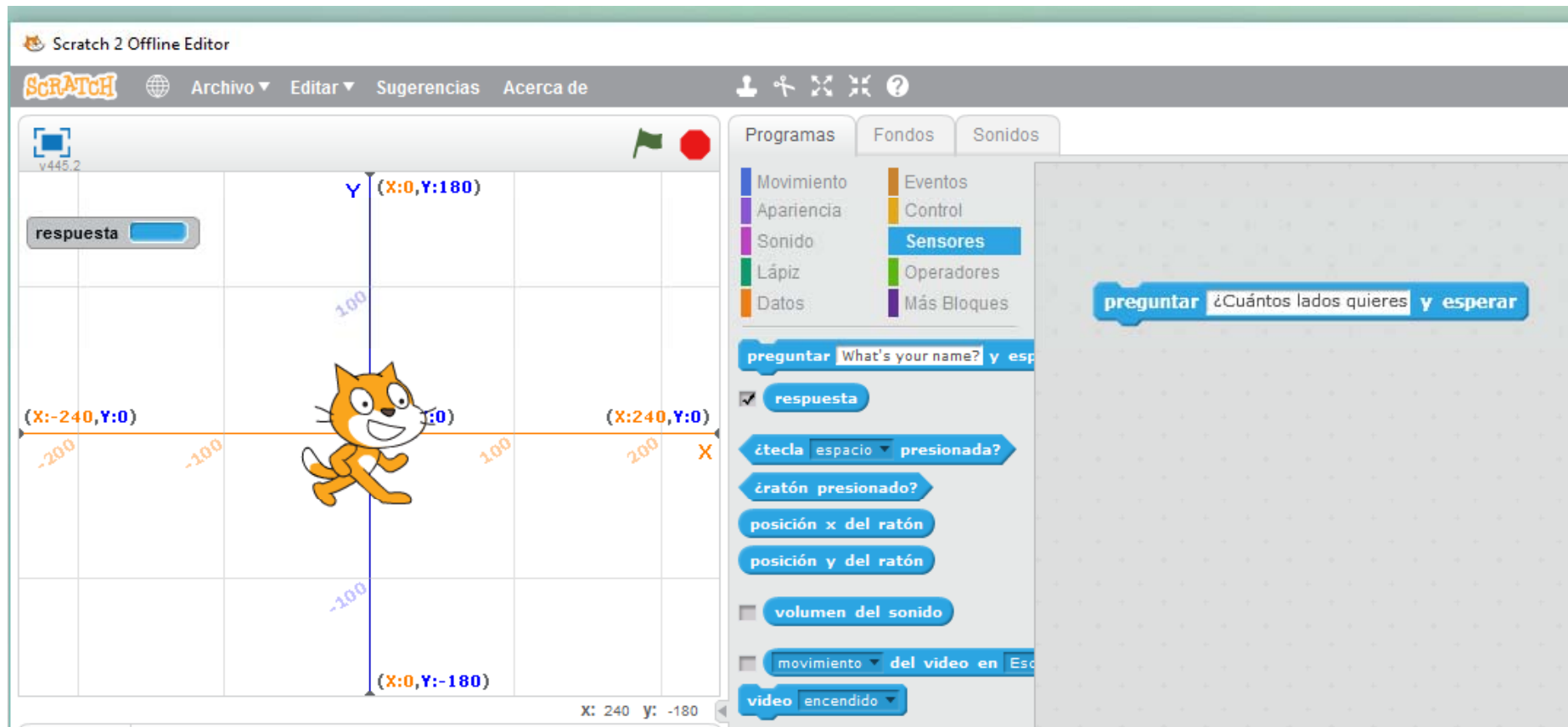
# Un triángulo



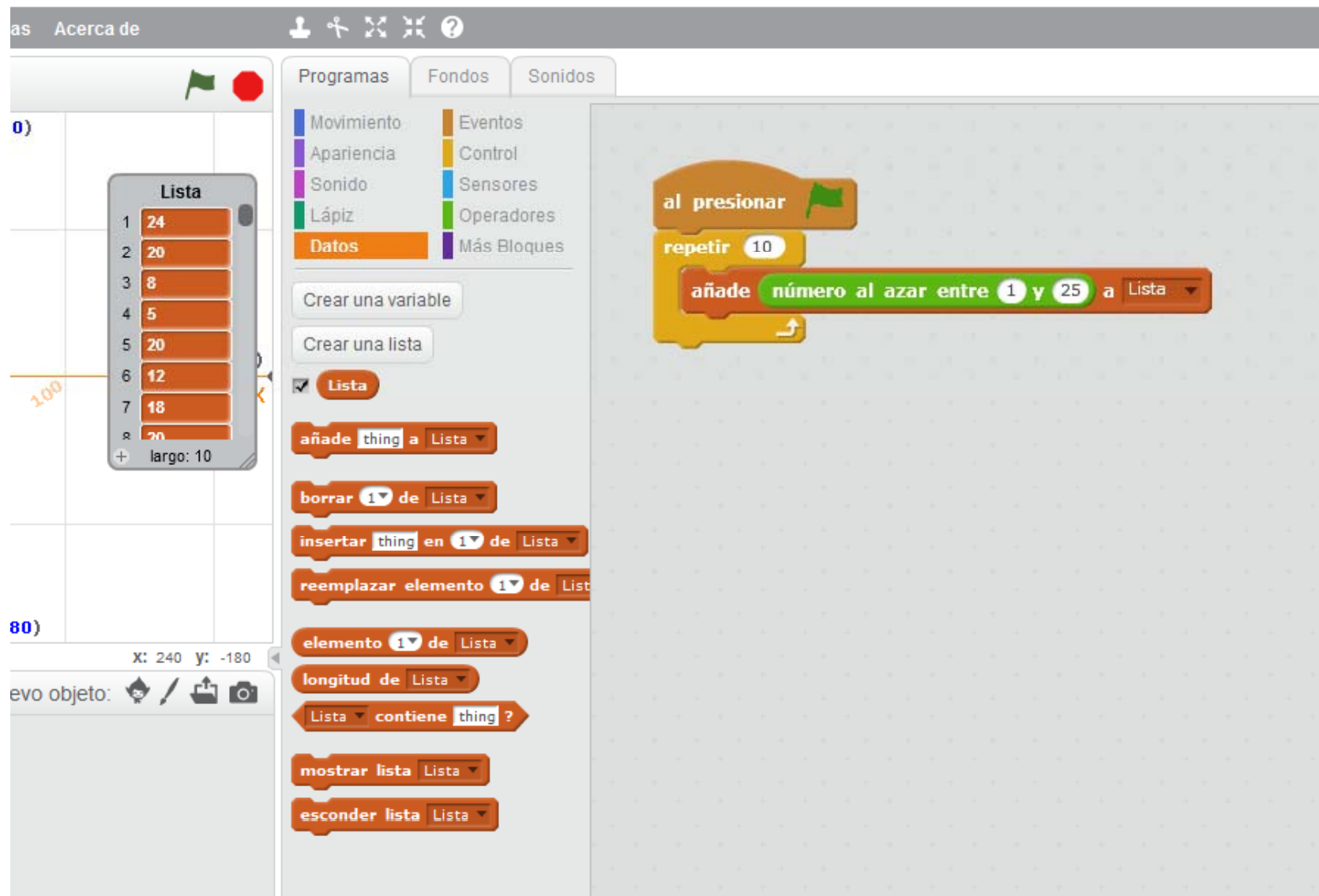
# Datos y Variables

The screenshot shows the Scratch 2 Offline Editor interface. The stage displays a coordinate grid with the Scratch cat character at the center (0,0). The grid has X and Y axes ranging from -200 to 200. A variable 'numLados' is shown in the top left corner with a value of 0. The right sidebar shows the 'Programas' tab with the 'Datos' category selected. The 'Crear una variable' button is visible, and the 'numLados' variable is listed with a checkmark. Below it, a script is shown with the following blocks: 'fijar numLados a 0', 'cambiar numLados por 1', 'mostrar variable numLados', and 'esconder variable numLados'. Two callout boxes with arrows point to the 'fijar' and 'cambiar' blocks, labeled 'Asignar' and 'Incrementar' respectively.

