

3. Conceptos fundamentales de IA adaptados a Infantil y Primaria

- [Conceptos fundamentales de IA adaptados a Infantil y Primaria](#)
- [Actividad 1. Detectives de reglas: la caja que aprende](#)
- [Actividad 2: El camino de decisiones: construimos un algoritmo \(tipo árbol\) y vemos el sesgo](#)
- [Actividad 3: Laboratorio de sesgos: ¿Es justo nuestro modelo?](#)
- [ANEXO I: La placa Echidna Shield Black y la plataforma EchidnaLM](#)
- [ANEXO II: Diferentes plataformas de Machine Learning](#)

Conceptos fundamentales de IA adaptados a Infantil y Primaria

En este apartado el protagonismo es del **alumnado**: la meta es que niñas y niños puedan **comprender con sentido** (no memorizar) ideas que están detrás de la IA —**algoritmo, datos, modelo y sesgo**— a través de experiencias **concretas, manipulativas y seguras**. Planteamos propuestas donde el alumnado observa qué ocurre cuando sigue una secuencia, clasifica ejemplos, prueba una regla y detecta cuándo esa regla falla. Este enfoque conecta con metodologías activas en las que el aprendizaje se construye a partir de la curiosidad, la exploración, la comparación y la revisión de ideas previas, especialmente pertinentes en edades tempranas (Medina Gaité, 2017).

A la vez, cuidamos una perspectiva inclusiva: la IA y lo “algorítmico” pueden convertirse en una oportunidad para diseñar situaciones donde todo el alumnado participe con apoyos, andamiajes y diferentes formas de demostrar comprensión, reforzando la equidad en el aprendizaje (Vega Maldonado, Vázquez Padilla, & Moscoso Chávez, 2023). ▀

Educación Infantil (3–6): construir los cimientos sin tecnicismos

En Infantil no buscamos explicar “qué es” la IA, sino fortalecer habilidades que la sostienen: **secuenciación, causa-efecto, clasificación, reconocimiento de patrones, atención** y una relación sana con el error. Las experiencias deben ser **breves, lúdicas y tangibles**, respetando ritmos y posibilidades de acción del grupo (Medina Gaité, 2017). ▀

Aquí los conceptos se viven así (sin necesidad de nombrarlos siempre):

- **Algoritmo** = “pasos” para lograr algo: recorridos en el suelo, “robot humano”, recetas visuales con tarjetas de acciones.
- **Datos** = “ejemplos”: objetos reales (tapones, figuras, imágenes) para clasificar por un criterio (color, tamaño, forma).
- **Modelo** = “la regla” que construimos: “si tiene... entonces va aquí”.

- **Sesgo (semilla)** = cuando la regla falla porque faltan ejemplos o están desequilibrados: si solo clasificamos objetos rojos, después no sabemos qué hacer con otros colores.

El valor educativo está en el lenguaje que acompaña y da seguridad: “¿Qué hicimos primero?” “¿Qué cambió?” “¿Qué necesitamos para que funcione con todas las tarjetas?”; así el error se convierte en una oportunidad para ajustar y aprender (Medina Gaite, 2017).

Educación Primaria (6–12): poner nombre, probar y pensar críticamente

En Primaria el alumnado ya puede **nombrar** los conceptos y, sobre todo, **justificarlos con evidencias**: comparar resultados entre grupos, registrar aciertos/errores y explicar por qué una regla “generaliza” bien o mal. Propondremos experiencias donde el alumnado entienda que:

- un **algoritmo** es una secuencia clara (y depurable) de pasos;
- los **datos** son ejemplos que “alimentan” una decisión;
- un **modelo** es la regla que “aprende” a partir de esos ejemplos;
- y el **sesgo** aparece cuando los datos no representan bien la realidad o cuando la regla se construye con un conjunto limitado de casos.

