

# 6. Experimentar con herramientas de IA. Propuestas didácticas. Segundo y tercer ciclo

- [Propuestas de aula enchufadas: aprender haciendo paso a paso.](#)
- [La placa Echidna Shield Black y la plataforma EchidnaLM](#)
- [Teachable Machine en el aula: entrenar con ejemplos \(imagen, sonido y postura\)](#)
- [Teachable Machine vs EchidnaML: ¿cuál elegir según tu aula?](#)
- [Actividad 1. Vivo o no vivo: entrenamos un clasificador con objetos del aula](#)
- [Actividad 2. Radar Sonoro: entrenamos un clasificador de sonidos del entorno](#)
- [Actividad 3. Entrenar con datos sesgados: cuando la IA aprende "mal"](#)
- [Actividad 4. IA + robótica con Echidna: la barrera inteligente \(reconocimiento y acción\)](#)
- [Actividad 5. La lámpara inteligente: entrenamos un modelo de texto \(enciende/apaga\) y lo mejoramos con sinónimos](#)

# Propuestas de aula enchufadas: aprender haciendo paso a paso.

En este bloque aprenderemos cómo entrenar modelos con imágenes, sonidos o gestos, y cómo conectar esos modelos a proyectos tangibles con materiales como los que aporta el programa **código Escuela 4.0 (placas programables de Echidna o Micro:bit)**, de modo que la IA cobre vida en la robótica educativa, con una idea clave como punto de partida: **la seguridad y la protección de datos es una parte importante del diseño de la actividad**. Este enfoque práctico permite no solo experimentar con la tecnología, sino también reflexionar sobre la responsabilidad que implica entrenar una IA y sobre la importancia de detectar y cuestionar posibles sesgos.

Este enfoque práctico y experimental nos ayudará a desmitificar la IA, promoviendo una actitud científica, ética y reflexiva. Porque cuando los estudiantes entienden cómo aprende una máquina, también aprenden a pensar mejor sobre su propio aprendizaje y sobre el impacto social de la tecnología que utilizan.

De esta manera las propuestas que encontrarás a continuación combinan dos elementos inseparables: una dinámica clara (que harán los alumnos y para qué) y una configuración responsable.

El objetivo es que puedas trabajar la IA de manera sencilla y práctica teniendo en cuenta que datos conviene usar, cuales evitar, qué ajustes aplicar y que alternativas existen cuando el centro necesita trabajar sin conexión o con recurso limitados. Vas a ver actividades guiadas "paso a paso" con herramientas accesibles y conocidas del entorno educativo como **Teachable machine**, **Scratch offline**, así como opciones complementarias para enriquecer los proyectos como **Machine Learning for Kids**, **Google Labs**, **Quickdraw (muy útil en infantil)** o **Code IA**.



A lo largo de los apartados posteriores trabajaremos con un enfoque muy docente:

- **Qué se pretende aprender** (adaptado a la etapa y áreas a trabajar).
- **Cómo se entrena y se prueba un modelo** con ejemplos cotidianos (sonidos, objetos, palabras clave, imágenes, dibujos).
- **Cómo evaluar el resultado** con mirada crítica (errores, sesgos, mejoras posibles).

En definitiva, este bloque está diseñado para que puedas decir: *“lo entiendo, lo puedo aplicar y sé hacerlo de forma segura” a través de distintas actividades planteadas.*

Aquí empieza el trabajo más visible: **convertir la IA en una experiencia educativa significativa, responsable y adaptada a tu aula**



Ilustración 1 Imagen generada con ChatGPT (OpenAI). 2026

# La placa Echidna Shield Black y la plataforma EchidnaLM

## 1. Introducción: hardware y software para la robótica educativa con IA

La integración de la programación, la robótica educativa y la inteligencia artificial en el aula requiere herramientas que sean accesibles para el profesorado y el alumnado, pero que al mismo tiempo permitan desarrollar proyectos con un cierto nivel de complejidad. Dentro de este contexto surge el ecosistema Echidna, un proyecto educativo de hardware y software abierto diseñado específicamente para facilitar la enseñanza de la programación y el pensamiento computacional en Educación Primaria y Secundaria.

Este ecosistema está formado principalmente por dos elementos:

- Hardware educativo, representado por placas como Echidna Black o la Echidna Shield.
- Software educativo, como EchidnaML y su módulo de aprendizaje automático LearningML.

La combinación de ambos elementos permite trabajar en el aula desde actividades muy básicas de programación por bloques hasta proyectos avanzados que integran sensores, actuadores y modelos de inteligencia artificial.

En el contexto del programa Escuela 4.0, estas herramientas adquieren especial relevancia ya que permiten desarrollar competencias relacionadas con la programación, la robótica y la inteligencia artificial de forma práctica y contextualizada.

## 2. La placa Echidna Shield Black

### 2.1 Qué es Echidna Black

La Echidna Black es una placa electrónica educativa basada en la arquitectura de Arduino. A diferencia de otras placas que requieren múltiples componentes externos, esta integra en un único dispositivo varios sensores y actuadores que permiten comenzar a programar y experimentar sin necesidad de realizar cableados complejos.

Esta característica la convierte en una herramienta especialmente adecuada para el entorno educativo, ya que reduce las dificultades técnicas iniciales y permite centrarse en el aprendizaje de la programación y el diseño de proyectos.

Una de las principales diferencias respecto a la Echidna Shield es que la versión Black incorpora su propio microcontrolador (basado en Arduino Nano), por lo que funciona de forma autónoma y no necesita una placa adicional.

---

## 2.2 Sensores y actuadores integrados

Uno de los elementos más interesantes de esta placa es la integración de diferentes sensores y actuadores directamente en el hardware. Esto permite desarrollar proyectos interactivos desde el primer momento.

Entre los sensores integrados se encuentran:

- Pulsadores digitales, que permiten introducir eventos o interacciones del usuario.
- Joystick analógico, con control en los ejes X e Y.
- Sensor de luz (LDR) para medir la intensidad luminosa del entorno.
- Sensor de temperatura que permite registrar cambios térmicos.
- Micrófono, que capta variaciones de sonido.
- Acelerómetro, capaz de detectar movimiento o inclinación.

En cuanto a los actuadores, la placa incorpora:

- LEDs de estado.
- LED RGB, que permite representar diferentes colores.
- Salida de audio mediante zumbador o jack.

Además, dispone de ocho entradas tipo Makey Makey, lo que permite utilizar objetos cotidianos como interfaces de entrada (por ejemplo plastilina, frutas o materiales conductores).

Esta integración de componentes facilita enormemente el trabajo en el aula, ya que evita errores de conexión y reduce el tiempo necesario para preparar las prácticas.

---

## 2.3 Conectividad y ampliación

La placa también permite ampliar sus posibilidades mediante diferentes conexiones externas.

Entre sus características técnicas destacan:

- Compatibilidad con Arduino Nano y Arduino UNO.
- Conectividad Bluetooth para comunicación inalámbrica.
- Pines de entrada y salida para conectar sensores y actuadores externos.
- Configuración de alimentación para conectar pequeños motores o servos.

Estas posibilidades permiten evolucionar desde actividades sencillas hasta proyectos más complejos de robótica o automatización.

## 3. Programación de la placa

La programación de Echidna Black puede realizarse utilizando diferentes entornos, tanto mediante programación textual como mediante programación por bloques.

Entre las opciones más habituales encontramos:

- Arduino IDE, utilizando lenguaje basado en C++.
- Entornos gráficos basados en Scratch.
- Plataformas educativas específicas del ecosistema Echidna.

En educación primaria y en los primeros cursos de secundaria suele utilizarse la programación por bloques, ya que permite centrarse en la lógica del programa sin necesidad de dominar la sintaxis de un lenguaje de programación.

Este enfoque facilita el desarrollo de habilidades relacionadas con el pensamiento computacional, como la descomposición de problemas, el uso de algoritmos o el reconocimiento de patrones.

## 4. EchidnaML y LearningML

### 4.1 Un entorno de programación educativo

Para facilitar el uso de estas placas en el aula, el proyecto Echidna incluye su propio entorno de programación: EchidnaML.

Este entorno está basado en programación por bloques similar a Scratch, lo que permite a estudiantes sin experiencia previa crear programas que controlen sensores y actuadores de la placa.

Una de las características más interesantes de este entorno es que puede utilizarse en local, lo que significa que puede instalarse en los equipos del centro educativo y utilizarse sin necesidad de conexión a internet. Esto resulta especialmente útil en contextos educativos donde la conectividad es limitada o donde se desea trabajar con datos de forma local.

---

### 4.2 Introducción a la inteligencia artificial con LearningML

Dentro del entorno EchidnaML se integra el módulo LearningML, que permite trabajar conceptos básicos de aprendizaje automático.

LearningML es una plataforma diseñada para introducir la inteligencia artificial en educación de forma accesible. Permite crear modelos sencillos de aprendizaje automático utilizando datos generados por el alumnado.

El proceso de trabajo suele seguir tres fases:

1. Recogida de datos

El alumnado recopila datos utilizando sensores, imágenes, sonidos u otras fuentes.

2. Entrenamiento del modelo

El sistema utiliza esos datos para generar un modelo de clasificación o predicción.

3. Uso del modelo

El modelo puede integrarse en un programa que controle dispositivos físicos.

Por ejemplo, se pueden crear proyectos como:

- Sistemas que detecten diferentes gestos utilizando el acelerómetro.
- Clasificadores de sonido usando el micrófono.

# Teachable Machine en el aula: entrenar con ejemplos (imagen, sonido y postura)



Teachable Machine es una herramienta gratuita y muy intuitiva de Google que permite **crear modelos de Inteligencia Artificial sin programar**. Dicho de forma sencilla: sirve para “enseñarle” a un ordenador a **reconocer** algo a partir de ejemplos, igual que hacemos en clase cuando el alumnado aprende a diferenciar colores, sonidos o animales por repetición y comparación.

Con Teachable Machine puedes entrenar modelos de tres tipos:

- **Imágenes:** por ejemplo, que reconozca “lápiz” vs. “goma”, o gestos como “mano arriba” vs. “mano abajo”.
- **Sonidos:** como “aplauso” vs. “silencio”, o “voz” vs. “ruido”.
- **Posturas:** movimientos del cuerpo captados por la cámara (ideal para juegos de psicomotricidad o dinámicas de aula).

