



TRABAJO FIN DE MÁSTER

Juego de Robótica para la ESO con placa Echidna

Estudiante: Borja Hernández López

Especialidad: Tecnología y Digitalización

Tutor: Juan Ramón Fernández González

Curso académico: 2023-24









RESUMEN

Con el presente Trabajo Final de Máster, se pretende abordar un proyecto para alumnos de Tecnología y Digitalización de 4º curso de Educación Secundaria Obligatoria (ESO), en el que se den las pautas para desarrollar un juego de robótica para la placa de desarrollo Echidna Black, con una marioneta accionada por servos que reacciona a imágenes usando *machine learning* ante el verdadero o falso de la pregunta realizada previa programación en entorno scratch.echidna.

Dicho proyecto se enmarca en la iniciativa de Echidna Educación, la cual promueve la programación de sistemas físicos en entorno visual para potenciar la competencia STEM (Matemáticas, Ciencia, Tecnología y Arte). A través de este trabajo, se busca no solo enseñar a los estudiantes conceptos tecnológicos avanzados, sino también promover competencias clave, como la resolución de problemas, la creatividad y el pensamiento computacional. La integración de la placa Echidna y Scratch en el aula enriquece el proceso de enseñanza-aprendizaje, preparando a los alumnos para enfrentar los desafíos tecnológicos del futuro, fomentando un enfoque interdisciplinario en la educación STEM.

ABSTRACT

The aim of this Master's Final Project is to tackle a project for students of Technology and Digitalisation in the 4th year of Compulsory Secondary Education (ESO), in which guidelines are given for developing a robotics game for the Echidna Black development board, with a servo-driven puppet that reacts to images using machine learning when faced with a true or false question asked after programming in the scratch.echidna environment.

This project is part of the Echidna Education initiative, which promotes the programming of physical systems in a visual environment to enhance the STEM (Mathematics, Science, Technology and Art) competence. Through this work, the aim is not only to teach students advanced technological concepts, but also to promote key skills such as problem solving, creativity and computational thinking. The integration of the Echidna board and Scratch in the classroom enriches the teaching-learning process, preparing students to face the technological challenges of the future, fostering an interdisciplinary approach to STEM education.

PALABRAS CLAVE

Machine learning; TIC; Interface; Modulación PWM; Servomotor; Robótica; Código; Placa Echidna; Robótica; ESO.





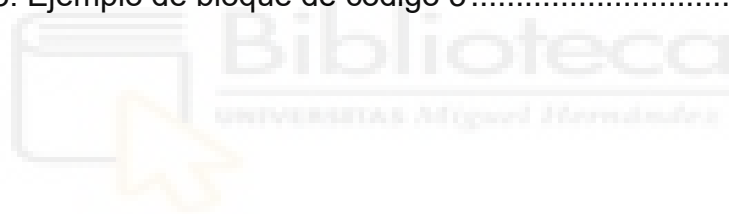
ÍNDICE DE CONTENIDO

1.	INTRODUCCIÓN.....	- 3 -
1.1.	Justificación y objetivos.....	- 3 -
1.2.	Marco teórico y estado del arte	- 4 -
2.	METODOLOGÍA.....	- 7 -
3.	ANTECEDENTES	- 9 -
4.	RESULTADOS	- 13 -
4.1.	Proyecto Echidna Educación	- 13 -
4.2.	Placa de desarrollo Echidna.....	- 14 -
4.3.	Conexión con el PC y configuración.....	- 18 -
4.4.	Programación de la marioneta	- 20 -
5.	DISCUSIÓN Y CONCLUSIONES.....	- 27 -
5.1.	Recomendaciones y líneas futuras	- 28 -
6.	CONTRIBUCIONES PRÁCTICAS.....	- 31 -
7.	REFERENCIAS, BIBLIOGRAFÍA y enlaces webs.....	- 33 -
7.1.	REFERENCIAS Y BIBLIOGRAFÍA.....	- 33 -
7.2.	Enlaces a webs de consulta	- 34 -
8.	Anexos	- 37 -
8.1.	Situación de aprendizaje	- 37 -



ÍNDICE DE ILUSTRACIONES

Ilustración 1. Placa de desarrollo Echidna Shield.....	- 16 -
Ilustración 2. Placa de desarrollo Echidna Black.....	- 17 -
Ilustración 3. Puerto USB de la placa Echidna Black	- 18 -
Ilustración 4. Vista gráfica del funcionamiento entre la placa Echidna y el IDE Echidna.scratch.....	- 18 -
Ilustración 5. Fallo de conexión entre scratch.echidna.es y la placa Echidna Black	- 19 -
Ilustración 6. IDE de scratch.echidna	- 19 -
Ilustración 7. Prueba de funcionamiento del entrenamiento de las variables de machine learning creadas para nuestro código.....	- 21 -
Ilustración 8. Ejemplo de bloque de código 1	- 22 -
Ilustración 9. Ejemplo de bloque de código 2.....	- 22 -
Ilustración 10. Subprograma "Aplauso".....	- 23 -
Ilustración 11. Subprograma "Mal"	- 23 -
Ilustración 12. A la izquierda los disfraces del programa principal. A la derecha los disfraces del subprograma Aplauso con las distintas imágenes o Disfraces para simular movimiento en el aplauso.....	- 24 -
Ilustración 13. Ejemplo de bloque de código 3.....	- 25 -







1. INTRODUCCIÓN

Hoy en día la interacción entre los distintos campos tecnológicos es más intensa que nunca, creciendo diariamente y haciéndose patente en la vida cotidiana, por lo que a la hora de afrontar un proyecto es notable la cantidad de recursos necesarios de distintos campos de estudio y por supuesto, no únicamente tecnológicos.

Teniendo esto en cuenta es muy importante inculcar en el alumnado unas buenas bases y prácticas a la hora de afrontar un trabajo o proyecto, sea o no tecnológico, potenciando su conocimiento y creatividad en el ámbito de las TICs, siempre desde un punto de vista crítico.

En base a esto se pretende abordar este TFM con la intención de que sirva de ayuda al lector, sea educador o educando para introducirle o mejorar, en caso de que sea conocedor de la programación en entorno visual para placas de desarrollo pensadas en el ámbito educativo; siempre desde el punto de vista de la relación continua docente-discente, utilizando una metodología constructivista basada en el proceso de enseñanza-aprendizaje a la vez que se atienden la diversidad del alumnado y las NEE de quien lo requiera.

1.1. Justificación y objetivos

El presente Trabajo Final de Máster, tiene como objetivo principal, hacer un estudio de algunas de las herramientas de programación disponibles para la placa de desarrollo Echidna, haciendo un repaso previo de los antecedentes en los lenguajes de programación gráfica, excepto algún caso en el que se amplíe información y/o algunas otras placas de desarrollo disponibles y programables en los mismos lenguajes.

Además, se quiere instruir o adentrar al educador y al educando que desconozca esta parte de la tecnología, en el uso de las TICs aplicadas y enfocadas a la asignatura de Tecnología y Digitalización para cuarto curso de ESO.

En particular para el caso del alumno, se pretende dar a conocer las herramientas y conocimientos necesarios en la programación en entorno gráfico enfocado al desarrollo y uso de herramientas de *machine learning* para poder garantizar la obtención de muchas competencias clave, como la competencia personal, social y de aprender a aprender, la competencia digital y principalmente la competencia matemática y en ciencia, tecnología e ingeniería (en adelante STEM), ya que abordar un proyecto de estas características, supondrá un esfuerzo bastante alto por parte de alumno para poder alcanzar el objetivo final y para el profesorado a la hora de guiar a los alumnos.