Esta aproximación es coherente con la evidencia de que las herramientas basadas en IA pueden complementar métodos tradicionales y favorecer aprendizajes (especialmente si se integran con estrategias pedagógicas adaptadas), pero que el impacto educativo depende del diseño didáctico y del contexto (Mayorga Villegas et al., 2025).

En términos prácticos, trabajaremos con recursos manipulativos (tarjetas, cajas de clasificación, “casos sorpresa”) y pequeñas rutinas de pensamiento: “¿con qué ejemplos entrenamos?” “¿qué ejemplo faltaba?” “¿a quién deja fuera esta regla?”; así el alumnado aprende a detectar limitaciones y a proponer mejoras, un paso clave hacia un uso crítico y responsable (Vega Maldonado et al., 2023).

Actividad 1. Detectives de reglas: la caja que aprende

En Infantil no buscamos “explicar IA”, sino ayudar a que el alumnado construya **cimientos cognitivos y emocionales**: secuenciar, clasificar, detectar patrones, anticipar consecuencias y vivir el error con seguridad. Este enfoque encaja con metodologías activas donde la curiosidad y la exploración guían el aprendizaje, y el error se entiende como parte natural del proceso.

Datos de la actividad

- **Etapas:** Educación Infantil (3-6 años)
- **Duración:** 35-45 minutos (recomendable repetir en 2 sesiones cortas)
- **Agrupación:** pequeños grupos (4-6) + asamblea final
- **Espacio:** Rincones
- **Formato:** manipulativo y desenchufado (sin pantallas)

Objetivo general

Que el alumnado experimente, de forma lúdica y concreta, **el alumnado “enseña” a una caja** a decidir dónde van las cosas. La caja no piensa: **solo sigue una regla**. El alumnado descubre que:

- seguir **pasos claros** (secuencia) ayuda a que algo “funcione” (base de *algoritmo*).
- los **ejemplos** importan para decidir (base de *datos*), si usamos **muchos ejemplos** aprende mejor.
- la **regla** que inventamos puede funcionar o fallar (base de *modelo*),
- y que si solo usamos algunos ejemplos, la regla se vuelve “injusta” o se equivoca (sesgo, sin decir la palabra)

Materiales

- 2-3 cajas o aros en el suelo.

- Pegatinas o pictos para etiquetar cajas (por ejemplo: □ “sí”, □ “no”; o “va aquí / no va aquí”).
- Tarjetas/objetos variados (20–30): animales, alimentos, juguetes, prendas, formas, colores...
- Un “objeto sorpresa” (no previsto) para provocar reflexión (p. ej., murciélago si el criterio era “vuela = pájaro”).
- Opcional DUA: tarjetas con pictos grandes, objetos reales (tamaño manipulable), panel visual de “pasos”.

Desarrollo de la actividad

1) Asamblea de arranque (5–7 min): “¿Cómo decide la caja?”

Presentamos una caja como si fuera una “máquina” simpática: **no piensa sola**, decide según una regla que le damos.

Preguntas guía (lenguaje infantil):

- “¿Qué necesita la caja para saber dónde va cada cosa?”
- “¿Y si nos equivocamos... qué podemos hacer?”

Creemos clima seguro para probar y corregir; favorecer que expresen ideas sin miedo al error, como recomienda el trabajo por proyectos en Infantil..

2) Juego 1 (10–12 min): “Pasos claros” (base de **algoritmo**)

En el suelo colocamos una **ruta de 3 pasos** con pictos grandes:

1. “Miro la tarjeta/objeto”
2. “Busco la pista” (color/forma/parte del cuerpo/ruedas...)
3. “Lo pongo en la caja que toca”

Objetivo: que el alumnado vea que **si sigo una secuencia**, me equivoco menos.

El alumnado practica en turnos: cada niña/niño sigue los pasos para colocar un objeto.

Lo importante: si hay dudas, no damos la respuesta; preguntamos:

- “¿En qué paso estás?”
- “¿Qué pista has visto?”

Trabajamos secuenciación y autorregulación: “sigo pasos”, “me paro y reviso”.