## ¿Qué tipo de herramienta es?

Es una herramienta de **aprendizaje automático supervisado**: el docente (o el alumnado) crea **clases o categorías** (por ejemplo “sí / no”, “animal / planta”), aporta **ejemplos** para cada clase y la herramienta aprende patrones para poder **predecir** a qué clase pertenece un ejemplo nuevo. No “piensa” ni “entiende” como una persona: **clasifica** según lo que ha visto durante el

entrenamiento.

## ¿Qué proceso sigo en su utilización?

El proceso es muy parecido a una rutina didáctica:

1. **Elegir el tipo de proyecto** (imagen, sonido o postura).
2. **Crear categorías** (clases) con nombres claros.
3. **Recoger ejemplos** para cada categoría (con cámara, micrófono o archivos).
4. **Entrenar el modelo** (la herramienta “aprende” con esos ejemplos).
5. **Probarlo** con casos nuevos para ver si acierta.
6. **Mejorarlo**: añadir más ejemplos, equilibrar categorías y corregir errores.
7. **Usarlo o compartirlo**: se puede exportar y utilizar en actividades, juegos o proyectos (por ejemplo, conectándolo con Scratch).

## ¿Qué utilidad nos ofrece?

Porque permite trabajar la IA de forma **muy visual y práctica**, ayudando a comprender conceptos clave como **datos, entrenamiento, prueba, error, sesgo y mejora**, y además se adapta muy bien a Infantil y Primaria: se aprende “haciendo”, con ejemplos cotidianos del aula.

# Teachable Machine en Primaria

Infografía: Pros, contras y buenas prácticas para un uso seguro y útil en el aula

Aprendizaje por ejemplos

Sin programar

Prueba y mejora

## PROS (ventajas)

- ✔ Permite crear modelos de IA de forma visual y rápida (sin programar).
- ✔ Convierte la IA en algo 'manipulable': el alumnado ve datos, errores y mejoras.
- ✔ Ideal para trabajar clasificación, patrones, probabilidad y pensamiento crítico.
- ✔ Favorece metodologías activas: aprender haciendo, por equipos y con roles.
- ✔ Útil en muchas áreas.
- ✔ Facilita explicar conceptos clave: sesgo, generalización, sobreajuste, 'predicción'.
- ✔ Se puede usar en directo en el monitor: motiva y genera debate inmediato.

## CONTRAS (riesgos y límites)

- ✘ Si los ejemplos están desequilibrados o son pobres, el modelo se sesga y falla.
- ✘ No 'entiende': puede dar falsas seguridades (aciertos con baja calidad).
- ✘ Riesgos de privacidad si se usan caras, nombres, voces identificables o datos personales.
- ✘ Dependencia de condiciones: luz, fondo, ruido y cámara/micrófono afectan mucho.
- ✘ Puede confundir categorías parecidas (especialmente al inicio) y frustrar si no se guía.
- ✘ Requiere tiempo mínimo de preparación y pruebas para que salga bien en clase.
- ✘ Si se comparte el enlace del modelo sin control, puede circular fuera del aula.

## Buenas prácticas (para que funcione a la primera)

- ✔ Empieza con 2 clases muy distintas (p. ej., triángulo vs cuadrado) y objetos NO personales.
- ✔ Equilibra ejemplos por clase y añade variedad (ángulos, tamaños, fondos, distancia).
- ✔ Prueba con ejemplos nuevos y registra resultados (tabla de aciertos y 'por qué falló').
- ✔ Si falla, no es 'magia': añade ejemplos del error, mejora luz/fondo y vuelve a entrenar.
- ✔ En monitor interactivo: portátil + HDMI/USB táctil + webcam externa si hace falta.



# Teachable Machine vs EchidnaML: ¿cuál elegir según tu aula?

A continuación, presentamos distintas plataformas para trabajar el machine learning o aprendizaje supervisado.

	<b>Learning ML (Echidna en local)</b>	<b>Learningml 2.0</b>	<b>Teachable machine</b>
Uso	Local	<a href="https://learningml.org/lml-editor/">https://learningml.org/lml-editor/</a>	<a href="https://teachablemachine.withgoogle.com/">https://teachablemachine.withgoogle.com/</a>
Tipo de herramienta	Aplicación de escritorio integrada	Plataforma web de Machine Learning educativo	Plataforma web de aprendizaje automático
Conexión a internet	No necesaria	Necesaria	Necesaria
Programación	Programación por bloques basada en Scratch (EchidnaBlocks)	Integración con Scratch/EchidnaScratch	No incluye entorno de programación propio
Integración con Hardware	Directa con placas Echidna	Sí, mediante herramientas web y enlace con la placa	No directa (solo exportación a otros sistemas)
Tipos de modelos de IA	Texto, imágenes y datos numéricos	Texto, imágenes y datos numéricos	Imágenes, audio y poses (movimiento corporal)
Entrenamiento de modelos	En el propio ordenador	En la plataforma web	En el navegador del usuario
Privacidad y datos	Los datos se procesan localmente en el ordenador	Los datos se procesan en servidores web	Procesamiento principalmente en el navegador (modelo Cliente)*
Integración con robótica	Muy alta	Alta	Baja

## **Acerca de la privacidad de teachable machine:**

Los modelos creados en Teachable Machine se entrenan localmente en el navegador del



usuario y pueden alojarse en Google Drive o en los servidores de Google (mediante una URL pública) al exportarlos. Respecto a la privacidad, las imágenes de entrenamiento no se comparten ni se suben a servidores por defecto durante el proceso de entrenamiento.

### ¿Dónde se alojan los modelos?

Entrenamiento (Local): Todo el proceso de entrenamiento ocurre dentro de la pestaña de tu navegador (client-side) utilizando TensorFlow.js.

Almacenamiento del proyecto: Puedes guardar tus proyectos (clases e imágenes) en tu cuenta personal de Google Drive.

Publicación/Alojamiento: Al hacer clic en "Upload my model" (subir mi modelo), Google aloja el modelo entrenado en sus servidores y genera una URL compartible.

Descarga: Puedes descargar el modelo entrenado (como TensorFlow.js, TensorFlow o TensorFlow Lite) para usarlo de manera totalmente local.

### ¿Se comparten las imágenes de entrenamiento?

Privacidad por defecto: Las imágenes de tu webcam o los archivos que subes no abandonan tu ordenador durante el entrenamiento.

Guardado en Drive: Si eliges guardar tu proyecto en Google Drive, las imágenes se guardan allí, pero siguen siendo de tu propiedad.

Publicación de modelo: Cuando publicas el modelo para obtener una URL, **solo se sube el modelo matemático (el programa que predice), no las imágenes de ejemplo que utilizaste para entrenarlo.**

En resumen, Teachable Machine es seguro y diseñado para la privacidad, manteniendo los datos de entrenamiento en tu equipo a menos que decidas explícitamente guardarlos en la nube.

## CreateAI: la IA de micro:bit para reconocer movimiento

**CreateAI** es una herramienta web del ecosistema de la placa **BBC micro:bit** que permite crear modelos de **aprendizaje automático basados en movimiento**.

La idea principal es utilizar el **acelerómetro de la micro:bit** para recoger datos de movimiento y entrenar un modelo capaz de reconocer patrones físicos.

### Cómo funciona

El proceso de trabajo sigue tres fases principales:

#### 1. Recoger datos (Collect data)

El alumnado realiza movimientos con la micro:bit (por ejemplo: aplaudir, saludar o saltar).

El acelerómetro registra los datos.



## 2. Entrenar el modelo (Test model)

La herramienta utiliza esos datos para entrenar un modelo de aprendizaje automático que aprende a distinguir los movimientos.

## 3. Programar con el modelo (Code)

El modelo se integra en el entorno **MakeCode** mediante bloques específicos de Machine Learning.

---

## Qué tipo de proyectos permite

Con CreateAI el alumnado puede crear proyectos como:

- un interruptor que enciende luces cuando detecta un aplauso,
- un temporizador que reconoce actividades físicas,
- un contador de pasos o movimientos,
- juegos que responden a gestos.

En esencia, el sistema permite que la placa **interprete el movimiento del cuerpo humano como una entrada inteligente**, convirtiendo gestos físicos en eventos programables.

---

## Valor educativo

CreateAI es especialmente interesante en el aula porque:

- introduce el **concepto de entrenamiento de modelos de IA**,
- trabaja **datos reales recogidos por sensores**,
- conecta **inteligencia artificial y computación física**,
- y permite comprender la importancia de **los datos de entrenamiento**.

Esto lo convierte en una herramienta muy potente para enseñar **machine learning de forma tangible**, ya que el alumnado puede ver cómo la máquina aprende directamente de sus propios movimientos.



# Actividad 1. Vivo o no vivo: entrenamos un clasificador con objetos del aula



Ilustración 1 Imagen generada con ChatGPT (OpenAI). 2026.

# Vivo / No vivo con objetos del aula

## Justificación

Esta actividad acerca la IA al aula de forma manipulativa: el alumnado “enseña” a una máquina con fotos de objetos reales. Comprenden que la IA aprende con ejemplos (datos) y que puede equivocarse si los datos son pocos o poco variados. Además, refuerza la clasificación científica básica y el pensamiento crítico (“¿por qué falla?”).

## Datos de la actividad

- Curso: 1.º Primaria (6 años)
- Área: Ciencias
- Tipo TM: Proyecto de imagen (clasificación)
- Tiempo: 35-50 min (recomendable realizar en 2 sesiones)
- Agrupación: Equipos + asamblea final
- Producto: Clasificador VIVO vs NO VIVO

## Objetivo didáctico

Entrenar y probar un modelo de IA que clasifique objetos como VIVO o NO VIVO, entendiendo que el acierto depende de las fotos usadas para aprender, detectando errores y mejorándolo con nuevos ejemplos.

## Vinculación Curricular



Esta actividad une el área de **Conocimiento del Medio** (clasificación de seres vivos e inertes) que nos permite trabajar la **Competencia Digital** en el aula con nuestro alumnado. Al entrenar un modelo con sus propias fotos, el alumnado descubre de forma práctica que la IA no es 'magia', sino que aprende de los datos que nosotros le damos. Esto desarrolla el pensamiento computacional y la actitud crítica desde 1.º de Primaria, transformando a los niños de simples usuarios en creadores de tecnología capaces de detectar errores y mejorar el aprendizaje de la máquina.

## Materiales

- Objetos “vivos o relacionados”: hoja, flor, fruta, ramita, planta...
- Objetos “no vivos”: piedra, goma, lápiz, clip, borrador...
- Carteles o cajas: VIVO / NO VIVO
- Dispositivo con webcam + Teachable Machine (y monitor/proyector si se usa en gran grupo)
- Tabla/registro (con □ / □ )

## Desarrollo de la actividad

### 1. Pregunta detonante

“¿Creéis que una máquina puede aprender a decir si algo está VIVO o NO VIVO solo mirando fotos? ¿Qué necesita para aprender?”