# Datos y Variables



# Listas



The image shows the Scratch interface with a list object and its associated blocks.

**Lista (List):**

- 1: 24
- 2: 20
- 3: 8
- 4: 5
- 5: 20
- 6: 12
- 7: 18
- 8: 20
- + largo: 10

**Programas (Programs):**

- Movimiento (Movement)
- Apariencia (Appearance)
- Sonido (Sound)
- Lápiz (Pen)
- Datos (Data)
- Eventos (Events)
- Control (Control)
- Sensores (Sensors)
- Operadores (Operators)
- Más Bloques (More Blocks)

**Crear una variable (Create a variable):**

- Crear una lista (Create a list)

**Lista (List):**

- ☒ Lista
- añade thing a Lista
- borrar 1 de Lista
- insertar thing en 1 de Lista
- reemplazar elemento 1 de Lista
- elemento 1 de Lista
- longitud de Lista
- Lista contiene thing ?
- mostrar lista Lista
- esconder lista Lista

**Script Area:**

- al presionar (when clicked)
- repetir 10 (repeat 10 times)
- añade número al azar entre 1 y 25 a Lista (add random number between 1 and 25 to Lista)

# Propuestas para el aula

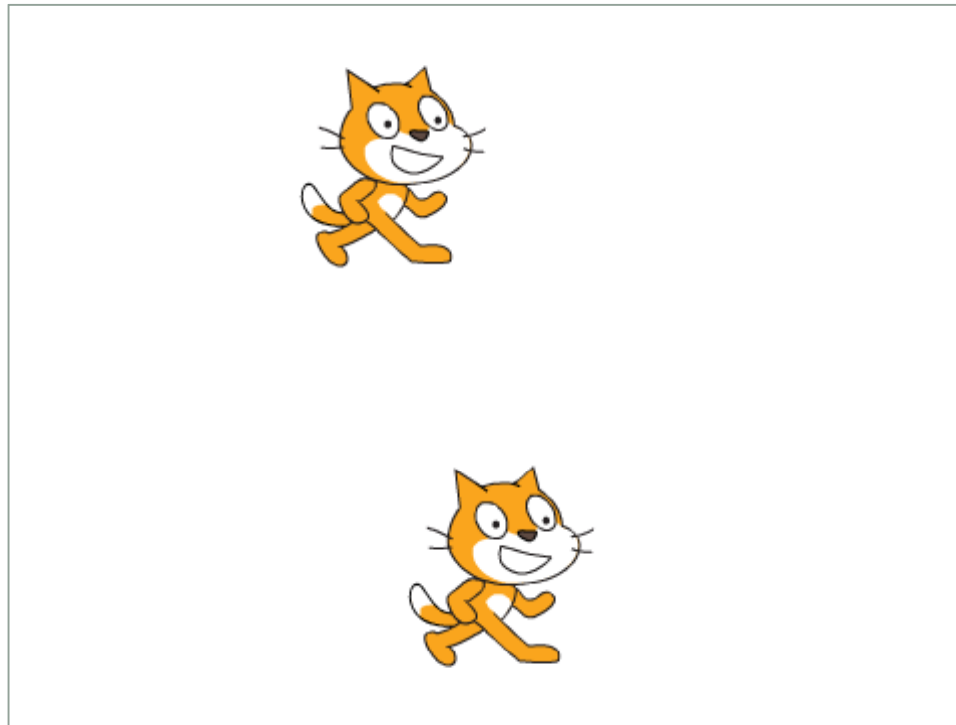
- Simular un tiro parabólico con un balón