3) Juego 2 (12–15 min): “Ejemplos para descubrir la regla” (base de **datos y modelo**)

Ahora el reto cambia: el grupo debe **adivinar la regla** de la caja solo mirando ejemplos.

Objetivo: que el alumnado descubra que una “máquina” necesita **ejemplos** para aprender.

- El o la docente (o un grupo) clasifica **5 objetos** en silencio (ejemplos).
- El resto observa y propone: “Creo que la regla es...”
- Probamos con 3 objetos nuevos.

Aquí aparece de forma natural la idea de **modelo = regla**: “la caja hace esto con lo que hemos visto”.

4) Momento clave (3–5 min): “El objeto sorpresa” (semilla de **sesgo**)

Introducimos un objeto que **rompe la regla** o la deja en evidencia (p. ej., si la regla parecía “los que vuelan van aquí”, metemos un avión o un murciélago).

Objetivo: que vean que la regla puede fallar si los ejemplos no eran variados.

Preguntas guía:

- “¿Qué ha pasado?”
- “¿La regla sirve siempre?”
- “¿Qué ejemplos nos faltaban para no equivocarnos?”

Aquí sembramos la idea de **sesgo** con lenguaje accesible: *si la caja solo vio algunos ejemplos, luego se confunde con otros diferentes.*

5) Cierre en asamblea (5–7 min): “Hoy hemos entrenado a la caja”

Construimos un mini-resumen oral (con apoyo visual si se desea):

- “Hemos seguido **pasos**.”
- “Hemos usado **ejemplos**.”
- “Hemos inventado una **regla**.”
- “Y cuando faltaban ejemplos, la regla **fallaba**.”

Este cierre refuerza metacognición y lenguaje, y normaliza el error como parte del aprendizaje, tal como se destaca en propuestas activas en Infantil.

Accesibilidad e inclusión (DUA aplicado)

Para que participe **todo el alumnado** con seguridad:

- **Representación:** usar objetos reales + pictos grandes; apoyos visuales de pasos.
- **Acción/expresión:** permitir responder señalando, moviendo el objeto o explicando oralmente; turnos flexibles.
- **Implicación:** ofrecer roles rotatorios (“quien observa”, “quien coloca”, “quien comprueba”), y opción de reto (“inventa una regla nueva”).

La meta es reducir barreras y aumentar oportunidades de participación, coherente con una mirada inclusiva de la innovación educativa.

Evaluación

Checklist rápido (docente):

- Sigue una secuencia de pasos con ayuda.
- Justifica con una pista (“porque es rojo”, “porque tiene ruedas”).
- Acepta corregir y volver a probar sin frustración excesiva.
- Propone cambios cuando la regla falla (“necesitamos más ejemplos”, “la regla no vale para todos”).



Actividad 2: El camino de decisiones: construimos un algoritmo (tipo árbol) y vemos el sesgo

Actividad 2. El camino de decisiones: construimos un algoritmo tipo árbol.

El alumnado entiende cómo “decide” una IA mediante un algoritmo en forma de preguntas (árbol de decisiones), usando objetos reales. Comprende que el modelo es el árbol final construido con datos (ejemplos) y que aparece sesgo si los ejemplos no representan bien la realidad. Es muy manipulativa y favorece el razonamiento.

Datos de la actividad

- Curso: 3º Primaria (8-9 años)
- Tiempo: 45-60 min
- Agrupación: equipos de 4-5 y una puesta en común final.
- Espacio: aula (suelo o mesas grandes para montar el “camino”)
- Formato: manipulativa y sin pantalla.

Objetivo general

Construir un árbol de decisiones para clasificar objetos y comprender el proceso de aprendizaje seguido por **datos, algoritmo, modelo y sesgo** influyen en los aciertos y errores.