Este debate inicial nos sirve como introducción para dar paso a la actividad.

### 2. Organización

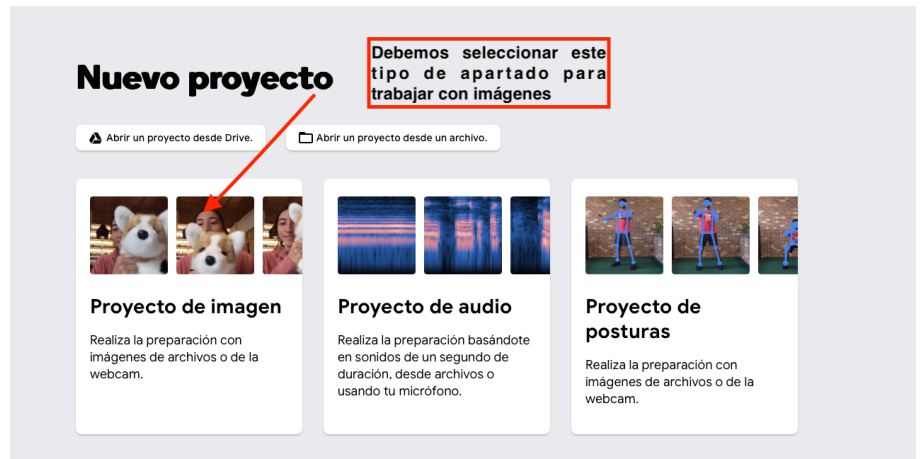
Roles sencillos (elegibles):

- **Fotógrafo/a:** acerca el objeto a la cámara
- **Ayudante de orden:** coloca objetos en VIVO/NO VIVO
- **Portavoz:** dice el resultado (“ha dicho VIVO”)
- **Observador/as:** detecta errores (“se ha confundido”)
- 

### 3. Pasos en Teachable Machine

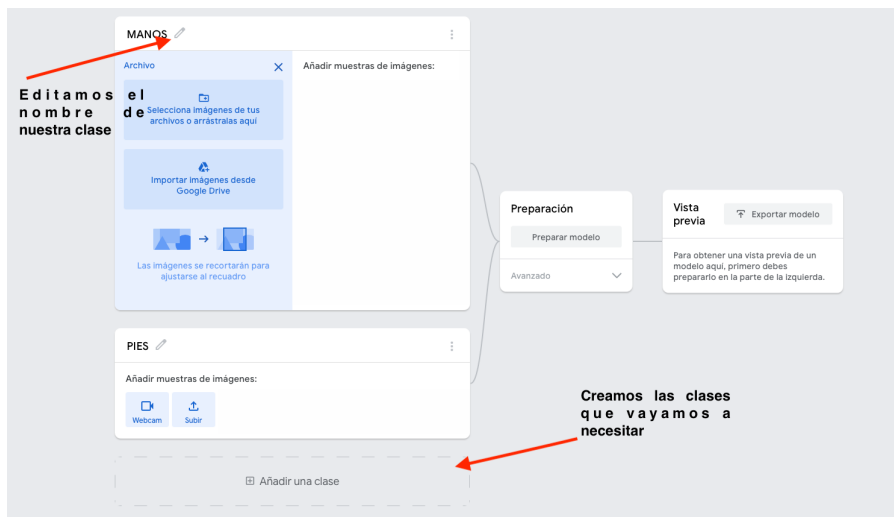
#### a) Crear proyecto y clases

- Proyecto de imagen —> Modelo de imagen estándar



Teachable machine. (s. f.). <https://teachablemachine.withgoogle.com/>

- Clases: VIVO y NO VIVO



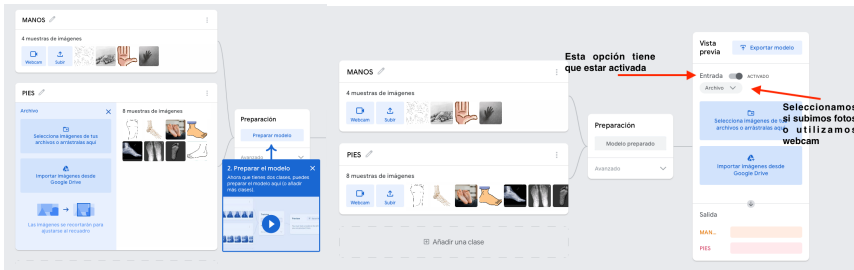
Teachable machine. (s. f.). <https://teachablemachine.withgoogle.com/>

## b) Recoger datos (fotos)

- Tomar 10 ejemplos por clase.
- Variar objetos, ángulos, distancia y fondo (para que aprenda mejor)

## c) Entrenar y probar

- Preparación del modelo.
- Probar con objetos nuevos (no usados en las fotos iniciales)



Teachable machine. (s. f.). <https://teachablemachine.withgoogle.com/>

#### d) Mejorar

7. Cuando falle, añadir fotos de “lo difícil” (hoja seca, piedra distinta, fruta cortada...)
8. Re-entrenar y volver a probar.

DUA (representación):

- Apoyos visuales: pictogramas “VIVO/NO VIVO”, ejemplo real en mano, carteles grandes.
- Lenguaje claro y repetido: “La máquina aprende con muchas fotos”.

#### 4. Reto de comprobación (objeto sorpresa)

- Cada equipo trae un objeto sorpresa.
- Lo muestran a la cámara y observan predicción
- El portavoz dice: “La máquina ha dicho...” y el grupo decide: “¿Estamos de acuerdo?”

El "**Objeto Trampa**": El docente saca un objeto que "engaña" a los sentidos (ej. una flor de plástico o un peluche de animal). Lo prueba: y el alumnado comprobar qué dice la IA. Si dice "VIVO", abrimos el debate: "¿Por qué se ha equivocado si tiene pétalos/ojos?". **Conclusión:** Ayuda a entender que la IA solo compara formas y colores y a veces necesita nuestra ayuda para mejorar.

DUA (acción y expresión): Se puede explicar de 3 formas:

- Oral: “se ha confundido porque...”
- Señalando: “por el color / por la forma / por la luz” (tarjetas)
- Con dibujo rápido: “nuestro objeto” + carita (acierta/falla)

#### 5. Opciones de mejora propuestas por el alumnado

Se pueden pedir 2 ideas por equipo o como ayuda al alumnado podemos plantear o darle un “menú de mejoras” visual para que todos puedan elegir.(pueden elegir con tarjetas):

- “Más fotos”
- “Fotos desde otros lados”
- “Cambiar el fondo”
- “Cambiar la luz”
- “Meter objetos nuevos parecidos” (casos límite)

Luego eligen 1 para aplicar.

## 6. Comprobar mejoras (segunda ronda)

En esta segunda ronda el alumnado a través de la mejora propuesta comprobará si el modelo ha cambiado y como lo ha hecho.

- Re-entrenar y repetir 3 pruebas:
  1. objeto fácil, 2) objeto difícil, 3) objeto sorpresa
- Comparar en voz alta: “Antes fallaba / ahora acierta más”.

## 7. Debate final (asamblea breve)

En este punto se trata de crear debate con el alumnado y que aporte su propia visión sobre la actividad realizada a través de distintas preguntas que planteemos como por ejemplo:

- **¿La máquina acertó mucho o poco?**
- **Cuando se equivocó, ¿qué creéis que miró?**
  - “Color” / “Forma” / “Fondo” / “Luz”
- **¿Qué le ayudó a mejorar?**
  - “Más fotos” / “Fotos diferentes” / “Objetos nuevos”

Cierre (El Espejo): "Si yo le enseño a la IA solo fotos de manzanas rojas y luego le enseño una manzana verde... ¿sabrá qué es?". Esto refuerza la idea de que la IA solo sabe lo que nosotros le enseñamos (dependencia de datos). Cierre para el aula:

“La IA no entiende ‘vida’ como nosotros: aprende patrones con las fotos que le damos.”



## 8. Evaluación final por grupos: Semáforo

Semáforo del equipo (con tarjeta o dibujo)

Cada grupo elige un color y completa una frase (con apoyo del docente):

- VERDE: “Nuestro modelo acertó muchas veces.”
- AMARILLO: “Nuestro modelo a veces acierta, a veces falla.”
- ROJO: “Nuestro modelo falló bastante.”

El grupo marca o dice una causa :

- “Necesitábamos más fotos”
- “Necesitábamos fotos diferentes”
- “El fondo engañó”
- La luz era mala
- “Nuestro objeto era nuevo/diferente”

Podemos hacer una previsión e respuestas y que el alumnado pueda elegir o crear su propia respuesta a partir de las que hemos preparado.

Evidencia rápida (para el docente)

Criterios de Éxito	Equipo 1	Equipo 2	Equipo 3	Equipo 4	Equipo 5
1. Calidad de Datos: Aportó fotos variadas a ambas clases (V/NV).					
2. Generalización: Probó el modelo con al menos 2 objetos nuevos.					
3. Iteración: Propuso y aplicó 1 mejora (más fotos, luz, fondo...).					
4. Pensamiento Crítico: Explicó un error (color, forma, luz, fondo).					



Puntuación Final (Semáforo)	□ □ □ □	□ □ □ □	□ □ □ □	□ □ □ □	□ □ □ □
--------------------------------	---------	---------	---------	---------	---------

# Actividad 2. Radar Sonoro: entrenamos un clasificador de sonidos del entorno



Ilustración 1 Imagen generada con ChatGPT (OpenAI). 2026.

## Radar Sonoro: IA en modo escucha

Esta actividad usa Teachable Machine (proyecto de audio) para entrenar un clasificador de sonidos del entorno (aves, tráfico y agua). El alumnado comprende que **la IA no entiende el mundo: aprende patrones** a partir de ejemplos, puede confundirse y mejora cuando ampliamos y equilibramos los datos.

## Datos de la actividad

- Curso: 5.º Primaria (10-11 años)
- Área: Conocimiento del medio natural, social y cultural
- Tipo TM: IProyecto de sonido
- Tiempo: 55-60 min (recomendable realizar en 2 sesiones)
- Agrupación: Equipos + asamblea final
- Producto: Clasificador de sonidos
- Objetivo general

## Objetivo didáctico

Con esta actividad se pretende que el alumnado construya un modelo sencillo que reconozca sonidos y aprenda a trabajar con el ciclo: definir clases - recopilar datos - entrenar - probar - mejorar. La actividad enfatiza pensamiento crítico: que registren errores, expliquen por que ocurren y decidan que datos faltan para mejorar.