Atendiendo a lo anterior, debemos tener en cuenta que el proceso de enseñanza-aprendizaje en el ámbito escolar se basa en una relación continua docente-discente que, generalmente, se desenvuelve en una situación de grupo. Para que este proceso sea eficaz precisa de una serie de métodos y estrategias de enseñanza que posibiliten la comunicación e interacción entre estos, favoreciendo así la transmisión de conocimientos, pero dentro de esta situación de colectivización de la enseñanza observamos también que cada alumno/a tiene su propio proceso de aprendizaje y sobre el cual la acción del docente ha tenido como objetivo facilitararlo, atendiendo para ello a los principios de individualización y diversidad de forma que esos aprendizajes incidan positivamente en dicho proceso.

Por ende, la metodología a utilizar estará basada en los principios de actividad e individualización, contribuyendo así a una continua interacción docente-discente que de sentido al principio de diversidad. Esta combinación entre el principio de educación común y una educación personalizada que responda a los intereses del alumnado es lo que nos va a permitir establecer las medidas necesarias tanto a nivel organizativo como curricular que resulten más adecuadas a las características de los alumnos/as.

Estos principios básicos que definen la concepción constructivista del proceso de enseñanza-aprendizaje en referencia al tratamiento educativo de las diferencias individuales, es el mismo que propone hacer frente a la diversidad mediante la aplicación de medidas curriculares que favorezcan la utilización de métodos de enseñanzas diferentes en función de las necesidades educativas y características individuales del alumno.

El alumnado que durante su etapa en ESO y bachiller conozca y aprenda estas herramientas y desee cursar estudios superiores relacionados con la tecnología o ingeniería, tendrá unas bases tanto de procedimiento a la hora de afrontar un proyecto como habilidades suficientes para desarrollarlo, finalizarlo, exponerlo y defenderlo.

1.2. Marco teórico y estado del arte

Actualmente existen gran cantidad de lenguajes de programación y placas de desarrollo, con lenguaje propio o a través de lenguajes ya existentes, pero esta tecnología no es nueva de este siglo. Se podría decir que su inicio se remonta a las décadas de 1960 y 1970, aunque la programación en computadoras era principalmente el dominio de profesionales y expertos.

Sin embargo, algunos lenguajes de programación, como LOGO, desarrollado por Seymour Papert y su equipo se introdujeron en entornos educativos para enseñar conceptos de programación a estudiantes jóvenes. LOGO, por ejemplo,

presentaba un entorno gráfico donde los estudiantes podían controlar un "turtle" (tortuga) para dibujar formas y patrones.

Con el paso de los años y a medida que la tecnología avanzaba, surgieron lenguajes de programación visuales diseñados específicamente para el entorno educativo o cualquier persona que quisiera adentrarse en el mundo de la programación, como por ejemplo HyperCard, creado por Apple en 1987, permitía a los usuarios crear "pilas" de tarjetas con elementos visuales y enlaces de hipertexto, lo que facilitaba la creación de aplicaciones interactivas.

Con el tiempo, surgieron herramientas de programación diseñadas específicamente para enseñar a los niños a programar. Un ejemplo temprano es LEGO Mindstorms, desarrollado por la empresa de juguetes Lego, que introdujo a los niños en la robótica y la programación a través de la construcción y programación de robots LEGO.

Hoy en día existe gran cantidad software y entornos de programación visual dirigidos al sector educativo que fomentan el aprendizaje de todo el estudiantado, desde niños hasta adultos. así como al profesorado, siendo Scratch uno de los programas más extendidos y versátiles tanto en su versión local como online, desarrollado por el Grupo Lifelong Kindergarten del MIT Media Lab, lanzado por primera vez en el año 2007.

Scratch es una plataforma o entorno de programación visual que permite a los usuarios crear proyectos interactivos, historias y juegos al ensamblar bloques de código simplificando la sintaxis de programación al eliminar la necesidad de escribir código, lo que la hace accesible para niños y principiantes.

Además, es posible desarrollar código para algunas placas de desarrollo, lo que facilita a estudiantes que están en etapas de ESO y bachiller la programación de sensores, luces, altavoces y multitud de actuadores de todo tipo, sin la complejidad de tener que conectarlos desarrollando la circuitería necesaria para hacerlos funcionar, lo que aventaja en gran medida el aprendizaje de sistemas más complejos. Un ejemplo de estas placas de desarrollo es Echidna.

La placa Echidna programada con Scratch funciona como una interfaz entre el entorno de programación visual de Scratch y los componentes físicos conectados a la placa, proporcionando un entorno de programación visual basado en bloques, que con la extensión Echidna incorporada, se agregan bloques específicos que te permiten interactuar con los sensores y actuadores conectados a la placa, como son mover un motor, encender un LED, esperar la detección de un sensor o tomar lecturas, etc.

En resumen, la placa Echidna programada con Scratch actúa de enlace entre el entorno de programación visual de Scratch y el mundo físico, permitiéndote crear proyectos interactivos que respondan a los datos de los sensores y controlen los componentes electrónicos en tiempo real, ya sea dando un uso de movimiento a

una marioneta, un juego de luces o cualquier otro programa que fomente una educación basada en el constructivismo.



2. METODOLOGÍA

A la hora de afrontar un proyecto de este tipo, es muy conveniente tener clara la finalidad del mismo y hasta donde se pretende llegar y desarrollar, de manera que no se alargue en el tiempo de manera indefinida.

Atendiendo al resumen del presente documento y dado que es para poder ser desarrollado por alumnos de cuarto de ESO enfocado a la asignatura de Tecnología y Digitalización, se acotará el proyecto en las siguientes fases generales:

Estudio previo y estado del arte. Primeramente, se hará un estudio de antecedentes en los lenguajes de programación, así como un estudio de las distintas placas de desarrollo, repasando sus ventajas e inconvenientes. Esta será la parte principal del proyecto, pues, se investigarán los antecedentes a las placas y lenguajes actuales y su evolución histórica hasta la situación actual, enfocado siempre al marco educativo.

Construcción de la marioneta. Esta debe ser de manera sencilla, sin grandes trabajos de carpintería, ebanistería o tratado de madera, pues no es este el principal objetivo, por consiguiente, con una maqueta humanoide sin mucho nivel de detalle será suficiente. Es posible realizarla con materiales cotidianos y ligeros como puedan ser tubos de cartón del papel de cocina pegados con celo o similar.

Programación en Scratch. Para programación del código fuente, lo haremos basándonos en el principio de que la respuesta debe ser booleana o de una, pudiendo ser un ejemplo de respuestas el mover los brazos arriba y abajo como una respuesta correcta y de manera contraria, dejarlos levantados o fijos como resultado de una respuesta incorrecta, también estando fijos hacia arriba para verdadero o falsos hacia abajo, etc. Esto se definirá más adelante y se detallará con la programación.

Para la programación de la placa existe la posibilidad de hacerlo mediante lenguaje escrito o gráfico, aunque se hará usando lenguaje gráfico en el entorno de desarrollo scratch.echidna, pues es más comprensible e intuitivo.

Para introducirnos un poco en materia, veremos a continuación las principales características de algunos de los lenguajes gráficos y escritos mediante código fuente.

- Lenguajes gráficos que pueden ser mediante símbolos o gráficos, como Scratch, Snap4Arduino, Mblock o S4A
 - o Ventajas. Mucho más sencillo e intuitivo, ideal para principiantes en programación como es nuestro caso.

