

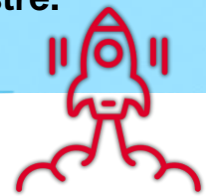


Título: ARTE HUSKY LENS

Nivel educativo: 3º y 4º Secundaria.

Materia/s: Tecnología, Digitalización y Educación Plástica.

Temporalización: 3 sesiones durante el 3er trimestre.



Descripción breve de la actividad

Los estudiantes diseñarán una exposición interactiva para el **Día del Arte**. Utilizando la cámara **HuskyLens** y la placa **micro:bit**, **los alumnos** programarán un sistema de visión artificial inteligente, capaz de identificar al menos 20 obras pictóricas. Al reconocer un cuadro, la placa deberá mostrar en pantalla: **nombre de la obra, autor, fecha y/o movimiento artístico**.



Objetivos

- Configurar el algoritmo de **clasificación de objetos** de la HuskyLens para reconocer patrones pictóricos con la IA.
- Programar estructuras condicionales que vinculen un ID de imagen con cuatro variables de información artística.
- Organizar una exposición con las obras seleccionadas en la muestra.



Competencias clave a desarrollar:

STEM2: Utilizar el pensamiento computacional para resolver problemas mediante el desarrollo de algoritmos.

CD3: Programar dispositivos hardware (micro:bit) y configurar herramientas de Inteligencia Artificial (HuskyLens).

CCEC3: Expresar y enriquecer la identidad cultural a través del estudio de obras maestras.



Pasos a seguir

- 1. Fase de Ingeniería Inversa:** Observar el funcionamiento de los 2 cuadros iniciales para deducir la relación entre el ID de la cámara y la variable de salida.
- 2. Investigación de Catalogación:** Los alumnos investigan la ficha técnica (Autor, Fecha, Estilo) de los cuadros asignados. Se ofrece una galería de 30 obras pictóricas como ejemplo.
- 3. Configuración de Hardware:** Conexión de HuskyLens a micro:bit y placa de expansión mediante protocolo I2C.
- 4. Entrenamiento de la IA:** Grabación de los 20 modelos en la cámara, asignando IDs del 1 al 20. Se usa el algoritmo "Object Classification" de Husky Lens.
- 5. Programación Lógica:** Crear bloques "Si ID = X" que asignen valores a las cadenas de texto del cuadro. Uso de las funciones para simplificar la programación.
- 6. Inauguración de la Exposición:** Los alumnos presentan su sección de la galería al resto del centro.



Sugerencias

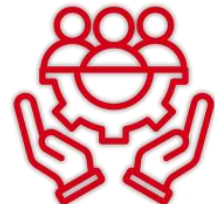
En la preparación de la actividad, podemos añadir también una breve investigación sobre el movimiento artístico al que pertenece cada obra, incorporando el nombre del movimiento artístico a la información de la cámara HuskyLens cuando modifiquen la programación.

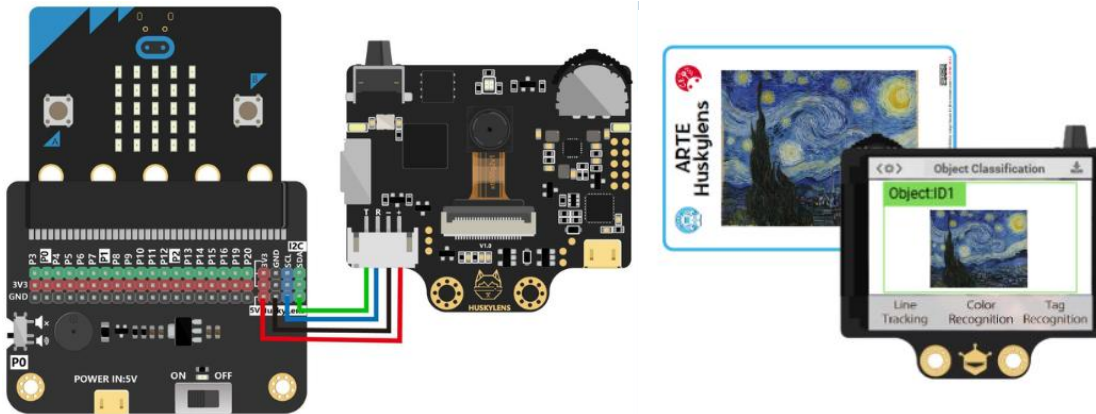
De cara a la preparación de la exposición inteligente, como ampliación, se puede proponer a los alumnos que diseñen un espacio web con la colección de obras y su información más relevante, a modo de galería virtual. Enlazando la web a Códigos QR que se visualicen en cada cuadro, pegados junto a la ficha de la actividad.