Materiales

- 25-35 objetos o [tarjetas](#) (material escolar variado).
- Cartulinas o post-its grandes para escribir preguntas SÍ/NO.
- Cinta adhesiva para marcar el “camino” en el suelo.
- 3 categorías (carteles): por ejemplo, ESCRIBIR / CORTAR (tarjeta) /SUJETAR.
- Hoja de registro con 10 pruebas (correctas/incorrectas) + breve causa de error (¿Por qué falló?).

Desarrollo paso a paso

1. Pregunta detonante (3 min)

¿Podemos construir un camino de preguntas SÍ/NO para que una “máquina” clasifique objetos?

2. Organización y roles (3 min)

- **Constructor/a:** coloca preguntas y flechas.
- **Probador/a:** pasa objetos por el camino.
- **Detective de errores:** registra fallos y causas.
- **Portavoz:** explica el modelo al final.

3. Datos iniciales (8-10 min)

El docente entrega al equipo un set de **12 objetos** ya “etiquetados” (con pegatina o tarjeta) en una de las 3 categorías elegidas.

Ejemplo (si usamos función):

- ESCRIBIR: lápiz, boli, rotulador, tiza
- CORTAR: tijeras (tarjeta), cúter de cartón (solo imagen), sacapuntas (a debate)
- SUJETAR: clip, pinza, goma elástica, cinta, grapa (tarjeta)

Aquí se introduce: **DATOS = ejemplos con etiqueta (la respuesta correcta).**



4. Construcción del algoritmo (15-20 min)

El equipo crea un **camino de preguntas SÍ/NO** para llegar a una categoría.

Ejemplos de preguntas útiles:

- “¿Sirve para hacer marcas o letras?”
- “¿Tiene filo o corta?” (si se usan tarjetas, no objetos peligrosos)
- “¿Sirve para sujetar/juntar cosas?”
- “¿Se usa con las manos para apretar?”

Reglas del juego (para que sea claro):

- Máximo **3 preguntas** antes de llegar a un destino
- Cada pregunta debe poder responderse **SÍ o NO**
- Cada objeto debe poder “viajar” por el camino

Aquí se nombra: **ALGORITMO = secuencia de preguntas para decidir.**

5. El modelo (2 min)

Cuando el camino funciona con los 12 ejemplos, se pega un cartel:

MODELO = nuestro árbol final de preguntas (lo aprendido con los datos).

6. Reto de comprobación: objetos nuevos (10 min)

El docente da **10 objetos nuevos** (no estaban en los 12 ejemplos) y el equipo los pasa por el modelo.

Registran ☐ / ☐ .

Ejemplos de objetos “nuevos” que provocan pensamiento:

- regla (¿es para medir o para trazar?)
- pegamento (¿juntar = sujetar?)
- estuche (¿sirve para...?)
- goma de borrar (¿es escribir o corregir?)
- botella (ninguna categoría: obliga a revisar)

7. Sesgo: faltaban ejemplos (8-10 min)

Ahora el docente revela una condición:

“Vuestros **datos** tenían muchos ejemplos de ESCRIBIR y muy pocos de SUJETAR (o faltaba ‘pegamento’, ‘goma’, etc.).”

Se repite una mini-prueba con 4 objetos “conflictivos”.

El equipo identifica:

- ¿Qué tipo de ejemplo faltó?
- ¿Qué pregunta está mal o es ambigua?

Aquí se nombra: **SESGO** = **cuando el conjunto de ejemplos no representa bien y el modelo aprende una idea incompleta.**

8. Mejora del modelo (5-7 min)

Cada equipo aplica 1 mejora:

- Añadir 2-3 ejemplos nuevos (datos) y rehacer una pregunta
- Cambiar el orden de preguntas
- Crear una “salida extra”: **OTROS / NO CLASIFICA** (muy realista)

Vuelven a probar con los 4 objetos conflictivos.

9. Cierre y puesta en común (5 min)

- “¿Qué ayudó más: cambiar preguntas o añadir ejemplos?”
- “¿Puede un modelo ser ‘injusto’ si solo ve algunos casos?”
- “¿Qué harías para que el modelo fuese más fiable?”