- Inconvenientes. No se llega a aprovechar todo el potencial de la placa de desarrollo, pues a este nivel, no es posible tener acceso al control de determinadas instrucciones y herramientas de programación.
- Lenguaje escrito o código fuente, como puede ser el caso de ArduinoIDE, que está basado en lenguaje C, pero también se pueden usar otros lenguajes como Phyton, que para determinados proyectos tienen librerías con herramientas específicas más desarrolladas. En este tipo de proyectos educativos, si se hace uso de código, será ArduinoIDE y en la medida de lo posible, de ejemplos genéricos proporcionados por la empresa Arduino, lo que facilitará enormemente la labor al alumnado.
 - Ventajas. Se aprovecha todo el potencial de programación y por tanto, hay un mayor control sobre la placa de desarrollo, pudiendo hacer programas mucho más complejos y de mayor envergadura.
 - Inconvenientes. Como todo lenguaje de programación, tiene su dificultad y abstracción. Aunque programar en lenguaje gráfico no está exento de esta dificultad, si es considerablemente menor y más comprensible y, sobre todo, intuitivo.

De las herramientas de programación descritas en entorno gráfico, usaremos principalmente Scraht y todo el código desarrollado en ArduinoIDE, será hecho de la manera más simple, en cuanto a lo que a condicionales y bucles se refiere y en la medida de lo posible, importado directamente de la web de Arduino.

3. ANTECEDENTES

La programación en el entorno educativo ha evolucionado significativamente a lo largo del tiempo y aunque actualmente existe una inmensa oferta de software y placas de desarrollo, no siempre ha sido así.

Si nos remontamos a la década de los años 60, concretamente en 1967, Seymour Papert¹ junto con Wally Feurzeig² y Cynthia Solomon³ desarrollaron el lenguaje de programación Logo, que fue diseñado como una herramienta para que los niños exploraran conceptos matemáticos y computacionales de una manera divertida y accesible. Seymour Papert creía que, al programar en LOGO, los estudiantes podían construir activamente su comprensión de la geometría, la lógica y el pensamiento algorítmico, en línea con los principios del constructivismo.

LOGO se caracterizaba por su simplicidad y su enfoque en la manipulación de gráficos en una pantalla de computadora. Utilizando comandos simples como "avanzar", "girar" y "repetir", los niños podían crear dibujos y patrones mientras desarrollaban estas habilidades computacionales y matemáticas. Además de ser una herramienta de enseñanza efectiva, LOGO también se convirtió en un símbolo del potencial educativo de la informática en el aula.

Seymour Papert trabajó con el psicólogo educativo Jean Piaget en la Universidad de Ginebra, desde 1959 hasta 1963, lo que se refleja en el desarrollo de Logo como una herramienta que permitió a los estudiantes construir activamente su comprensión de conceptos matemáticos y computacionales, al mismo tiempo que promovía un enfoque de aprendizaje activo y autónomo. El legado de Papert y su trabajo continúa influyendo en la educación y la informática educativa hasta el día de hoy.

Seymour Papert propuso el construccionismo como una extensión del constructivismo. Papert argumentaba que los estudiantes aprenden mejor cuando están involucrados en proyectos de construcción activa. En el ámbito de la programación, esto significa que los estudiantes pueden desarrollar un entendimiento más profundo de los conceptos al construir y manipular programas informáticos

El enfoque constructivista enfocado a la enseñanza de la programación sigue los principios fundamentales del constructivismo, adaptándolos al nivel educativo y las distintas necesidades de los alumnos. Estos principios son:

¹ Pretoria, Sudáfrica, 29 de febrero de 1928-Blue Hill, Hancock, Maine, 31 de julio de 2016.

² Chicago, Estados Unidos, 10 de junio de 1927- 4 de enero de 2013

³ Somerville, Estados Unidos, 8 de diciembre de 1938.

Construcción Activa del Conocimiento: El constructivismo enfatiza que los estudiantes construyen su propio conocimiento a través de la interacción con el mundo y la resolución de problemas, ofreciendo a los estudiantes la oportunidad de participar activamente en la creación y manipulación del programa a desarrollar a la vez que adquieren mayor destreza y habilidad en las nuevas tecnologías y especialmente en el uso de las TICs.

Experiencias Precedentes en 3º de ESO: En el caso de que el alumnado al llegar a cuarto curso haya cursado o tenido alguna introducción al mundo del desarrollo de software, puede influir en la forma en que se enseña la programación, permitiendo una mayor profundización en los conceptos y prácticas de programación, lo que dará cierta ventaja y beneficios para los estudiantes que, tras finalizar el curso deseen hacer alguna rama técnica de bachiller.

Aplicación de Herramientas de Programación Gráfica: El constructivismo promueve la exploración, la experimentación y la creatividad y el uso de herramientas y entornos de programación pueden fomentar de manera muy significativa estas destrezas.

Enfoque en la Resolución de Problemas: El constructivismo destaca la importancia de la resolución de problemas de manera autónoma como un medio para construir el conocimiento. Con el desarrollo de software se plantean desafíos y proyectos que requieren que los estudiantes utilicen su habilidad y creatividad

Fomento de la Colaboración y la Creatividad: El constructivismo también enfatiza la importancia de la colaboración entre pares y el fomento de la creatividad. Los proyectos basados en programación gráfica pueden diseñarse de manera que fomenten esta colaboración entre estudiantes mediante el trabajo cooperativo.

Siguiendo una evolución temporal, en 1987, Apple Inc. Desarrolló HyperCard. Este software era una herramienta de autoría multimedia que permitía al usuario crear aplicaciones interactivas. Aunque HyperCard no estaba especialmente desarrollado para el entorno educativo, su enfoque en la interactividad y la facilidad de uso influyó en el desarrollo de entornos de programación visuales posteriores, como pueda ser hoy en día Scratch, del cual hablaremos más adelante.

En el año 1998 apareció LEGO Mindstorms, desarrollado por la empresa de juguetes LEGO, es una línea de kits de robótica que permite a los usuarios construir robots y programarlos para realizar diversas tareas. Si bien LEGO Mindstorms no es un entorno de programación visual en sí mismo, su enfoque en la combinación de construcción física y programación de software inspiró a muchos educadores y desarrolladores a explorar nuevas formas de enseñar

programación a través de la creación de proyectos físicos interactivos. Además, al ser desarrollado por LEGO para sus juguetes, facilita mucho la construcción y posterior programación, pues estos kits vienen acompañados de manuales tanto de construcción como de programación, además de ejemplos.

En resumen, desde el desarrollo de Logo por Seymour Papert, la informática y la programación educativa han evolucionado significativamente, ofreciendo una variedad de herramientas y plataformas que hacen que el aprendizaje de la programación sea accesible, motivador y relevante para los estudiantes de todas las edades. Estas aplicaciones continúan promoviendo un enfoque de aprendizaje activo y creativo, y como es de esperar, han experimentado un notable avance desde el desarrollo de Logo por Seymour Papert hasta la actualidad. Desde entonces, han surgido numerosas aplicaciones y plataformas que continúan esta tradición de enseñanza de la informática y la programación de manera accesible y motivadora. Algunos ejemplos notables incluyen:

- Scratch. es un lenguaje de programación visual desarrollado por el Grupo Lifelong Kindergarten del MIT Media Lab. Está diseñado para enseñar conceptos de programación a niños y principiantes mediante el uso de bloques de código que se encajan juntos en una interfaz gráfica permitiendo a los usuarios crear proyectos interactivos, juegos y animaciones de manera intuitiva. Ofrece una biblioteca de bloques de código que representan acciones y operaciones comunes, eliminando la necesidad de escribir código desde cero. Es ampliamente utilizado en entornos educativos para enseñar pensamiento computacional y habilidades de programación básicas.
- S4A (Scratch for Arduino). es una extensión de Scratch que permite controlar hardware del mundo real utilizando placas de desarrollo Arduino. Combina la simplicidad de Scratch con la capacidad de interactuar con sensores, actuadores y otros dispositivos electrónicos facilitando la integración de hardware y software, permitiendo a los usuarios crear proyectos físicos interactivos.