Recursos

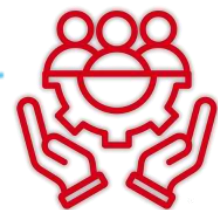
- **Personales:** Equipos cooperativos de 3 alumnos con roles de: Programador, Historiador de Arte y Técnico de IA.
- **Materiales:** Kit micro:bit Secundaria, HuskyLens, reproducciones A4 de cuadros, ordenadores con acceso MakeCode.
- **Espacios:** Taller de tecnología y zona de exposición comunitaria.





Materiales para descargar:

- [Fichas de obras pictóricas.](#)
- [Programa de Make Code para Micro:bit](#)





Evaluación

Criterios de Evaluación	4 Excelente	3 Satisfactorio	2 Mejorable	1 Insuficiente
Identificación y Catalogación (CCEC)	La micro:bit muestra con éxito los 4 datos (Nombre, Autor, Año, Estilo) de cada cuadro asignado sin errores históricos.+1	Muestra la información técnica, pero presenta algún error menor en las fechas o la denominación del estilo artístico.+1	La información es incompleta (faltan campos) o la ortografía dificulta la lectura en el panel LED.+1	No se identifica la obra ni se aporta información sobre la misma.+1
Programación y Algorítmica (STEM)	Crea un código eficiente usando funciones o variables globales para gestionar los 20 cuadros, optimizando la memoria de la placa.+1	El código es funcional y utiliza condicionales correctamente para cada ID, aunque es redundante.+2	El programa presenta fallos lógicos que hacen que la información se mezcle o se salte pasos.+1	El código no logra vincular los IDs de la cámara con las cadenas de texto.+1
Entrenamiento de Visión Artificial (CD)	Calibra la HuskyLens para que el reconocimiento sea instantáneo y estable bajo diferentes condiciones de luz.+1	Logra que la cámara reconozca las obras, aunque requiere varios intentos o una distancia muy específica.+1	El sistema confunde cuadros similares (mismo autor o estilo) debido a un entrenamiento deficiente.+1	La cámara no reconoce ninguno de los cuadros nuevos propuestos en el reto.+1
Pensamiento Crítico y Depuración	Identifica errores de lectura y propone soluciones técnicas (como ajustar el umbral de confianza o la iluminación).	Es capaz de corregir errores en el código con ayuda mínima del docente.+1	Detecta que algo no funciona pero no sabe identificar si el fallo es del hardware (cámara) o del software.+1	Abandona la tarea ante los errores técnicos iniciales sin intentar depurar el sistema.+1



Pensamiento computacional

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

Diseño de la secuencia: Captura -> Clasificación -> Salida de datos.

Descomposición (dividir en partes): dividir un problema grande en partes más pequeñas y manejables, que son más fáciles de entender y resolver.

Separar la ficha técnica en variables: Nombre, Autor, Año, Estilo.

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

Identificar los rasgos clave (pincelada, color) que definen el movimiento artístico para la IA.

Autoría

Esta actividad ha sido realizada por **Laura Azorín Martínez, Amancio Moreno Rodríguez y Manuel Jiménez Serrano**, en el marco del **Programa Código Escuela 4.0_Madrid**.