## Vinculación curricular

La situación de aprendizaje '**Radár Sonoro**' se integra de forma transversal en el currículo de 5º de Primaria, vinculándose directamente con el área de **Conocimiento del Medio Natural, Social y Cultural** y la **Competencia Digital (CD)**. A través del diseño y entrenamiento de un modelo de aprendizaje supervisado, el alumnado trabaja el **Criterio de Evaluación 3.2**, orientado al diseño de soluciones tecnológicas para problemas del entorno, y la **Competencia Específica 5**, que promueve el uso crítico, ético y responsable de la tecnología. La actividad nos permite igualmente el trabajo del **pensamiento computacional** y la alfabetización de datos, permitiendo que el alumnado identifique los sesgos y limitaciones de los algoritmos de IA.

## Materiales

- Monitor interactivo o proyector + ordenador (portátil) con navegador.
- Micrófono (el del portátil suele valer; mejor microfono externo si el aula es ruidosa).
- Tres colecciones de sonidos (20-40 clips por clase): aves, tráfico, agua. Preferible grabados por el docente o del centro, sin voces identificables.



- Altavoces para reproducir los clips durante la prueba.
- Hoja de registro por equipo (incluida al final del documento).
- Opcional: rincón silencioso para grabación; cartel visual de normas de audio (no nombres, no conversaciones).

## Desarrollo de la actividad

### 1. Pregunta detonante

- Si cierro los ojos, puedo saber si oigo un pajarito, un coche o agua. Como crees que lo haría una IA?
- ¿Qué necesita para aprender a distinguir sonidos?
- ¿Crees que se equivocara? ¿Qué podemos hacer si falla?

### 2. Organización y roles

- Equipo de 4:
  - **Operador/a** (usa TM),
  - **Gestor/a de datos** (elige y etiqueta clips),
  - **Probador/a** (lanza pruebas),
  - **Anotador/a** (registra resultados).

Norma de privacidad: no grabar voces ni nombres; si hay voces en un clip, se descarta.

### 3. Preparación de datos

- Cada clase debe tener cantidad similar de clips (por ejemplo 20/20/20).
- Variedad: aves distintas, tráfico cercano y lejano, agua en fuente/lluvia/agua corriente.
- Calidad: clips cortos y claros (1-2 s) con el menor ruido posible.
- Se recomienda usar una carpeta con clips preparados (recomendado para la primera vez). Podría utilizarse un micrófono conectado pero no se recomienda para esta actividad.

Puedes usar y descargar sonidos desde este enlace [INTEF](#)

## 4. Pasos en Teachable Machine

- Entrar en Teachable Machine y elegir **proyecto de audio**.
- Renombrar clases: AVES, TRAFICO, AGUA (opcional: RUIDO MIXTO).
- Para cada clase, subir o grabar clips hasta llegar a la meta acordada.
- Entrenar: pulsar **preparar modelo**.
- Prueba inicial: reproducir clips nuevos (no usados) y anotar prediccion y porcentaje.

## 5. Reto de comprobación

- Reto por equipos: cada equipo selecciona 6 clips sorpresa (2 por clase) y los prueba en el modelo del aula.
- Se registra: clip, prediccion, porcentaje, acierto/fallo.
- Pregunta clave: en que sonidos se equivoca mas? (por ejemplo, agua fuerte vs trafico lejano).

## 6. Mejora guiada y segunda ronda

El grupo decide una mejora basada en evidencias y repite el ciclo una vez:

- Si confunde agua con trafico: anadir mas ejemplos de agua con diferentes intensidades y fondos.
- Si el aula tiene ruido: crear clase RUIDO MIXTO y meter ejemplos parecidos para que el modelo no los etiquete como aves/trafico/agua.
- Equilibrar datos: si una clase tiene menos clips, completarla antes de reentrenar.
- Reentrenar y repetir la prueba con los mismos 6 clips sorpresa para comparar.

## 7. Cierre y debate final

- Que ha mejorado tras anadir datos?
- Que limites tiene este modelo (ruido, sonidos mezclados, microfono, distancia)?



- En que situaciones reales seria util un clasificador de sonidos? En cuales seria peligroso confiar sin supervision?

No te preocupes si el modelo falla. De hecho, **es mejor que falle**. Un modelo perfecto no genera debate; un modelo que confunde un grifo con un motor de coche es la oportunidad perfecta para explicar que la IA no tiene sentido común, solo matemáticas.

## 8. Accesibilidad e inclusión (DUA aplicado)

- **Representación: apoyo visual** de categorías con pictogramas; ejemplos de sonido etiquetados (tarjetas AVES/TRAFFICO/AGUA).
- **Acción y expresión:** participar grabando, anotando, decidiendo en equipo o explicando oralmente; opción de responder señalando.
- **Implicación:** roles rotatorios y retos graduados (2 clases al inicio; 3 clases para extension; clase RUIDO MIXTO para avanzado).
- **Apoyos:** plantillas de registro con iconos, tiempo extra para quien lo necesite y espacio mas silencioso para grabar.

## 9. Evaluación

### Instrumentos de registro (para llevar a la practica)

Puedes utilizar esta rubrica y plantilla de registro para cada equipo. Sirve para registrar entrenamiento, pruebas y mejoras.

#### Rubrica rapida por equipo

Indicador	En proceso	Logrado	Avanzado
Datos	Pocos clips o conruido/mezcla	Clips suficientes y claros	Variedad alta y datos equilibrados
Analisis	Describe aciertos sin explicar	Explica errores conuna causa	Relacionacausas y propone mejoras precisas
Mejora	Reentrena sin cambiar datos	Cambia datos ycompara	Itera 2 veces y documenta evidencias

### Plantilla de registro (pruebas)

Sugerencia: realiza 12 pruebas (4 por clase) antes y después de la mejora. Marca acierto y anota el porcentaje.

#	Clip (descripcion)	Prediccion	%	Acierto	Nota / causa del error
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

Cierre recomendado: cada equipo comparte 1 error frecuente y 1 mejora que funciona. Así la clase construye un 'manual de buenas practicas' para futuros modelos.

## Recomendaciones finales

- **Respetar la privacidad:** sin voces, sin nombres, sin conversaciones en los clips.
- **Equilibrar ejemplos** por clase y explica por que es importante.
- **Registrar resultados** (predicción, porcentaje, acierto) y detecta patrones de error.
- **Proponer al menos una mejora** basada en evidencias y comprueba si mejora el rendimiento.



- **Explicar con lenguaje propio que la IA no entiende:** predice según ejemplos.

# Actividad 3. Entrenar con datos sesgados: cuando la IA aprende "mal"

En esta propuesta, además de **afianzar como funciona el aprendizaje supervisado**, tiene como objetivo comprender cómo pueden llegar a **formarse sesgos** en las respuestas de la IA ya sean de un modo voluntario o involuntario.

Si el alumnado aun está interiorizando esta clase de conceptos, se recomienda realizar la [propuesta didáctica de primer ciclo](#)

## Entrenando a nuestra IA con sesgos

### Objetivos:

- Comprender que los sistemas de IA aprenden de los datos que les damos
- Reconocer que si los datos están sesgados, la IA puede tomar decisiones injustas o incorrectas.
- Desarrollar pensamiento crítico sobre la tecnología.

### Recursos materiales:

- Ordenadores con cámara web o tablets, con acceso a [EchidnaML \(local\)](#)
- [Cartulinas o tarjetas con imágenes de diferentes especies](#), en este caso perros blancos, perros marrones, primates y perros negros.
- [Ficha de reflexión](#)

**Temporalización:** 60 minutos

## Introducción

En primer lugar, explicaremos brevemente qué es una Inteligencia Artificial y cómo aprende, lo que viene a ser el aprendizaje supervisado. Por ejemplo podríamos decirles que *"la IA es como un robot que aprende viendo muchos ejemplos, igual que vosotros y vosotras aprendéis a reconocer animales viendo fotos o videos"*.



Una vez llegado a este punto, introduciremos el concepto de sesgo: "si le damos al robot solo fotos de gatos blancos, puede pensar que todos los gatos son blancos. Eso es un sesgo."

## Preparando el experimento

Para comenzar con el experimento, haremos agrupaciones de 4-5 alumnos. Cada uno de los grupos se encargará de entrenar a una IA para clasificar animales según especie como la que hemos incluido en el apartado recursos. La mitad de los grupos de alumnos recibirá un set de cartas, la otra mitad otro.

- **SET A:** Perros blancos, perros marrones y primates.
- **SET B:** Perros, blancos, perros marrones, **perros negros** y primates.

**Importante:** la mitad de los grupos recibirán datos sesgados a propósito (por ejemplo, sólo animales de un tipo) y otros datos más variados (SET A vs SET B)

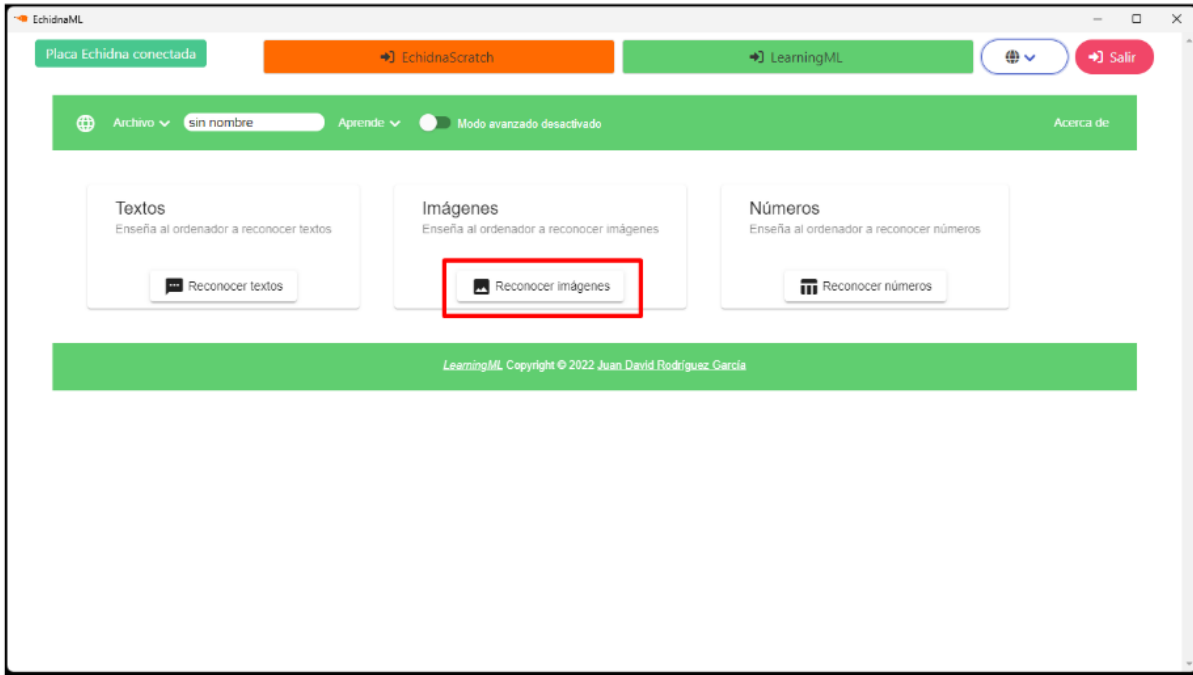
## Alimentando a la IA

Los grupos, cargarán sus imágenes en LearningML.