Proporciona bloques adicionales en la interfaz de Scratch para controlar pines de entrada/salida, leer sensores y enviar señales a actuadores. Esto lo convierte en una herramienta poderosa para la prototipación rápida y la experimentación en el campo de la electrónica y la robótica.

- Lego Mindstorms. Desarrolladas por la empresa de juguetes Lego, es un sistema de robótica educativa que combina bloques de construcción físicos con componentes electrónicos y un entorno de programación específico. Este cuenta con una interfaz gráfica que permite a los usuarios programar robots y dispositivos electrónicos de manera intuitiva. Ofrece una variedad de bloques de código predefinidos que

representan acciones y movimientos del robot, así como la capacidad de crear y personalizar funciones específicas. Es ampliamente utilizado en entornos educativos para enseñar conceptos de robótica, programación y resolución de problemas.

- Arduino. es una plataforma de hardware de código abierto que utiliza microcontroladores para la creación de proyectos electrónicos interactivos. Fue desarrollada para ser accesible para artistas, diseñadores, estudiantes y entusiastas de la electrónica. La programación en Arduino se realiza utilizando un lenguaje basado en C/C++ y una IDE (Entorno de Desarrollo Integrado) fácil de usar. Ofrece una amplia variedad de placas y módulos que se pueden combinar y personalizar para satisfacer las necesidades de proyectos específicos. Arduino es muy versátil y se utiliza en una variedad de aplicaciones, desde el arte interactivo hasta la domótica y la robótica.

Existen otros lenguajes y entornos de desarrollo que, aunque menos conocidos, no son menos potentes ni cuentan con menos herramientas, pero al no ser posible albergarlos todos en el presente documento, se citan los más relevantes, ya sea por número de usuarios en un momento dado o alguna característica o peculiaridad:

- Code.org. Code.org es una organización sin fines de lucro que ofrece cursos de programación gratuitos y accesibles para estudiantes de todas las edades. Su plataforma incluye tutoriales interactivos y actividades basadas en juegos para enseñar habilidades de codificación y promover la alfabetización digital.
- Minecraft: Education Edition: Minecraft, un popular juego de construcción y exploración, ha sido adaptado para su uso en entornos educativos. La versión de educación de Minecraft ofrece herramientas para que los estudiantes construyan y programen mundos virtuales, fomentando la colaboración y la resolución de problemas.
- Microbit. es una pequeña placa de desarrollo diseñada para la educación en programación y electrónica. Fue desarrollada por la BBC Make It Digital en el Reino Unido en 2015, en colaboración con varias organizaciones educativas y tecnológicas. La programación en Microbit se puede realizar mediante un entorno basado en bloques similar a Scratch, así como a través de la programación basada en texto utilizando lenguajes como Python y JavaScript. Incorpora una variedad de sensores integrados, como acelerómetro, brújula y LEDs, así como capacidades de comunicación inalámbrica, lo que lo hace ideal para una amplia gama de proyectos educativos y de bricolaje.

4. RESULTADOS

En apartados anteriores se ha comentado qué es Scratch, tanto a nivel de software como de asociación educativa sin ánimo de lucro. En el presente apartado, se pretende abordar cómo hacer el programa con Echidna a través de Scratch, probarlo e integrarlo en la marioneta para que responda gestualizando en función de la respuesta, pero antes de comenzar a explicar cómo se desarrolla el código en Scratch, así como detallar las configuraciones principales, se explicará qué es el proyecto Echidna Educación y la placa de desarrollo Echidna en su primer versión bautizada como Echidna Shield y su nueva versión con el nombre de Echidna Black.

4.1. Proyecto Echidna Educación

Echidna Educación es una asociación y proyecto educativo *Open Source* sin ánimo de lucro que pretende facilitar el acceso a la programación de sistemas físicos a estudiantes de primaria y secundaria potenciando el desarrollo de la competencia STEM.

Para conseguir esto sus autores⁴ diseñaron y construyeron una placa de desarrollo con varios sensores, con la que el alumnado puede aprender de una manera sencilla y divertida, a crear proyectos interactivos utilizando bloques de programación visual aprovechando el entorno Scratch mediante su IDE de programación `scratch.echidna`.

El proyecto tiene como objetivos los tres siguientes:

1. Promocionar la enseñanza de la programación, y robótica mediante el uso de herramientas de código abierto.
2. Facilitar el acceso a la programación de dispositivos físicos a través de la creación de hardware open source.
3. Crear guías didácticas y videotutoriales que apoyen el uso del hardware educativo y faciliten su uso por parte de otros docentes.

Además, para el cumplimiento de estos fines realiza las siguientes actividades:

- Creación de hardware educativo open source.
- Desarrollo de nuevas herramientas de código abierto.

⁴ En el siguiente enlace web, podemos ver las personas que conforman el equipo <https://echidna.es/quienes-somos/>

- Desarrollo de guías didácticas.
- Charlas y ponencias.
- Creación y mantenimiento de una Web con contenido educativo que sirva de apoyo a la actividad de la asociación.
- Realización de Talleres.

En primer lugar, el proyecto lanzó la placa de desarrollo llamada Echidna Shield, que ayuda en la programación de sensores, actuadores, etc. evitando realizar montaje de circuitería electrónica, lo que es de enorme beneficio ya que el proceso de desarrollo-aprendizaje puede estar centrado casi en su totalidad en la programación, evitando tener que involucrar al alumnado en conceptos más profundos de electrónica que no están pensados hasta cursos posteriores o incluso estudios superiores. Hay que indicar que esta placa es más un Interfaz entre Arduino y el ordenador en el que se está programando.

Tras esto, diseñaron una nueva placa, que han llamado Echidna Black y supone una gran mejora respecto a su antecesora puesto que incorpora un microcontrolador propio, por lo que no es necesario conectarla a Arduino, siendo totalmente autónoma al conectarla al ordenador a través de puerto USB.

Aunque estas placas se pueden programar utilizando Snap4Arduino, nos centraremos en Echidna Scratch, que utiliza todos los elementos disponibles en Scratch además de los bloques de código que añade Echidna para programar las placas.

En los apartados siguientes profundizaremos más sobre las características de ambas placas, su conexión y configuración con el ordenador y posterior programación mediante Echidna Scratch.

4.2. Placa de desarrollo Echidna

Como hemos visto en el apartado anterior, Echidna es una placa de desarrollo basada en Arduino que integra diferentes sensores, pulsadores, LEDs, entradas y salidas tanto pines para conectar con cable o directamente otros dispositivos electrónicos como aéreas, como BlueTooth, orientada al aprendizaje de la robótica y la programación desde educación primaria. A continuación, detallamos las características y componentes de ambas placas, aunque será con Echidna Black con la que se desarrolle el presente Trabajo Final de Máster.