DUA: adaptaciones para inclusión

- **Representación:** preguntas con pictos; demostración breve inicial.
- **Acción/expresión:** mover objetos por el camino; registro con pegatinas.
- **Implicación:** el equipo elige preguntas y puede crear categoría “OTROS”.

Semáforo por equipo

- ☐ “Nuestro modelo clasifica bien y sabemos explicar por qué fallaba.”



- ☐☐ “Clasifica a veces: tuvimos que mejorar preguntas o datos.”
- ☐☐ “Nos costó: necesitamos más ejemplos claros.”

Checklist docente

- Identifica datos como ejemplos etiquetados.
- Explica algoritmo como preguntas/pasos.
- Define modelo como el árbol final.
- Reconoce sesgo como falta de variedad en los datos.

Glosario rápido para explicarlo en clase

- **Datos:** ejemplos con etiqueta (respuesta correcta).
- **Algoritmo:** preguntas SÍ/NO que guían la decisión.
- **Modelo:** el árbol final que usamos para clasificar.
- **Sesgo:** cuando el árbol falla más con ciertos objetos porque faltaban ejemplos.
- **Prueba/validación:** comprobar el modelo con casos nuevos.

Actividad 3: Laboratorio de sesgos: ¿Es justo nuestro modelo?

Actividad 3. ¡Crea una IA justa! El laboratorio de datos y sesgos

Justificación

Esta actividad hace visible que “aprender” en IA significa **crear un modelo** a partir de **datos** y luego **probarlo** con casos nuevos. El alumnado detecta **sesgos** cuando ciertos grupos quedan poco representados y entiende por qué eso puede ser injusto. Esta actividad Integra pensamiento crítico, convivencia y lenguaje de IA sin usar pantallas.

Datos de la actividad

- **Curso:** 5º Primaria (10-11 años)
- **Tiempo:** 55-70 min (RECOMENDADO EN 2 SESIONES)
- **Agrupación:** equipos de 4-5 + puesta en común
- **Espacio:** aula (mesas)

Objetivo general

Construir y evaluar un modelo sencillo (reglas/puntuación) para tomar una decisión, usando datos de ejemplo, y comprobar cómo aparece el **sesgo** por representación y por “reglas mal diseñadas”.

Materiales



- 30 “fichas de caso” ([tarjetas](#)) con Robots y rasgos neutros del aula (sin datos sensibles):
 - Ej.: “Trae deberes 4/5 días”, “Ayuda a un compañero”, “Se distrae a veces”, “Llega puntual”, “Interrumpe”, “Trabaja en equipo”...
- 3 sobres: **ENTRENAMIENTO (datos)** / **PRUEBA** / **DIFÍCILES**
- Plantilla “[Nuestro modelo](#)” (tabla de puntos o reglas)
- Hoja de registro: 12 predicciones □ /□
- Post-its: **DATOS** / **ALGORITMO** / **MODELO** / **SESGO** / **PRUEBA**

Nota: Los “personajes” pueden ser animales o robots para hacerlo más lúdico.

Desarrollo de la actividad

1) Pregunta detonante (4 min)

“Si quisiéramos una ‘máquina’ que eligiera a quién elegir como el **Robot ayudante del aula...** ¿cómo le enseñaríamos a decidir?”

2) Organización (3 min)

[Roles:](#)

- **Diseñador/a del modelo** (escribe reglas/puntos)
- **Entrenador/a de datos** (analiza ejemplos)
- **Probador/a** (aplica el modelo a nuevos casos)
- **Auditor/a de sesgo** (revisa a quién perjudica y por qué)
- (Opcional) **Portavoz**

3) Datos (entrenamiento) (10–12 min)

Cada equipo recibe 10 tarjetas del sobre **ENTRENAMIENTO**, ya etiquetadas por el docente:

- “Sí es elegido”
- “NO es elegido”

El equipo analiza:

- ¿Qué rasgos aparecen mucho en los “Sí”?
- ¿Qué rasgos aparecen mucho en los “NO”?