Para ello, abriremos EchidnaML y entraremos en su apartado LearningML (para esta práctica no es necesaria la placa echidna).

➔ LearningML

Escogemos el entrenamiento de imágenes



Entrenan el modelo con las categorías en función de las cartas recibidas: PERRO BLANCO, PERRO MARRÓN, PRIMATE y si es el caso, PERRO NEGRO.



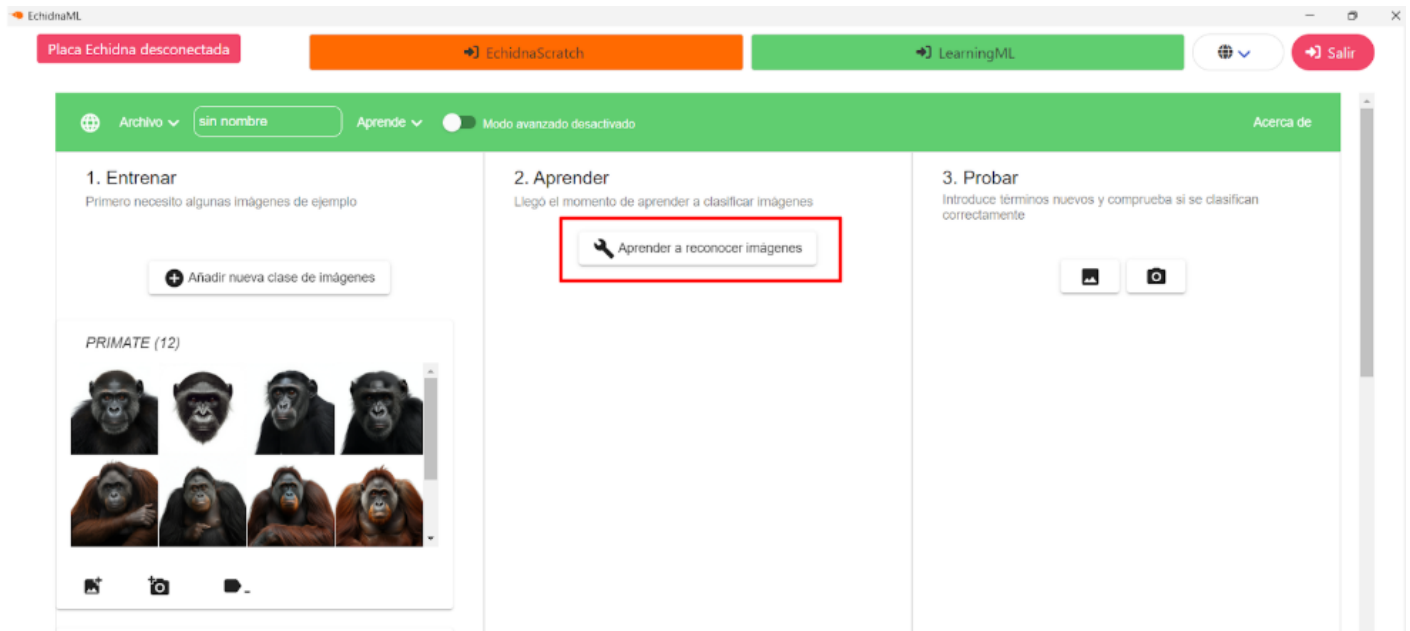
Escogemos un nombre para esta categoría, por ejemplo "PERRO BLANCO" y ahora tenemos dos opciones para alimentarla, o bien activar la webcam para capturar las tarjetas impresas o bien subir las imágenes en formato digital directamente.



Una vez están subidas las imágenes de esa categoría, crearíamos una nueva categoría y repetiríamos el proceso.

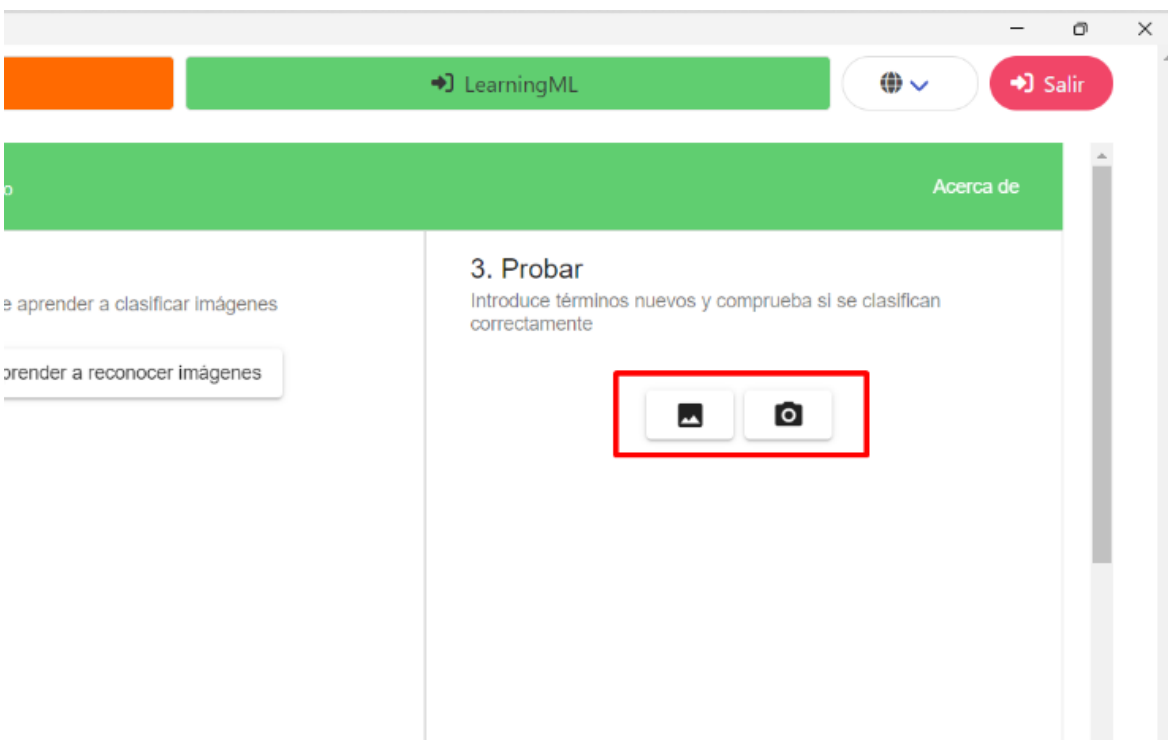
**NOTA\*:** En ocasiones es recomendable crear una categoría en la que no haya ningún elemento a aprender, sobre todo cuando se utiliza la opción webcam para evitar confusiones. Se puede llamar a esta categoría como "NADA" para que todo lo que no identifique con las otras categorías vaya ahí.

Cuando hemos terminado el entrenamiento (es decir, hemos terminado de alimentar a la IA con imágenes), es el momento de hacerle que aprenda el modelo:



Finalmente, observamos cómo la IA clasifica nuevas imágenes de prueba.

Para ello, nos vamos a **PROBAR** y tenemos dos opciones, subirle nuevas imágenes o mostrárselas a la webcam.



En este caso vamos a subir una imagen de un perro negro (categoría que no ha sido creada, pues era un set de tarjetas del grupo A).

### 3. Probar

Introduce términos nuevos y comprueba si se clasifican correctamente



- PRIMATE (97.97 %)
- PERRO MARRON (2.00 %)
- PERRO BLANCO (0.03 %)

Aquellos grupos que hayan recibido tarjetas de perros negros y hayan decidido crear la categoría perro negro, obtendrán respuestas acertadas mientras que aquellos que únicamente recibieron fotografías de perros blancos, marrones y primates pero no de perros negros, interpretará a estos como primates (por similitud -color-)

### Observación y discusión (10-15 min)

1. Cada grupo prueba su IA con imágenes fuera de su conjunto de entrenamiento.
2. Comparan los resultados:
  - ¿Qué pasó cuando las imágenes de prueba eran diferentes de las que entrenaron?
  - ¿La IA acertó siempre? ¿Falló? ¿Por qué creen que pasó eso?
3. Reflexionan sobre sesgos:
  - “Si entrenamos solo con fotos de perros blancos, ¿la IA reconocerá a los perros negros?”
  - ¿Por qué uno de los grupos fallaba?
  - ¿Cómo se puede solucionar este problema?

# Actividad 4. IA + robótica con Echidna: la barrera inteligente (reconocimiento y acción)

## Justificación

En educación primaria solemos aprender primero haciendo y después pensando sobre lo que hemos hecho. Sin embargo, cuando trabajamos con machine learning e inteligencia artificial aplicada a la robótica en el tercer ciclo de primaria, en este capítulo se propone hacerlo al revés: empezar por entender la idea y acabar en la acción del robot.

La inteligencia artificial funciona a partir de cosas que no se ven directamente, como aprender con datos, reconocer patrones o decidir entre varias opciones. Si el alumnado solo ve lo que hace el robot, puede pensar que “funciona solo” o que es magia. Por eso es importante que antes entiendan, de forma sencilla, qué está pasando por dentro.

Cuando esas ideas se llevan al robot y este actúa, el aprendizaje se vuelve mucho más claro y cercano. Ver cómo el robot hace exactamente lo que se ha entrenado permite conectar la teoría con algo real, haciendo que el aprendizaje sea más significativo, comprensible y motivador.

A continuación, veremos una propuesta didáctica del curso de aularagon de Echidna, que cumple con estas premisas.