Echidna Shield

- Características:

- Escudo para Arduino
 - Sensores integrados: pulsadores, joystick, acelerómetro, luz
 - Actuadores integrados: LEDs, LED RGB, audio
 - Flexible: entradas y salidas disponibles para complementos
 - 8 entradas tipo Makey Makey
 - Conexión BlueTooth
 - Open Source Hardware Certification OSHWA UID ES000007
- Componentes:
- Pulsadores
 - Joystick
 - Sensor de luz LDR
 - LED RGB
 - Dos salidas para reproducir audio a través de un zumbador
 - Ocho conectores MxMx ultrasensibles para detección de objetos
 - Acelerómetro
 - 3 diodos LED, rojo, naranja y verde
- Esquema de componentes:

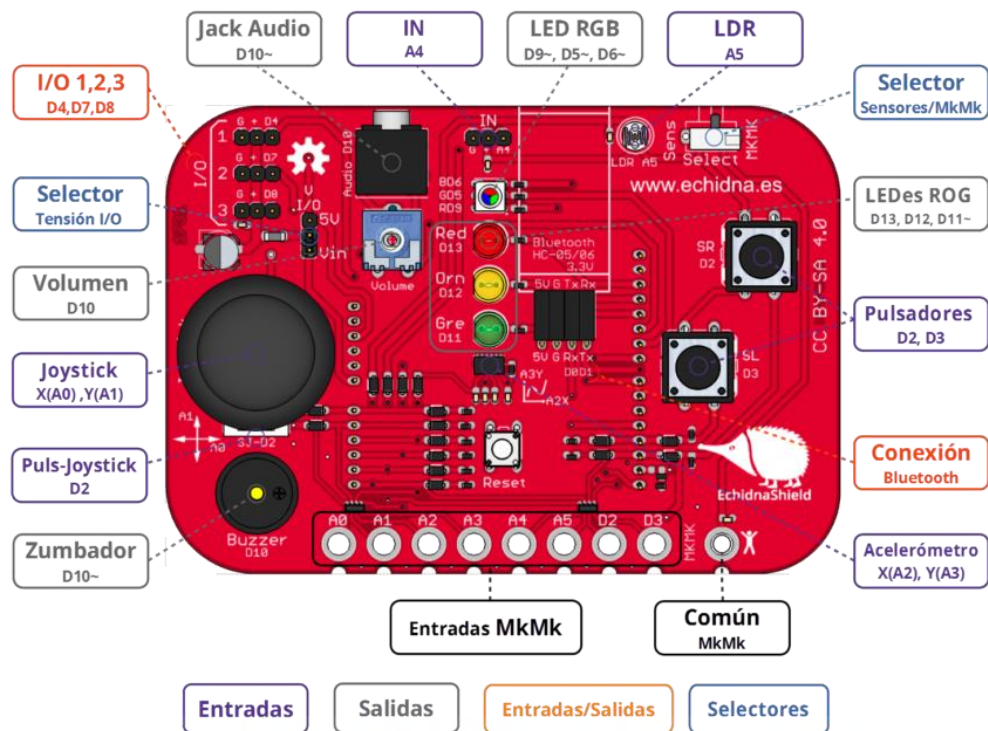


Ilustración 1. Placa de desarrollo Echidna Shield

Echidna Black

- Características:

- o Placa autónoma compatible con Arduino Nano/ Arduino UNO
- o Sensores integrados: pulsadores, joystick, acelerómetro, luz, temperatura, micrófono
- o Actuadores integrados: LEDs, LED RGB, audio
- o Flexible: entradas y salidas disponibles para complementos
- o 8 entradas tipo Makey Makey
- o Conexión BlueTooth
- o Open Source Hardware Certification con registro OSHWA UID ES000010

- Componentes:

- o Pulsadores
- o Joystick

- Sensor de luz LDR
- LED RGB
- Dos salidas para reproducir audio a través de un zumbador
- Ocho conectores MxMx ultrasensibles para detección de objetos
- Acelerómetro
- 3 diodos LED, rojo, naranja y verde
- Micrófono
- Sensor de temperatura

– Esquema de componentes:



Ilustración 2. Placa de desarrollo Echidna Black

4.3. Conexión con el PC y configuración

Lo primero es conectar la placa Echidna Black mediante el puerto USB (ver ilustración 3), que provee de alimentación y comunicación entre el ordenador y la placa. Esta comunicación se hace a través del programa *StandadFirmata*, que viene ya instalado con el programa *Bootloader Optiboot* configurado como Arduino Nano.

Además, debemos tener precaución con haber dado permisos en el ordenador de acceso a la placa al puerto serie y tener instalado el driver *CH430* (en SO de Microsoft).

También es necesario instalar el IDE de Arduino y tras esto, solamente nos quedaría instalar *Echidnalink*. Este último programa se necesita para establecer un enlace entre el navegador web, donde se ejecuta Echidna Scratch y la placa.

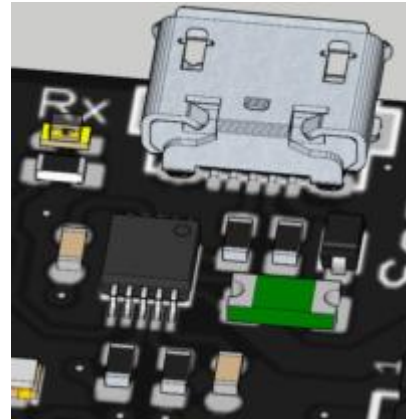


Ilustración 3. Puerto USB de la placa Echidna Black

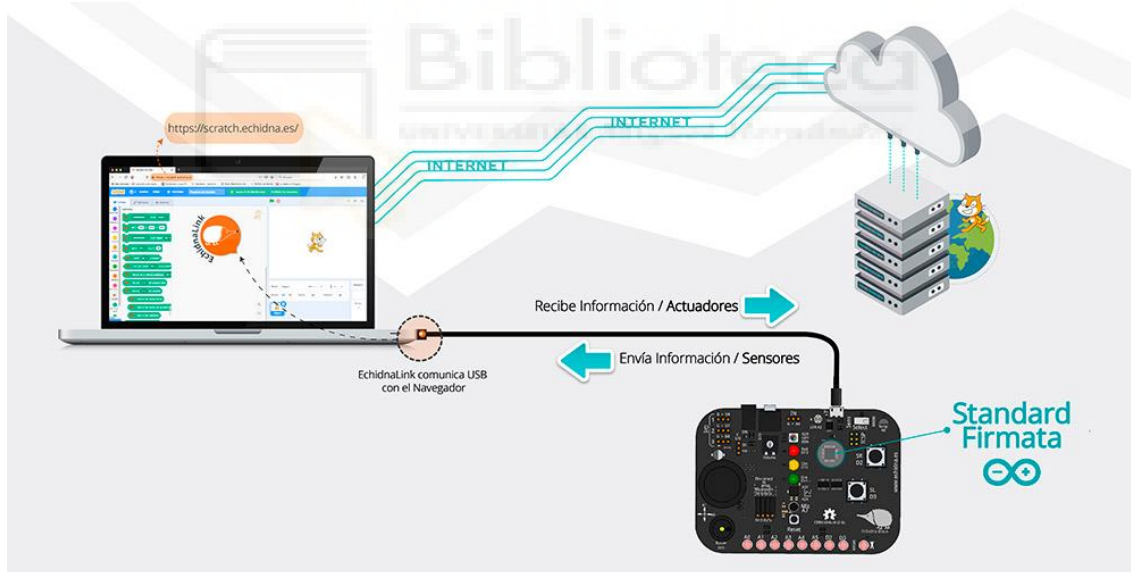


Ilustración 4. Vista gráfica del funcionamiento entre la placa Echidna y el IDE Echidna.scratch

Resumiendo, y teniendo en cuenta que para el alumnado y en determinados casos también en los ordenadores del profesorado, ya tendremos los distintos drivers necesarios instalados, los pasos a seguir serían:

1. Instalar en el ordenador Echidnalink
2. Conectar la placa al puerto USB
3. Ejecutar Echidnalink

Indicar que para hacer la programación en la plataforma Echidna Scratch⁵ es necesario tener la placa conectada al ordenador y totalmente funcional (drivers del párrafo anterior instalados y actualizados), de lo contrario, es posible que nos aparezca un mensaje de error como el de la ilustración 5.

