



## Título: Retratos cubistas

**Nivel educativo:** 4º/5º de Primaria

**Áreas Curriculares:** Educación Artística. Robótica

**Temporalización:** Cualquier trimestre



## Descripción breve de la actividad

En esta actividad crearemos un generador de **caras cubistas** al estilo de **Pablo Picasso** usando la herramienta Scratch. El programa combina diferentes versiones o **disfraces** de un sprite (ojos, nariz, boca...) mediante valores aleatorios, generando rostros únicos cada vez que se ejecuta. Se trata de una actividad interdisciplinar que combina arte, tecnología y pensamiento computacional



## Objetivos

- Conocer el estilo pictórico cubista de Picasso
- Comprender el uso de disfraces de sprites para representar variaciones.
- Introducir el concepto de aleatoriedad en programación.
- Desarrollar creatividad artística combinada con pensamiento lógico.

**Competencias clave a desarrollar:** Competencia matemática y en ciencia, tecnología e ingeniería (STEM), Competencia Digital, Competencia personal, social y de aprender a aprender y Competencia en conciencia y expresión culturales



## ¿Cómo lo hacemos?



**Sesión previa:** es recomendable que hayamos hablado sobre las obras de Pablo Picasso y, en concreto, de su etapa cubista y su representación de la figura humana haciendo especial hincapié en las figuras geométricas y perspectivas usadas para representar las diferentes partes del cuerpo.

**Preparación de la sesión:** ya que los alumnos utilizarán un banco de imágenes específico, se recomienda utilizar algún medio seguro para compartirlo con los alumnos, por ejemplo, Aula Virtual. También puedes tenerlo ya almacenado en local en todos los dispositivos que se vayan a utilizar en la sesión.

- 1. Presentación de la actividad:** contaremos a los alumnos que se realizará un generador de caras cubistas aleatorio, que cada vez que pulsen la bandera verde se creará una cara nueva.
- 2. Preparación del material:** los alumnos accederán a su Aula Virtual para descargar los archivos en el ordenador de trabajo o accederán a los archivos en el explorador de archivos de su dispositivo. Después abrirán el navegador, accederán a la web de Scratch y crearán un proyecto nuevo.
- 3. Importación de los sprites:** los alumnos importarán 7 sprites diferentes que corresponden con las diferentes partes de la cara (cara, ojo izquierdo, ojo derecho, oreja izquierda, oreja derecha, boca y nariz) y añadirán las diferentes versiones que se proponen. Podrás encontrar el paso a paso en el apartado “Recursos”.
- 4. Programación:** los alumnos utilizarán el bloque “cambiar disfraz a” del grupo “apariencia” para cada sprite. Además, añadirán el bloque “un número aleatorio entre 1 y X “ que encontramos en el grupo “operadores” para generar un evento aleatorio siendo X el número total de disfraces que han importado para esa parte del cuerpo. Todo lo unirán al bloque “al hacer clic en bandera verde”. Podrás encontrar el paso a paso en el apartado “Recursos”.
- 5. Comprobación:** una vez hayan generado los códigos de programación para todos los sprites, deberán comprobar el correcto funcionamiento de su programa y resolver los posibles errores que hayan cometido.
- 6. Creación del retrato cubista:** ahora podrán divertirse pulsando la bandera verde y creando los retratos cubistas más alocados.



## Sugerencias

Podrás encontrar el paso a paso de los puntos 3 y 4 en el apartado “Recursos”

Puedes ofrecer a los alumnos la opción de que ellos mismos dibujen las diferentes partes de la cara o alguna de sus versiones, dando más valor a la parte artística.

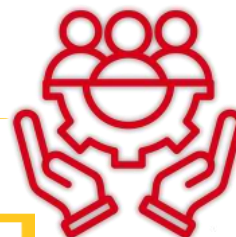
Si dispones de más tiempo o sesiones, pueden:

- Añadir un bloque extra que modifique el color de los sprites (apariciencia > dar al efecto - color - el valor /número aleatorio entre \_ y \_ /), para ello habrá que cerrar aquellas figuras con líneas abiertas y rellenarlas de un color con el cubo.
- Crear un botón de “Generar nueva cara” para sustituir el “al pulsar bandera verde”.
- Incluir movimientos a los sprites (varias opciones en grupo “movimiento”).



## Recursos

- **Personales:** profesor
- **Materiales:** ordenadores, archivos de sprites.



**Espacios:** sala de informática, aula, aula del futuro...

**Tipo de actividad:** individual (según disponibilidad de dispositivos)



[Archivo ZIP con todos los sprites y disfraces.](#)

[Paso a paso de creación y programación.](#)

**Proyecto completo de Scratch:**

<https://scratch.mit.edu/projects/1305680191>





## ¿Qué hemos aprendido?

<b>Criterios de Evaluación</b>	<b>4 Excelente</b>	<b>3 Satisfactorio</b>	<b>2 Mejorable</b>	<b>1 Insuficiente</b>
<b>Diseña un programa funcional con Scratch.</b>	El programa funciona de forma completa y fluida, cumple el objetivo (genera caras correctamente) y presenta una estructura clara y bien organizada.	El programa funciona en general y cumple el objetivo principal, aunque puede presentar pequeños fallos o falta de organización.	El programa funciona parcialmente; algunas partes no responden correctamente o el resultado no siempre es el esperado.	El programa no funciona o no cumple el objetivo propuesto.
<b>Utiliza correctamente eventos y cambios de disfraz.</b>	Utiliza adecuadamente eventos (bandera, bloques, etc.) y cambios de disfraz en todos los elementos, logrando una ejecución coherente y sincronizada.	Usa eventos y cambios de disfraz de forma correcta en la mayoría de los casos, aunque con alguna descoordinación menor.	Utiliza eventos o cambios de disfraz de forma limitada o con errores frecuentes.	No utiliza correctamente los eventos ni los cambios de disfraz, o están ausentes.
<b>Aplica la aleatoriedad para generar resultados variados.</b>	Aplica la aleatoriedad correctamente en todas las partes del proyecto, generando múltiples combinaciones claramente diferentes.	Aplica la aleatoriedad en varias partes, generando cierta variedad aunque limitada.	La aleatoriedad es escasa o poco efectiva, generando resultados repetitivos.	No utiliza la aleatoriedad o el resultado es siempre el mismo.
<b>Mejora el programa tras detectar errores.</b>	Identifica errores de forma autónoma y realiza mejoras significativas (funcionamiento, diseño o creatividad).	Detecta algunos errores y realiza correcciones básicas con ayuda o de forma parcial.	Detecta errores pero tiene dificultades para corregirlos o necesita mucha ayuda.	No identifica errores ni realiza mejoras en el programa.



## Pensamiento computacional

**Lógica (predicción y análisis):** utilizar el razonamiento para hacer predicciones sobre si el programa fallará, resolver posibles problemas y tomar decisiones basadas en lo observado.

**Algoritmos (pasos y reglas):** seguir una serie de pasos o instrucciones bien definidas para resolver un problema o completar una tarea.

**Patrones (detectar y usar similitudes):** identificar similitudes o patrones en la programación de los diferentes sprites, lo que facilita la realización de la misma y su adaptación a las diferentes características.



## Más información

También puedes usar estos Códigos QR para descargar los archivos:



[Sprites y disfraces](#)



[Paso a paso](#)

## Autoría

Esta actividad ha sido realizada por **Irene Úbeda Pedruelo**, en el marco del Programa Código Escuela 4.0 Madrid.