## Propuesta didáctica con Echidna

Vamos a desarrollar este proyecto de Jorge Lobo @Lobo\_tic

[https://x.com/lobo\\_tic/status/1923996982516814064](https://x.com/lobo_tic/status/1923996982516814064)

“ —Antes, para salir del parking del supermercado, había que escanear el ticket de compra, pero ahora la barrera se abre sola como si supiera que ese coche puede salir.

—¿Queréis saber cómo lo sabe?

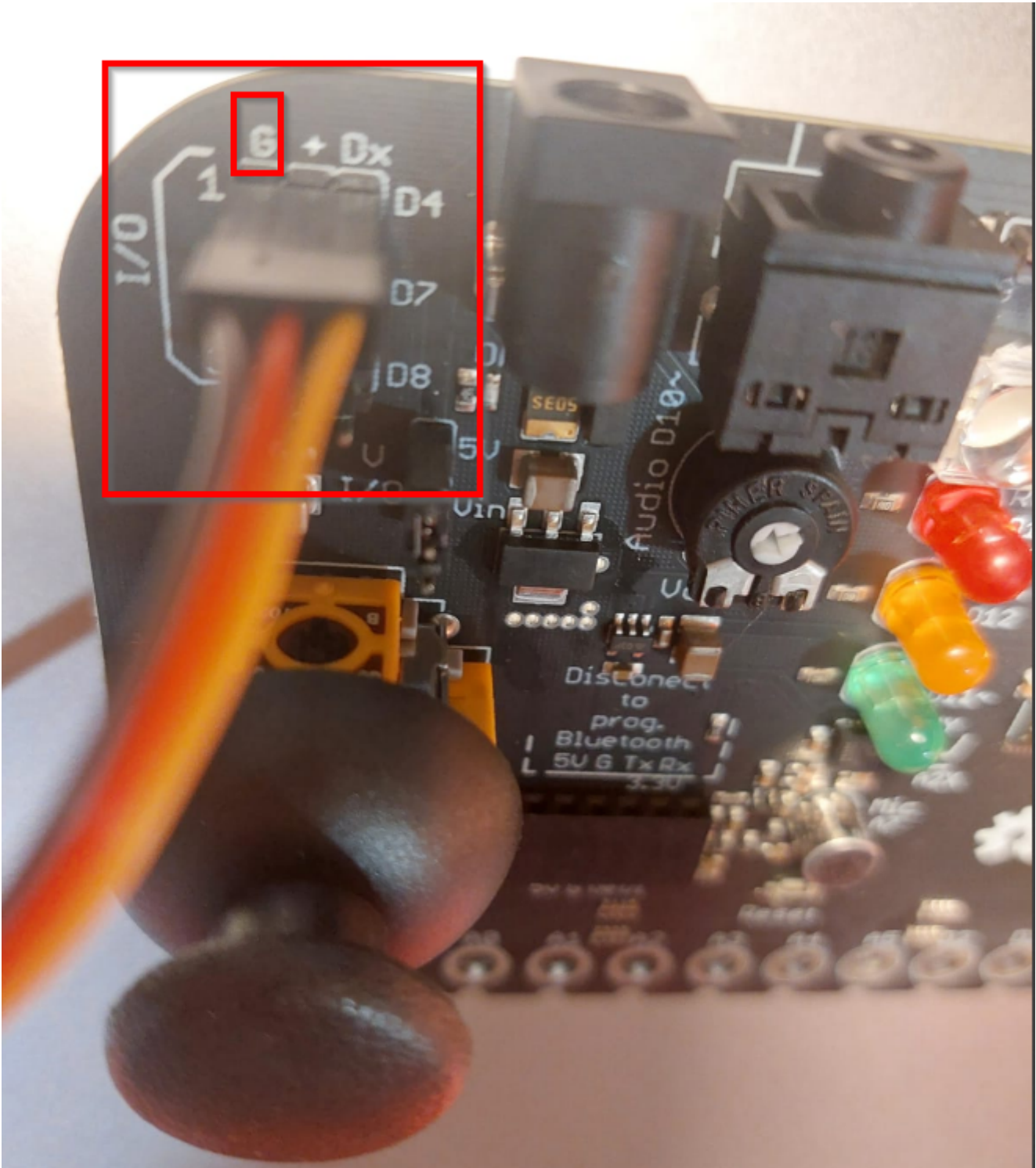
Así surge la idea de esta actividad con [@EchidnaSTEAM](#) y [@learningml](#)   
[pic.twitter.com/3Ew22WcURj](https://pic.twitter.com/3Ew22WcURj)

— Jorge Lobo (@lobo\_tic) [May 18, 2025](#)

## PRIMER PASO : MONTAJE

Ponemos un servo en D4

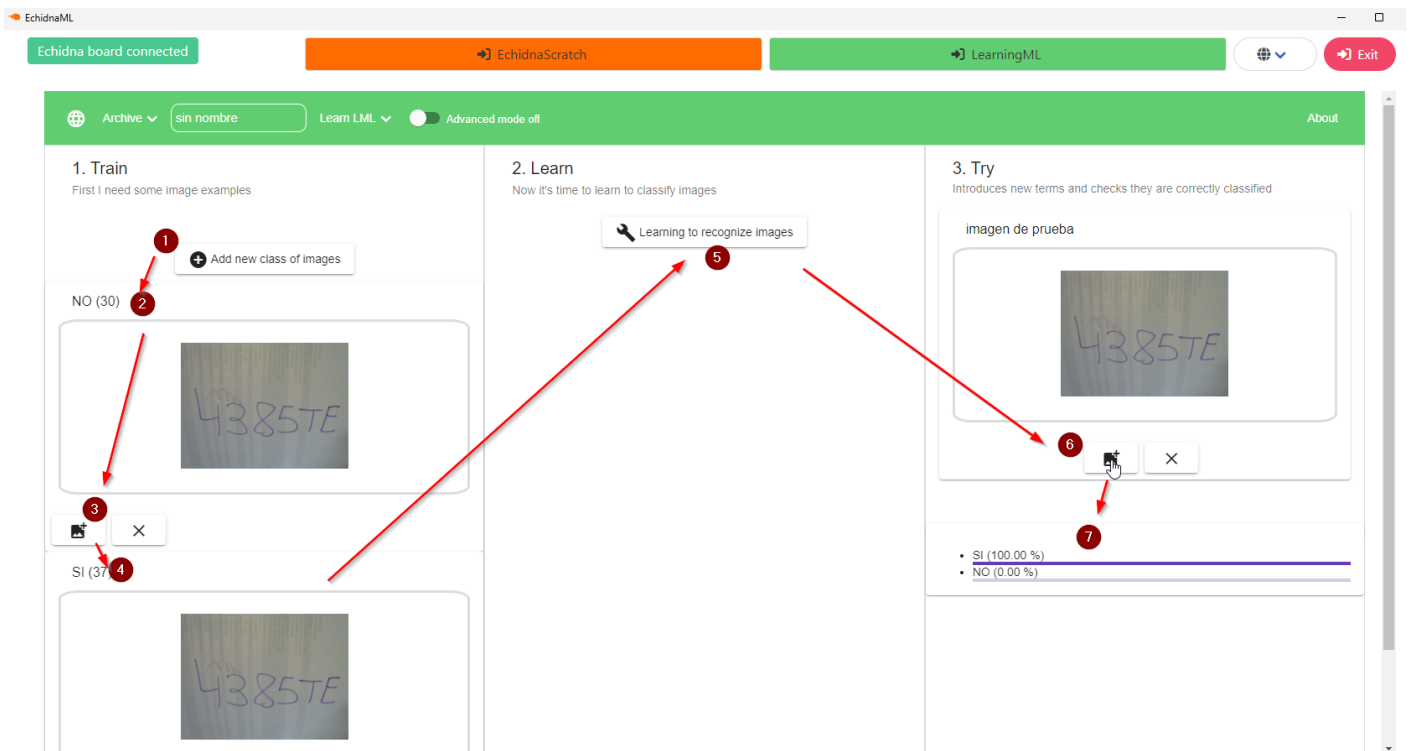
FÍJATE QUE EL **MARRÓN** A MASA GND SIMBOLO G TAL COMO LA FOTO Y EL AMARILLO EN DATO Dx



## SEGUNDO PASO : CREAR MÁQUINA IA DE RECONOCIMIENTO DE MATRICULAS

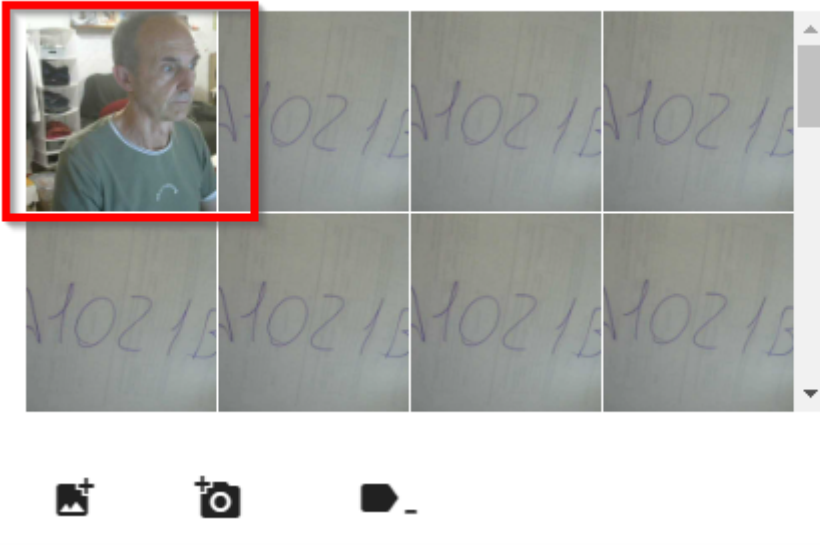
1. Entramos en crear máquina de reconocer imágenes
2. Creamos una clase que se llame **NO**
3. Añadimos imágenes desde nuestra webcam

1. Vamos enseñando matrículas que **no queremos que se abra el parking** de cada matrícula pulsamos una docena en diferentes posiciones (ligeros cambios)
2. También añadimos varias imágenes que existe cuando no hay ninguna matrícula, pues queremos que cuando no hay coche que no se abra el parking.
4. Añadimos la clase **SI**
  1. Repetimos el paso 3.1 con las matrículas que **queremos que se abra el parking**
5. Le damos a aprender
6. Probamos
7. Nos fijamos que funciona perfectamente