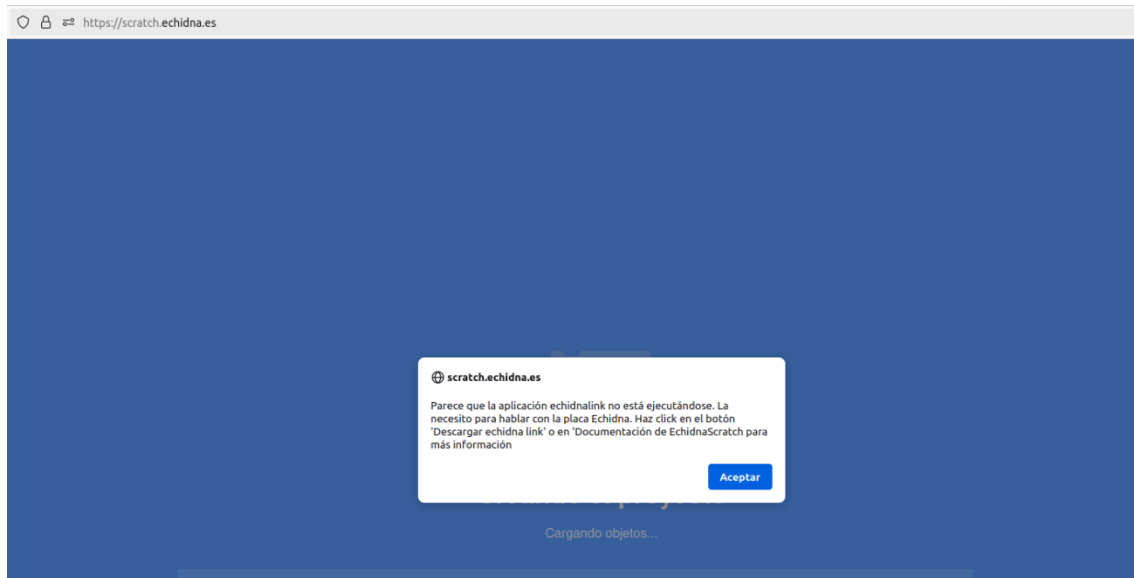


Ilustración 5. Fallo de conexión entre scratch.echidna.es y la placa Echidna Black

En tal caso, pulsamos aceptar y podemos empezar a programar o abrir un proyecto que tengamos almacenado localmente, ya que se abrirá el entorno de programación de Scratch (ver ilustración 6).

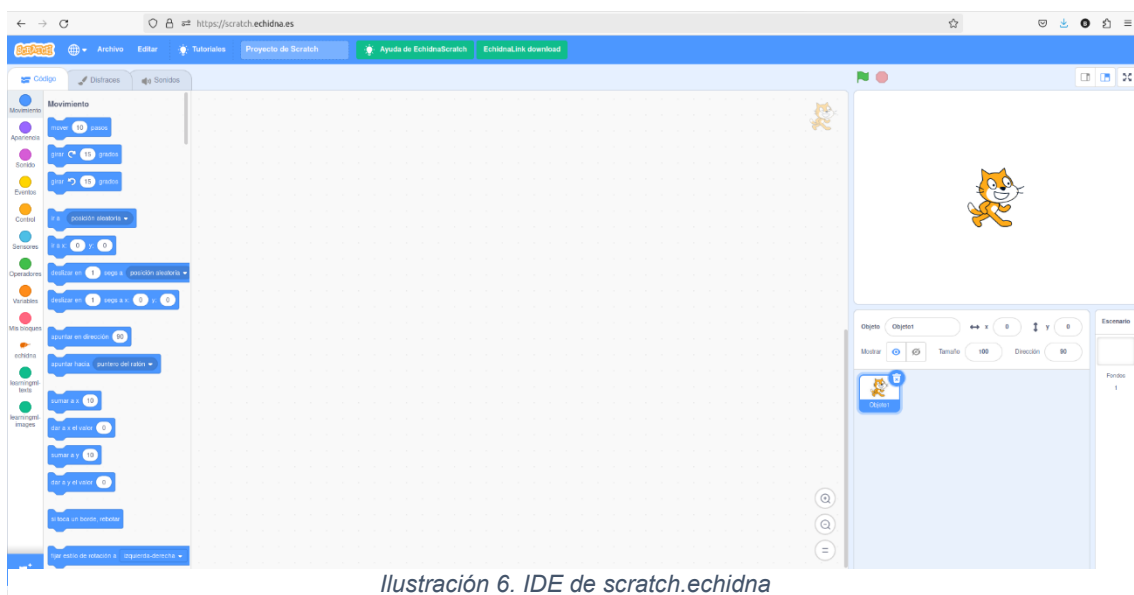


Ilustración 6. IDE de scratch.echidna

⁵ <https://scratch.echidna.es/>

Si no funciona bien teniendo en cuenta el mensaje de error y suponiendo que lo tenemos todo funcional, cerramos Scratch, desconectamos la placa (extrayendo con seguridad) y repetimos los tres pasos del proceso descrito al final del apartado anterior.

4.4. Programación de la marioneta

Antes de ponernos a programar la marioneta, se supone que ha sido construida, como se comentó en el apartado 2. Metodología, colocado los servos en sus ubicaciones con el cableado necesario conectado a la placa Echidna.

Es recomendable al hacer un proyecto de este estilo, concentrar todos los cables en un único punto para poder poner y quitar la placa Echidna con facilidad en la ubicación propuesta. Además, los cables deben tener la menor distancia posible para evitar fallos por caídas de tensión y los servos deben tener un par motor óptimo y estar bien anclados para evitar que tengan demasiada potencia y rompan la construcción o, por el contrario, les falte potencia y no puedan mover el objeto al que estén destinados.

Una vez estamos con el IDE scratch.echidna abierto, como con cualquier código va a depender del autor la manera de programar, pues es un arte en sí mismo y cada persona dentro de unos parámetros para el desarrollo y las explicaciones previas, tiene su manera de diseñar el mismo, pero por poner un ejemplo de programa, podría ser el que a continuación, se describe dando pequeños detalles e ideas.

Antes de iniciarlo resumir que, para el usuario final, se tratará de una aplicación que le preguntará: *¿Cómo te llamas?*, y esperará a que el usuario escriba su nombre. Tras esto e interactuando con él, le hará una serie de preguntas de reconocimiento de símbolos de manera sencilla para que el usuario pueda responder verdadero o falso con dos tarjetas de color verde o rojo, o con el teléfono y un fondo en uno de estos colores, etc., aplicando a la cámara conectada a la placa Echidna para indicar que la respuesta es verdadera o falsa. Cada respuesta dada al programa, nos sumará 1 a *Aciertos* o a *Fallos*.

Cuando acertemos o fallemos, el programa pasará el control a la animación de la marioneta moviendo los servos y además por pantalla nos mostrará las distintas animaciones. Una vez acaben las animaciones y movimientos, volverá el control del programa a la siguiente pregunta.

Dejando por el momento de lado la web de scratch.echidna, antes de ello debemos crear y entrenar la inteligencia artificial para reconocer el color verde o rojo para el verdadero o falso a las preguntas, utilizando imágenes de color verde

y rojo en distintas tonalidades utilizando un teléfono móvil y objetos en estos dos colores.