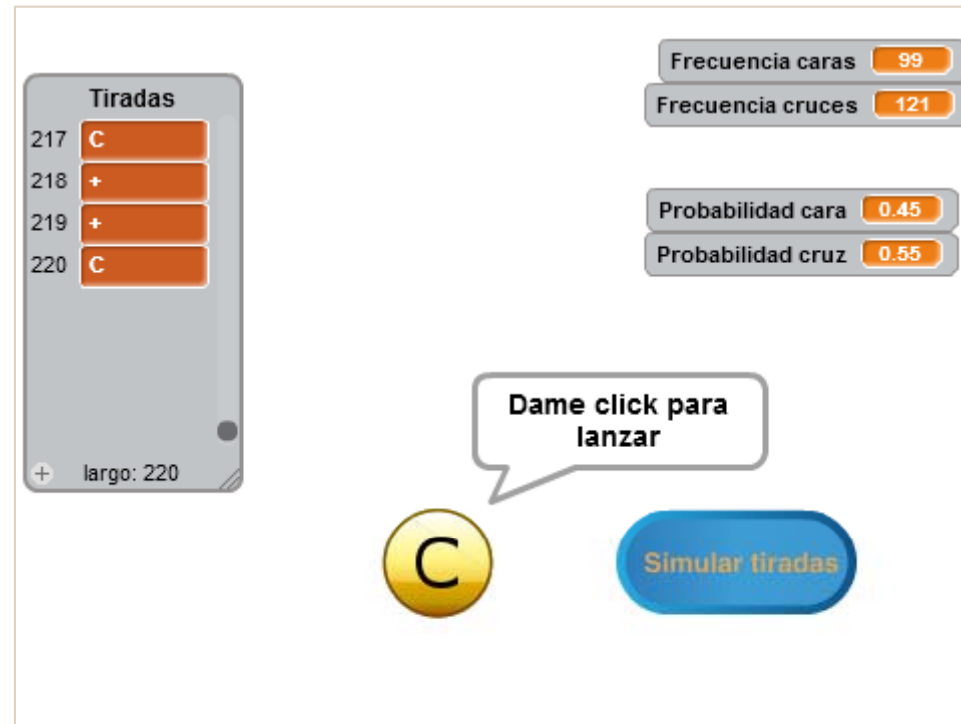
# Propuestas para el aula

- Dos objetos que se desplazan, ¿cuándo tengo que lanzar a uno o a otro para que se crucen, se alcancen, "gane" uno,...?