Detalle del paso 3.2

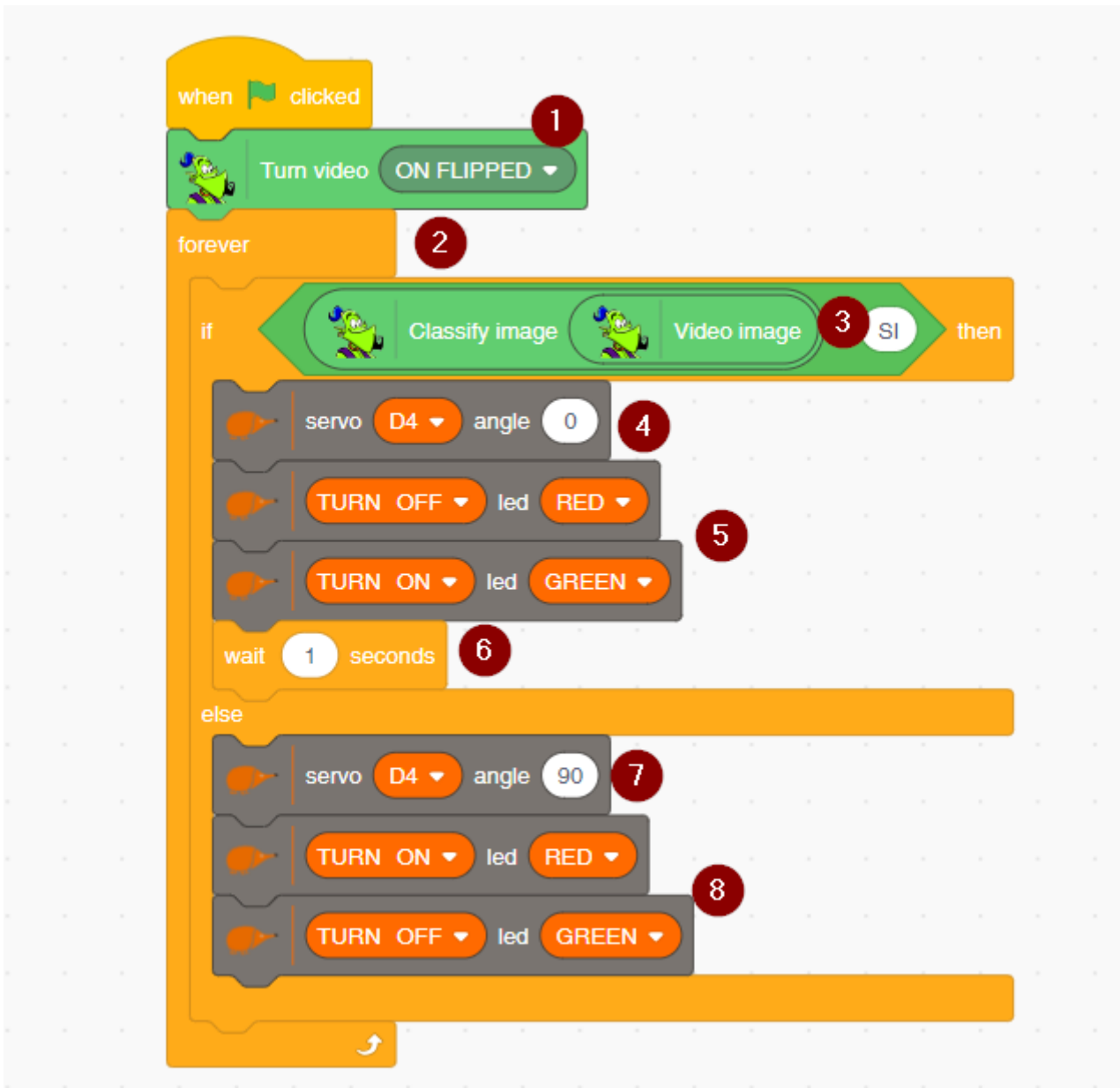
NO (30)



## TERCER PASO : PROGRAMAR

Ahora entramos en la ventana de programación usando el modelo:

1. En **learningml** arrastramos la instrucción de **encender cámara**
  1. curiosamente lo voltea, por lo tanto le decimos ON FLIPPED de lo contrario lee al revés las matrículas (curiosamente lo hace bien aunque a veces falla)
2. Ponemos un bucle infinito para que siempre lea los sensores y funcionen los actuadores (típico en programas de robótica)
3. Un condicional dentro del bucle si --- sino y en la condición que sea si LA MATRICULA ES DE LA CLASE **SI** pues que abra sino que la cierre
  1. Tiene dos instrucciones dentro de learningml uno que es que clasifique la imagen y otro que sea la imagen de la captura del vídeo
4. Si es SI pues que abra con el servo
5. También que encienda la luz verde y apague el rojo
6. Esta espera es para que el coche pase tranquilamente además que nos permite que no haga continuos aperturas y cierres de la barrera
7. SINO pues que cierre la barrera
8. Encienda rojo y apague verde



## CUARTO PASO : JUGAR

<https://www.youtube.com/embed/Bkvj0yK6t3A>

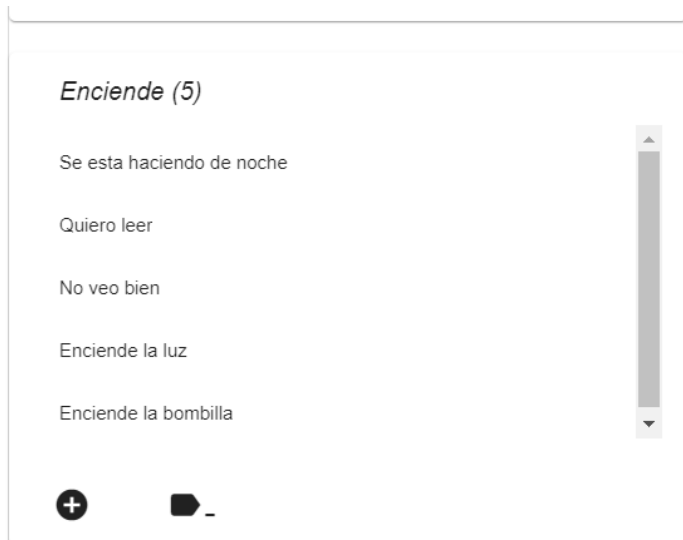
# Actividad 5. La lámpara inteligente: entrenamos un modelo de texto (enciende/apaga) y lo mejoramos con sinónimos

Vamos a encender y apagar un objeto de luz usando texto predictivo con IA

Al arrancar EchidnaScratch nos vamos a LEARNING ML, nos vamos a entrenar y que entrene con la clase **APAGA** las palabras que se nos ocurra:

The screenshot shows the EchidnaScratch LearningML interface. At the top, there are three buttons: 'Echidna board connected', 'EchidnaScratch', and 'LearningML' (highlighted with a yellow box and a red circle '1'). Below this is a green header bar with a globe icon, 'Archivo', 'sin nombre', 'Aprende', and a toggle for 'Modo avanzado desactivado'. The main area is divided into three columns: '1. Entrenar', '2. Aprender', and '3. Probar'. In the '1. Entrenar' column, there is a '+ Añadir nueva clase de textos' button (highlighted with a yellow box and a red circle '2'). Below it is a list of text examples: 'Apaga (5)' (highlighted with a yellow box and a red circle '3'), 'Quiero descansar', 'Me voy a dormir', 'Menos luz' (highlighted with a red circle '5'), 'Más oscuridad', and 'Apaga'. At the bottom of the list is a '+ ' button (highlighted with a yellow box and a red circle '4'). In the '2. Aprender' column, there is a 'Lenguaje de los textos' dropdown menu set to 'Español' and a 'Aprender a reconocer textos' button. In the '3. Probar' column, there is an 'Expresión' input field and a 'Comprobar' button.

Y con la clase **ENCIENDE** otras tantas palabras



Pulsamos que entrene (tarda un poco) y cuando nos diga que esta listo para usar el modelo, vamos a comprobarlo, como vemos, la palabra **MUCHA LUZ no aparece en las entrenadas**, pero detecta que tiene más posibilidades que lo que queremos es apagar la luz

A screenshot of a machine learning interface with a green header. The header contains a globe icon, "Archivo", a text input field with "sin nombre", "Aprende", a toggle switch for "Modo avanzado desactivado", and "Acerca de". The interface is divided into three main sections:

- 1. Entrenar**: "Primero necesito algunos textos de ejemplo". It features a button "+ Añadir nueva clase de textos". Below is a list of training examples: "Apaga (5)", "Quiero descansar", "Me voy a dormir", "Menos luz", and "Más oscuridad".
- 2. Aprender**: "Llegó el momento de aprender a clasificar textos". It has a dropdown menu for "Lenguaje de los textos" set to "Español" and a button "Aprender a reconocer textos" with a red circle '1' above it.
- 3. Probar**: "Introduce términos nuevos y comprueba si se clasifican correctamente". It shows an "Expresión" input field with "Mucha luz" and a red circle '2' next to it. Below is a "Comprobar" button with a red circle '3' above it. At the bottom, it says "Creo que pertenece a la clase Apaga, aunque no estoy muy segura" and shows a list of predicted classes with progress bars: "Apaga (53.78 %)" and "Enciende (46.22 %)".