En primer lugar, accedemos a la web: <https://scratch.echidna.es/learningml/>, pulsamos en *Reconocer imágenes* y tenemos que realizar los tres pasos que aparecen en la web (señalados en rojo en la ilustración 6):

1. Entrenar, donde mostraremos imágenes para verdadero en color verde y rojo con las etiquetas *VERDADERO* y *FALSO*
2. Aprender a reconocer las imágenes
3. Probar y una vez visto que funcione con cierta fiabilidad pulsamos el icono de Scratch

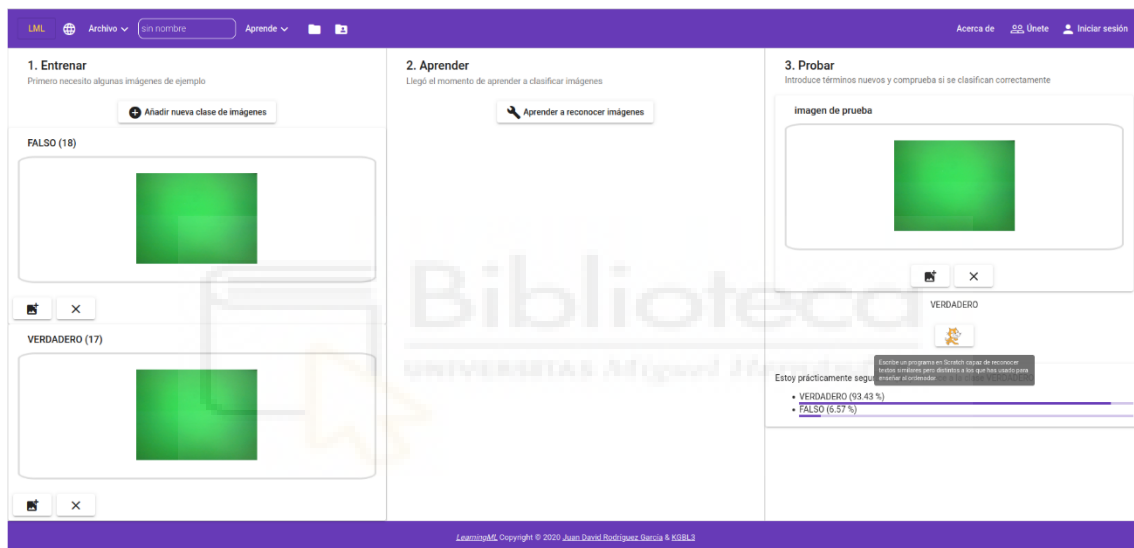


Ilustración 7. Prueba de funcionamiento del entrenamiento de las variables de machine learning creadas para nuestro código

Cuando pulsemos el icono de Scratch como decía el paso 3, marcado en verde y con la ayuda que aparece al poner el cursor encima, como aparece en la ilustración 7, se nos cargará el entorno [scratch.echidna](https://scratch.echidna.es) como aparece en la ilustración 6.

Atendiendo a la necesidad de simpleza en el código de manera que sea fácilmente asumible por el alumnado, podemos realizar un programa en el que primeramente nos pregunte el nombre, esperando recibir una respuesta por teclado, así el programa puede interactuar de manera personal con el usuario y resulta muy interesante para el alumnado, ver como un programa creado por ellos es capaz de interactuar con el nombre del jugador etc. En la ilustración 8 tenemos un ejemplo de un bloque de código hecho para tal fin.



Ilustración 8. Ejemplo de bloque de código 1

Una vez recibido el nombre, podemos crear un disfraz para los movimientos de la marioneta, lo que se traduce en primeramente llevarnos los servos a la posición cero, lo que será la posición “relajada” o sin movimiento y a partir de ahí establecemos los movimientos que deseemos.



Ilustración 9. Ejemplo de bloque de código 2

Para hacer los movimientos, hacemos un subprograma llamado *Aplauso* en el que ajustamos los servos para que hagan movimientos, a la vez decimos el

nombre del jugador para felicitarlo y tras esto pasamos a la siguiente pregunta, en nuestro caso el siguiente Disfraz. En la ilustración 9 podemos ver este ejemplo de subprograma.

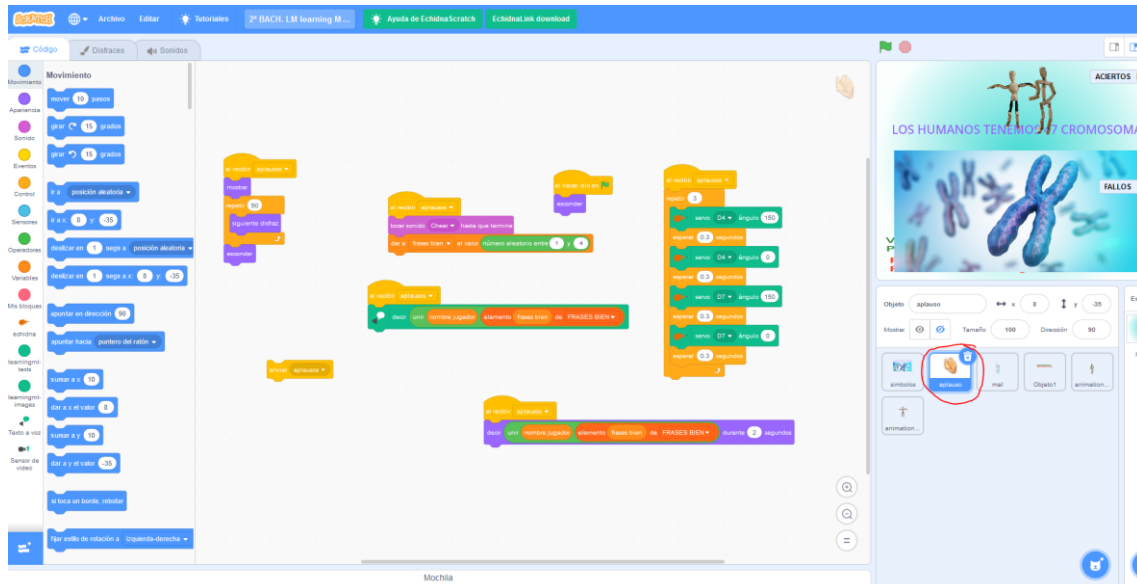


Ilustración 10. Subprograma "Aplauso"

De la misma manera, cuando tengamos un fallo, se puede crear un subprograma

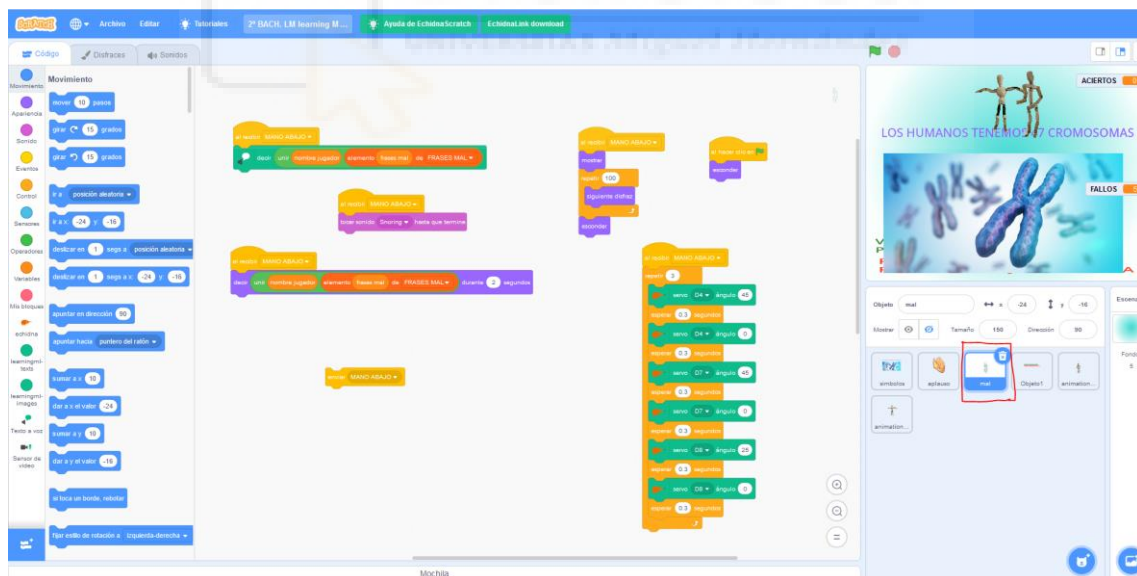


Ilustración 11. Subprograma "Mal"

Continuando con el programa, desde "variables", podemos crear una lista con las preguntas y otra con las respuestas.

