Sesión 3

Programación por bloques

Pensamiento computacional y robótica educativa · sesión 3 de 6

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

Objetivo

- Pasar los cuatro pilares a un entorno de bloques (Scratch o los entornos propios EduTicTac) mediante retos progresivos.
- Terminar con un proyecto propio que combine varios conceptos a la vez.

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

De consumidores a creadores de tecnología

Programar es escribir un código que entienda la máquina, con sus reglas y pasos: eso obliga a **entender de verdad** lo que le estamos pidiendo.

El objetivo no es que el alumnado consuma tecnología, sino que **cree con ella**.

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

Scratch y la programación por bloques

Programar puede parecer complejo para el aula, pero las aplicaciones visuales lo han resuelto con la **programación por bloques**.

Scratch es una de las más extendidas:

- Se programa **por bloques**, arrastrando piezas.
- Está **orientada a objetos**: a cada personaje se le asignan sus subprogramas.
- Entorno muy **amigable**, con gráficos intuitivos y manejo sencillo → muy motivadora en educación.

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

Los entornos propios de EduTicTac

Dos entornos equivalentes de **código libre**, con opción de alojamiento propio y uso sin cuenta. En una instancia local, los datos permanecen en el centro:

- **EduTicTac Blocs** (`blocs.edutictac.es`) — como Scratch, para Primaria y Secundaria.
- **EduTicTac Blocs Junior** (`blocs-junior.edutictac.es`) — como ScratchJr, para Infantil y 1r ciclo.

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

En valenciano y castellano, en el navegador. Cada uno trae **plantillas docentes** para leer, modificar y ampliar (interpretar → modificar → crear): doce en Blocs y seis en Blocs Junior.

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

De código máquina a bloques

La programación ha evolucionado hasta hacerse accesible:

código máquina (binario) → **ensamblador** → **lenguajes de alto nivel** → **lenguajes visuales por bloques**.

Los bloques **facilitan introducir la programación a edades muy tempranas** y asimilar los conceptos y estructuras computacionales para resolver cualquier reto.

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

También en el móvil: Mobile Learning

Muchas apps de programación funcionan en móvil o tablet, y permiten que el alumnado trabaje con sus propios dispositivos en el aula.

Con la creatividad de los gráficos, los procesos intuitivos y los tutoriales guiados, **introducirse en el pensamiento computacional nunca ha sido tan sencillo**.

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

Bloques = conceptos ya trabajados

Cada categoría de bloques corresponde a un concepto de la sesión 2:

Bloques	Concepto
Movimiento / apariencia en cadena	Secuencia (robots humanos)
Eventos ("al pulsar bandera verde")	Disparo de la secuencia
Control → condicionales	Cartas lógicas
Control → bucles	Repetir sin reescribir
Datos → variables	Sobres variables
Varios personajes a la vez	Paralelismo

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

Resolver antes de explicar

Idea clave

No se explica cada categoría antes de tocarla. Se plantea un reto concreto, se busca el bloque que hace falta, y **solo después** se nombra el concepto.

Primero resolver, después nombrar: una sucesión de retos cada vez un poco difíciles.

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

El proyecto propio: requisitos mínimos

Sin restricciones, el alumnado repite lo que ya sabe y evita lo que cuesta.

Por eso se fija un mínimo obligatorio:

- Un personaje con más de un disfraz
- Un escenario propio
- Al menos una iteración (bucle)
- Un condicional
- Una variable
- Un sonido propio

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

Revisión entre iguales

Intercambiar proyectos con un compañero y dar **una idea concreta de mejora** — no una valoración general.

Con alumnado, esto se formaliza con una rúbrica breve. Se retoma con más profundidad en la **sesión 6**.

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

Del proyecto individual al proyecto de aula

¿Qué parte se hace en gran grupo (descubrimiento inicial del entorno)? ¿Qué parte se deja para trabajo individual o por parejas (retos progresivos, diseño final)?

¿Cuántas sesiones de clase hacen falta para el mismo recorrido que aquí ocupa una sola sesión de formación?

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

Otros entornos de bloques

El vocabulario —secuencia, eventos, condicional, variable, bucle— es el mismo en todos: lo aprendido hoy es transferible aunque el centro use otra herramienta.

Infantil (3-5 años)

Nada de lectura ni bloques de texto. Paleta muy reducida —mover, girar, sonido, empezar al tocar— solo iconos, para tablet. Retos de 2-4 pasos. Es el caso de **EduTicTac Blocs Junior**, cuya pestaña Retos trae seis plantillas docentes listas.

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

Primaria (6-12 años): ciclo a ciclo

Decretos 106/2022 y 96/2026 · Matemáticas CE4 + Bloque 6 · Medio CE2

- **1r ciclo — interpretar**: ScratchJr o Blocs Junior en tablet, bloques de iconos. "¿Qué hará este programa?" antes de ejecutarlo.
- **2º ciclo — modificar**: Scratch o Blocs a partir de un proyecto que ya funciona (o una plantilla). Bucles, eventos, patrones; un cuadrado = repetir 4 (avanza, gira 90°).
- **3r ciclo — crear**: proyecto propio con requisitos mínimos (bucle, condicional, variable), como el de hoy; una plantilla de Blocs sirve de esqueleto.

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

Con Secundaria que ya domina bloques

Quedarse en los retos básicos no enseña nada nuevo. Dos saltos reales:

- **Listas** — colección de valores, se recorre, se añade o quita: cuestionarios, puntuaciones, sorteos.
- **Bloques personalizados con parámetros** — encapsular una secuencia repetida en una pieza reutilizable ("saludar (nombre)").

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

El puente al código de texto

Con este grupo, mostrar el mismo proyecto en un entorno con vista de código (MakeCode, el mismo de micro:bit en la sesión 4).

No hace falta escribir texto hoy — con reconocer ahí la secuencia, el condicional o el bucle ya programado en bloques basta. Desmonta la idea de que "bloques = fácil, texto = de verdad": son dos representaciones del mismo programa.

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

Dos rutas, un proyecto útil

Esencial: modificar una plantilla (evento, valor, repetición). **Desafío:** predecir antes de ejecutar y escribir la consigna para el alumnado.

Profundización: añadir una lista o un bloque personalizado. **Desafío:** probar dos entradas, registrar un error y explicar cómo se reutiliza la abstracción.

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

Cierre

Recapitulando

De los seis conceptos trabajados, ¿cuál le faltaba a vuestro proyecto la primera vez que lo intentasteis?

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.

Fuentes

- **«De consumidores a creadores de Tecnología»** — INTEF, MOOC *Pensamiento Computacional educativo* (#MoocBot). De aquí salen el enfoque «crear con la tecnología», la presentación de Scratch y bloques, la evolución de los lenguajes y el Mobile Learning.

© Samuel Soriano · Licencia CC BY-NC-SA 4.0.