



## **Título:** El código de las obras maestras.

**Nivel educativo:** 1º ciclo de Educación Primaria.

**Áreas Curriculares:** Tecnología y Robótica, Educación Artística.

**Temporalización:** Tercer trimestre (Día del Arte: 15 de abril), 1 o 2 sesiones.



## **Descripción breve de la actividad**

En esta actividad desenchufada, reproduciremos obras de arte famosas utilizando un concepto básico de compresión de datos y programación llamado Run-Length Encoding (RLE): un algoritmo de compresión de datos sin pérdida que simplifica la codificación almacenando secuencias consecutivas de datos idénticos como un único valor seguido de su recuento. Por ejemplo, la cadena "AAABBBCCCC" se comprime comúnmente como "A3B3C4", reduciendo el espacio necesario cuando los datos contienen muchas repeticiones, como en gráficos simples con áreas de color plano.



## **Objetivos**

Comprender los conceptos de algoritmo, codificación, decodificación y píxeles, trabajando la comunicación precisa y la resolución de problemas (depuración o debugging).

**Competencias clave a desarrollar:** competencia en comunicación lingüística; competencia matemática y competencia en ciencia, tecnología e ingeniería; competencia digital; competencia personal, social y de aprender a aprender; competencia en conciencia y expresión culturales.



## ¿Cómo lo hacemos?



### 1. Introducción:

Explicación a los alumnos. Las pantallas de los ordenadores y móviles no entienden de "pinceladas", sino que dividen las imágenes en pequeños cuadrados llamados **píxeles**.

Cada píxel tiene un color específico y, para que el ordenador sepa qué dibujar, necesita un código.

A mayor número de píxeles, mayor calidad de imagen.

### 2. El juego de "El programador y la pantalla"

En **parejas**. Uno será el "programador" y el otro será la "pantalla".

El programador recibe en secreto el código de un cuadro famoso.

La pantalla tiene la cuadrícula en blanco y lápices de colores.

El programador debe dictar el código fila por fila a la pantalla.

La pantalla va coloreando los píxeles según las instrucciones.

### 3. Cambio de roles y "El cuadro perdido"

Se intercambian los roles. Ahora, el nuevo programador recibe un cuadro en pixel art pero sin el código.

El desafío aquí es aplicar la abstracción: el programador tiene que mirar el dibujo, generar el código en su cabeza (o escribirlo en un papel sucio) y dictárselo a su compañero.

### 4. La galería de arte y depuración.

Al terminar, todas las pantallas muestran sus obras.

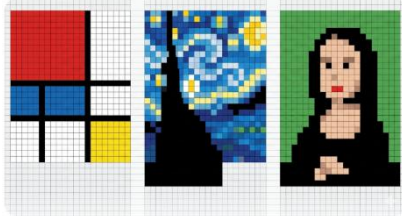
¿Se parecen al cuadro original de Van Gogh, Mondrian o Da Vinci?

Si hay un error (un píxel fuera de lugar), se aprovecha para explicar el concepto de **depuración (debugging)**: repasar el código para encontrar dónde falló la instrucción o la interpretación.



## Sugerencias

Con alumnos más mayores, se pueden usar cuadrículas con más píxeles para que comprueben cómo mejora la definición de la imagen.



Se puede animar a los alumnos a recrear otros cuadros famosos y crear el código para sus compañeros.



## Recursos

- **Personales:** Docente, alumno, compañeros.
- **Materiales:** Hojas con cuadrículas vacías, lápices o rotuladores de colores, tarjetas de referencia con el código correspondiente.

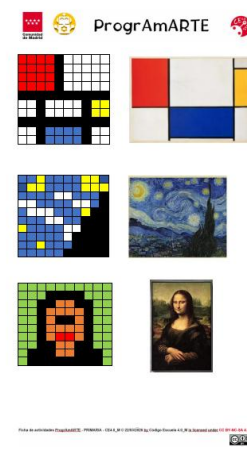
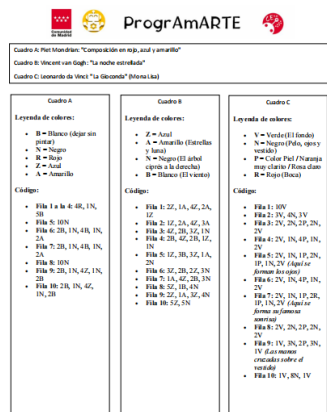
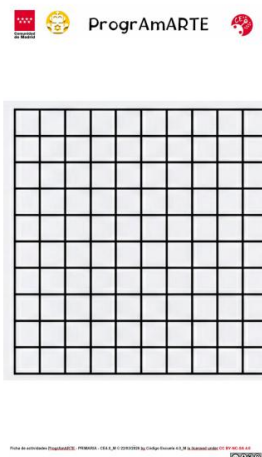


**Espacios:** aula clase

**Tipo de actividad:** pareja



Enlace a cuadrícula, código y cuadros: [pincha aquí.](#)





## ¿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>Decodificación (Rol: Pantalla)</b>  (Capacidad para interpretar y ejecutar el código RLE)	Interpreta y ejecuta el algoritmo a la perfección.	Sigue bien el código pero comete 1 o 2 errores menores.	Sigue el código con dificultad. Comete varios errores que alteran el cuadro original.	No logra seguir el código. Pinta casillas al azar, se salta filas o la imagen final es irreconocible.
<b>Codificación (Rol: Programador)</b>  (Capacidad para abstraer una imagen y convertirla en código RLE)	Crea un código exacto, ordenado fila por fila, demostrando una comprensión total de la compresión de datos.	Genera un código casi correcto, con algún pequeño error de cálculo en una o dos filas que se puede subsanar rápido.	Genera el código pero con varios fallos en el conteo de bloques o en la asignación de colores.	No comprende cómo transformar los píxeles de la imagen en letras y números. El código es ilegible o muy erróneo.
<b>Depuración (Debugging)</b>  (Identificación y resolución de problemas)	Demuestra gran autonomía. Al ver un píxel fuera de lugar, revisa la línea de código y corrige el fallo.	Es capaz de revisar el código junto a su compañero para encontrar el error, aunque tardan un poco en solucionarlo.	Detecta que hay un error en el dibujo final, pero necesita mucha ayuda del docente para encontrar en qué línea de código ocurrió.	Se frustra ante los errores, abandona la tarea o no es capaz de volver atrás para buscar dónde falló la instrucción.
<b>Comunicación y Trabajo en Equipo</b>  (Precisión en el lenguaje y respeto por las reglas)	Colaboración impecable. El programador dicta con un ritmo adecuado y lenguaje preciso.	Trabajan bien juntos. El dictado es bastante claro, aunque en momentos puntuales les falta un poco de paciencia o precisión.	La comunicación es confusa. Tienen que mirar el papel del compañero alguna vez para avanzar.	No se comunican de forma efectiva. Rompen la regla de "no mirar" constantemente o no respetan los roles asignados.



## Pensamiento computacional

**Lógica (predicción y análisis):** utilizar el razonamiento para hacer predicciones, resolver problemas y tomar decisiones basadas en la información disponible.

**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 problemas o datos, lo que facilita encontrar soluciones más rápidas y eficientes.

**Abstracción (eliminar detalles innecesarios):** simplificar un problema eliminando detalles que no son importantes, para enfocarse en lo que es relevante y esencial.



## Más información

Códigos QR vinculados con los recursos de la actividad:



## Autoría

Esta actividad ha sido realizada por **Andrés Rubio León**, en el marco del **Programa Código Escuela 4.0 \_Madrid**.