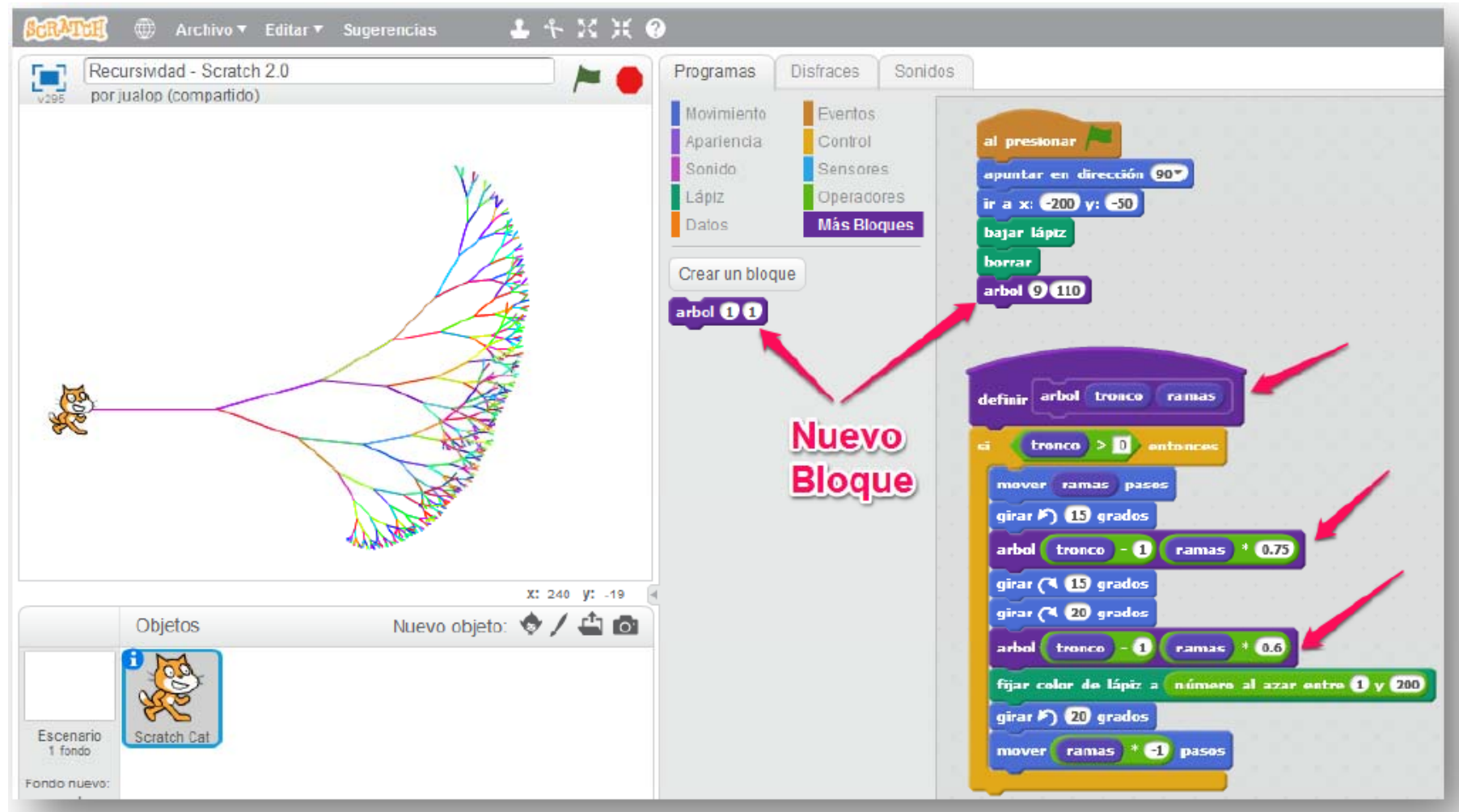
# Probabilidades

- Lanzamiento de una moneda (cara o cruz). Se muestra el resultado y se actualizan las frecuencias y probabilidades.
- Además, existe un botón para simular varios intentos al mismo tiempo y ver los resultados. Puedes comprobar que las probabilidades de las caras de la moneda tienden a  $1/2$ .



(<https://scratch.mit.edu/projects/105660743/>)

# Recursividad



<https://scratch.mit.edu/projects/36425832/>

# Sucesión de Fibonacci

- Analiza el programa **fibonacci.sb** que genera la llamada *sucesión de Fibonacci*.
- Explica “con palabras” qué datos se le dan al programa, cómo se genera la lista y hasta cuándo se generan términos.
- Construye otra lista que te dé el resultado de dividir cada término de *la sucesión* por el anterior. ¿Qué observas?

# Sucesión de Collatz

Construye un programa que, a partir de un número que se introduce, genere la sucesión de Collatz:

- Si un término es par, el siguiente resulta de dividirlo por 2
- Si un término es impar el siguiente se obtiene multiplicándolo por 3 y sumando 1.

Existe una conjetura que indica que la sucesión de Collatz siempre llega a un término igual a 1. El programa debe acabar cuando se cumpla esa condición.

(collatz.sb)

# ¡Muchas gracias!

chernan@infor.uva.es

bpalop@gmail.com

scratch@infor.uva.es

<http://scratch.infor.uva.es>