Se nombra: **DATOS** = ejemplos con respuesta correcta.

4) Construcción del algoritmo (modelo por puntos) (12–15 min)

El equipo crea un **algoritmo** tipo “puntos” (muy manipulativo y fácil de aplicar):

- +2 si “ayuda a compañeros”
- +2 si “respeta turnos”
- +1 si “trae material”
- –2 si “interrumpe”
- –1 si “llega tarde a veces”

Regla final: si suma ≥ 3 puntos $\rightarrow$ **SÍ**, si no $\rightarrow$ **NO**.

Se nombra: **ALGORITMO** = las reglas/pasos para sumar y decidir y **MODELO** = la tabla final de puntos (lo aprendido con los datos).

5) Prueba (validación) con datos nuevos (10–12 min)

Se entrega el sobre **PRUEBA** con 8 tarjetas nuevas (sin etiqueta).

El equipo aplica su modelo, predice SÍ/NO y registra.

Después el docente revela la etiqueta real (o una hoja de soluciones común).

6) Mini-laboratorio de sesgos (12–15 min)

Ahora entra el sobre **DIFÍCILES** con 4 tarjetas pensadas para mostrar sesgo. Ejemplos:

- Casos con rasgos “mixtos” que no aparecían en entrenamiento (“ayuda mucho pero se pone nervioso y habla rápido”)
- Casos con un rasgo que el modelo penaliza demasiado (“llega tarde a veces” por motivo ajeno, sin entrar en realidades personales)

El auditor responde en la hoja:

- ¿Qué tipo de caso recibe más “NO”?
- ¿Ese rasgo estaba poco representado en los datos?
- ¿La regla está exagerando una pista?



Se nombra: **SESGO** = cuando el modelo se equivoca más con ciertos casos porque los datos eran pobres o las reglas no eran justas.

7) Mejora del modelo (6–8 min)

Cada equipo elige UNA mejora (obligatorio justificarla):

- **Mejora de datos:** añadir 2 tarjetas de entrenamiento que faltaban (más variedad de casos)
- **Mejora de reglas:** bajar un castigo ($-2 \rightarrow -1$), subir un criterio positivo, o añadir “NO SÉ” si está en zona gris
- **Mejora de prueba:** crear una regla de “revisión humana” para casos límite

Se repiten las 4 tarjetas DIFÍCILES y se comparan resultados.

8) Debate final (5–7 min)

- “¿Qué mejoró más: cambiar datos o cambiar reglas?”
- “¿Cuándo sería peligroso usar este modelo sin revisar?”
- “¿Qué significa que un modelo sea ‘justo’?”

DUA (adaptaciones)

- **Representación:** tarjetas con iconos (mano ayudando, reloj puntual, boca hablando), lectura compartida.
- **Acción/expresión:** cálculo en grupo con fichas/puntos, explicación oral o con esquema.
- **Implicación:** rol de auditor (misión), debate con ejemplos concretos, opción “NO SÉ”.

Evaluación final

Semáforo por equipo

- ☐ “Definimos datos/algoritmo/modelo/sesgo y mejoramos el modelo con una razón.”
- ☐ “Lo entendemos pero nos cuesta explicar el sesgo o justificar la mejora.”
- ☐ “Necesitamos apoyo para construir reglas y ver el sesgo.”

Checklist docente (5 ítems)



- Identifica datos como ejemplos etiquetados.
- Describe algoritmo como reglas/pasos.
- Explica modelo como la tabla final.
- Detecta sesgo (a quién perjudica y por qué).
- Propone una mejora razonada (datos o reglas).

ANEXO I: 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.

ANEXO II: Diferentes plataformas de Machine Learning

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

	Learning ML (Echidna en local)	Learningml 2.0	Teachable machine
Uso	Local	https://learningml.org/lml-editor/	https://teachablemachine.withgoogle.com/
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.