**Acuerdate de** grabarlo lo voy a grabar como enciende-apaga.json, se graba LOS DATOS DE ENTRENAMIENTO luego al volverlo a abrir, tienes que volver a crear el modelo



Archive ▾ enciende-apaga Learn LML ▾ Advanced mode off About

1. New  
Upload from your computer  
Save to your computer

ESTO ES LO QUE SE GRABA

Apaga (5)  
Apaga  
Más oscuridad  
Menos luz  
Me voy a dormir  
Quiero descansar

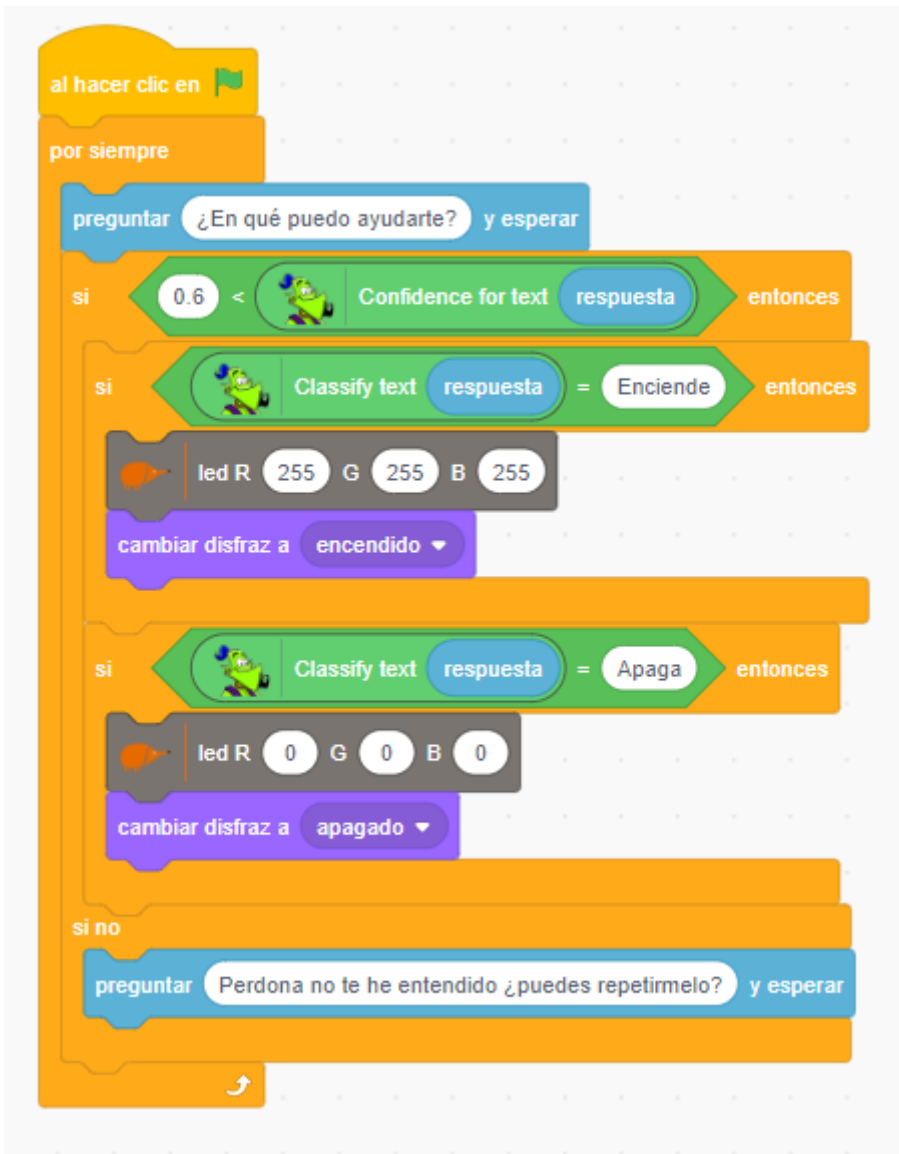
2. Learn  
Now it's time to learn to classify text  
Language of texts Español ▾  
Learning to recognize text

3. Try  
Introduces new terms and checks they are correctly classified  
Expression  
Mucha luz  
Check

Believe belongs to the class Apaga, but I am not sure

- Apaga (50.75 %)
- Enciende (49.25 %)

**Ahora si !!** vamos a EchidnaScratch y creamos este programa usando las instrucciones de Learning ML que se basan en el modelo aprendido anteriormente



Aquí lo tienes todos los procesos en vídeo :

<https://www.youtube.com/embed/do0mDcAuE2w?t=134s>

CC-BY SA Fuente <https://echidna.es/a-programar/echidnaml/como-empezar-con-machine-learning-y-echidna/>

El programa en este repositorio: <https://github.com/JavierQuintana/Echidna>

El sprite lámpara lo he obtenido de <https://scratch.mit.edu/projects/198689957/editor/>

<https://www.youtube.com/embed/XDMYKfE46b8>

## Mejoramos nuestro modelo pero con ayuda de ChatGPT

*Agradecimientos a Fran Perez CP Ana Abarca de Bolea*

Hacemos los mismos pasos, pero con pocos sinónimos, solo dos en cada clase, y lo guardamos. Si nos fijamos en el fichero .json (lo abres con un editor de texto, el bloc de notas por ejemplo) el fichero .json no es más que un fichero que almacena los datos de forma clasificada :

```
{"type":"text","data":{"Enciende":["Poca luz","On"],"Apaga":["Mucha luz","On"]}}
```

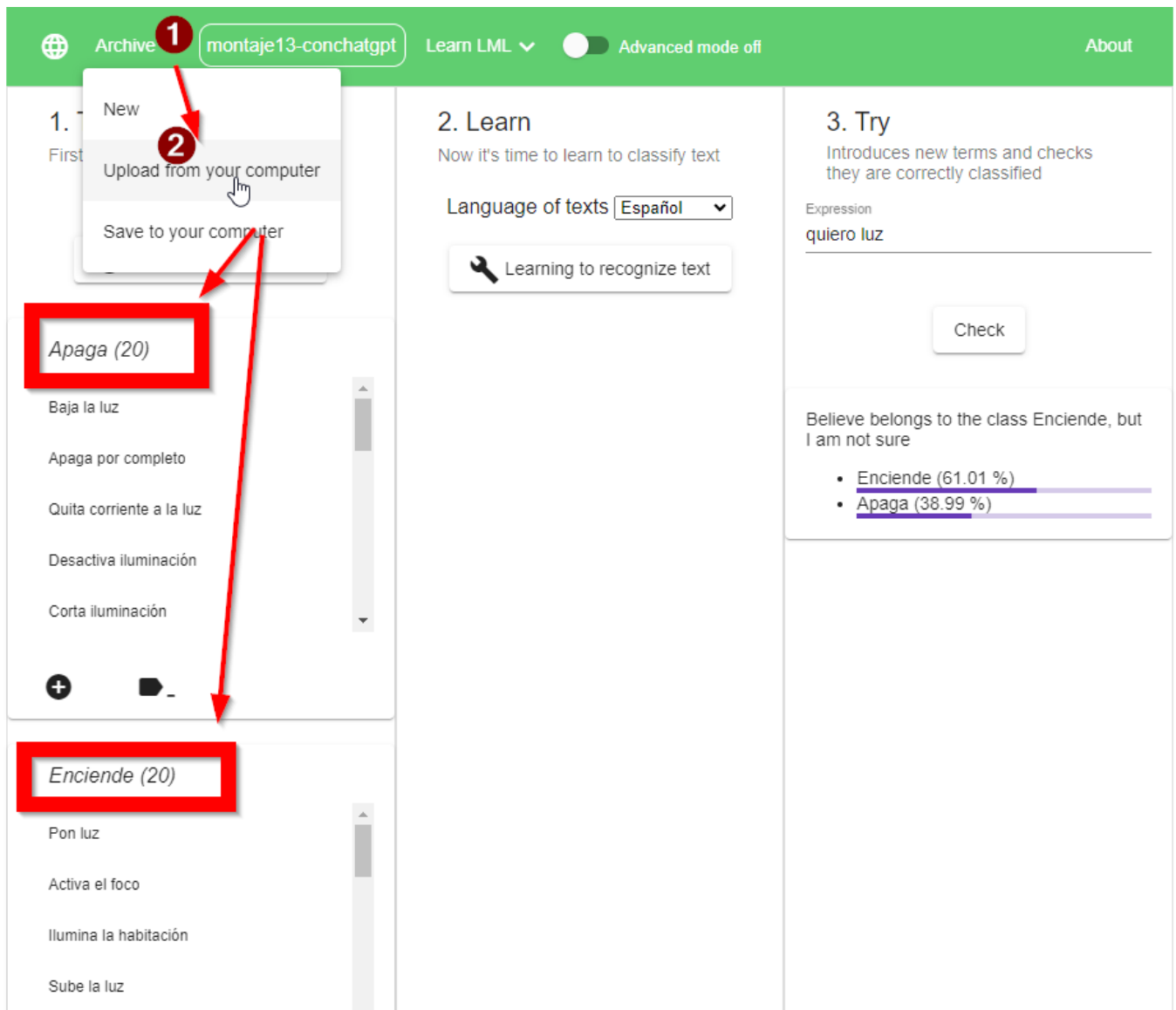
Le decimos a [chatgpt](#) que nos lo enriquezca con este prompt (*u otro similar*) con 20 sinónimos (*y si quieres más, pues más*):

“ quiero que insertes dentro del corchete [ ] de Enciende 20 sinónimos de órdenes para encender la luz y dentro del corchete de Apaga [ ] quiero que pongas 20 sinónimos de órdenes para apagar la luz no pongas saltos de línea, mantén el formato {"type":"text","data":{"Enciende":["Poca luz","On"],"Apaga":["Mucha luz","On"]}}

El chatgpt responde con este código:

```
{"type":"text","data":{"Enciende":["Enciende la luz","Prende la luz","Activa la luz","Ilumina","Dale luz","Pon la luz","Conecta la luz","Inicia la luz","Enciende iluminación","Prende iluminación","Activa iluminación","Enciende lámpara","Prende lámpara","Enciende bombilla","Prende bombilla","Dale corriente a la luz","Sube la luz","Ilumina la habitación","Activa el foco","Pon luz"],"Apaga":["Apaga la luz","Apaga todo","Desconecta la luz","Quita la luz","Apaga lámpara","Apaga bombilla","Corta la luz","Desactiva la luz","Oscurece","Quita iluminación","Apaga iluminación","Cierra la luz","Desconecta iluminación","Apaga el foco","Apaga la lámpara","Corta iluminación","Desactiva iluminación","Quita corriente a la luz","Apaga por completo","Baja la luz"]}}
```

lo metes en el bloc de notas y lo guardamos como *loquequieras.json* y lo importamos en LearningML ;j tenemos ahora 20 textos en cada clase !!! nuestra máquina de IA ha mejorado !!!



Archive **1** montaje13-conchatgpt Learn LML  Advanced mode off About

1. First

New

**2** Upload from your computer

Save to your computer

**Apaga (20)**

Baja la luz

Apaga por completo

Quita corriente a la luz

Desactiva iluminación

Corta iluminación

**Enciende (20)**

Pon luz

Activa el foco

Ilumina la habitación

Sube la luz

2. Learn

Now it's time to learn to classify text

Language of texts

Learning to recognize text

3. Try

Introduces new terms and checks they are correctly classified

Expression

quiero luz

Check

Believe belongs to the class Enciende, but I am not sure

- Enciende (61.01 %)
- Apaga (38.99 %)

## Otro ejemplo pero con una cerveza ;)

<https://libros.catedu.es/books/robotica-educativa-con-mbot/page/maquina-educable-con-machine-learning-for-kids-y-scratch>