En la pestaña *Disfraces* podemos incluir y cargar las imágenes que deseemos para asociarlas por ejemplo a las preguntas anteriores, los aciertos o los fallos en las respuestas, inicio y fin del juego, etc. y al igual en la pestaña *Sonidos*,

cargar e incluir todos los sonidos y canciones necesarias de la misma manera que los *Disfraces*, llamarlos durante la ejecución del programa como hayan sido programados. Estos *Disfraces*, contendrán las imágenes usadas para cada pregunta en el caso del subprograma *Símbolos* que es el programa principal, del mismo modo, en cada subprograma podemos tener distintas imágenes para usarlas como consideremos durante la ejecución de ese subprograma cuando sea llamado por el principal y este último, coja el control de ejecución.

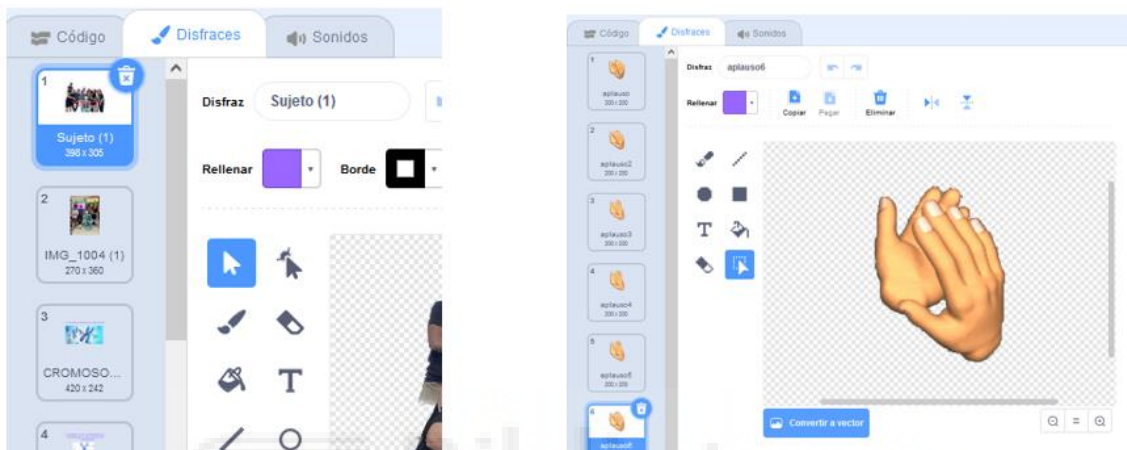


Ilustración 12. A la izquierda los disfraces del programa principal. A la derecha los disfraces del subprograma Aplauso con las distintas imágenes o Disfraces para simular movimiento en el aplauso

Resumiendo, la funcionalidad de las tres pestañas de la parte superior izquierda, tenemos:

- *Código*. Que será donde creamos el código fuente de ese subprograma
- *Disfraces*. Que será donde almacenemos las imágenes utilizadas durante la ejecución de este subprograma.
- *Sonidos*. Que será donde almacenemos los audios utilizados durante la ejecución de este subprograma.

De la misma manera, se deben crear todas las variables necesarias para el programa que deseemos y vayamos a necesitar.

En cuanto a la función principal del programa, podemos ver el ejemplo de la ilustración 10, marcado en rojo el lugar en que llamamos al sensor de video para el reconocimiento de la imagen mostrada a la cámara.

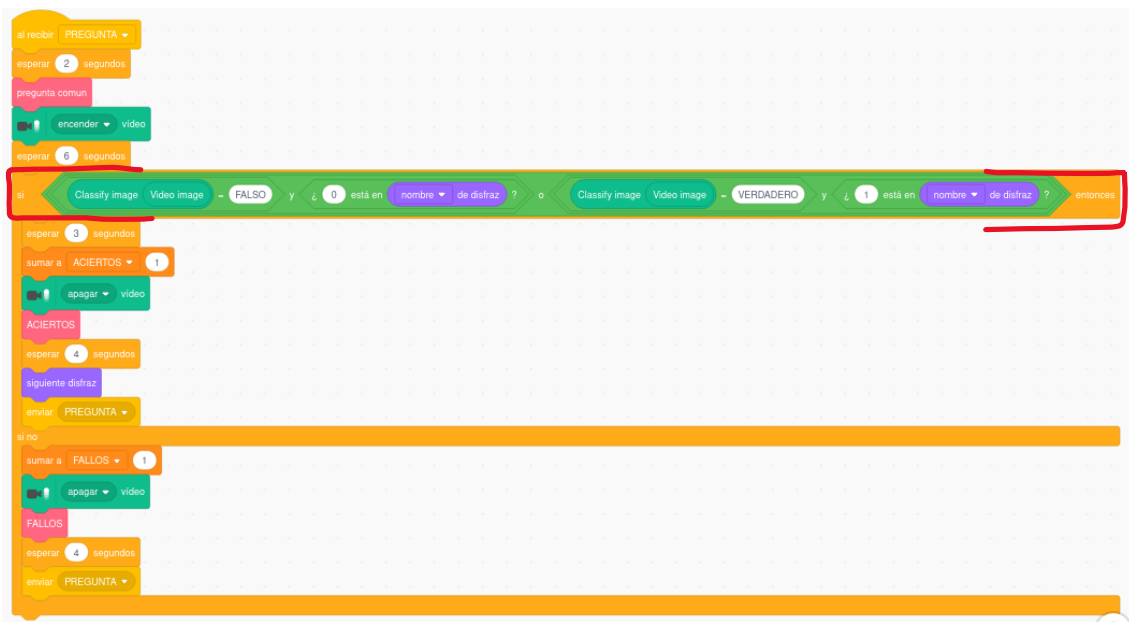
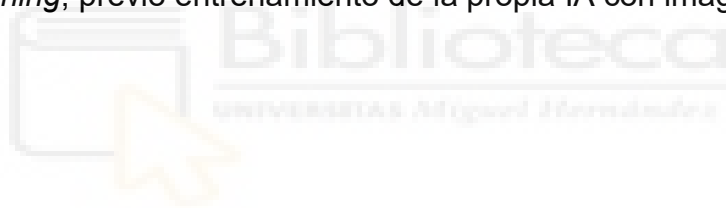


Ilustración 13. Ejemplo de bloque de código 3

Con estos pasos, podemos ver una guía ejemplo de cómo realizar fácilmente un programa de reconocimiento de imágenes en tiempo real de ejecución mediante *machine learning*, previo entrenamiento de la propia IA con imágenes.





5. DISCUSIÓN Y CONCLUSIONES

El proyecto de desarrollo de un juego de robótica para la placa Echidna Black destinado a alumnos de cuarto curso de ESO representa una oportunidad única para integrar conceptos de programación, tecnología y *machine learning* en el ámbito educativo. La utilización de herramientas como Scratch y la implementación de una marioneta accionada por servos que reacciona a imágenes, abre las puertas a un aprendizaje interactivo y práctico.

La introducción de la programación en entorno gráfico desde edades tempranas, como se propone en este proyecto, fomenta el desarrollo de habilidades clave en los estudiantes, tales como la resolución de problemas, la creatividad y la lógica computacional. Además, al centrarse en la placa Echidna, se promueve el uso de tecnologías actuales y accesibles, preparando a los alumnos para futuros desafíos en campos relacionados con la tecnología e ingeniería.

La implementación de este proyecto ha contribuido significativamente al desarrollo de habilidades clave en los estudiantes, como la resolución de problemas, la creatividad y la lógica computacional. Además, ha fomentado la adquisición de competencias STEM fundamentales, preparando a los alumnos para enfrentar los desafíos tecnológicos del futuro.

El desarrollo de un proyecto de estas características, también es impulsor a la vez que fomenta la competencia personal, social y de aprender a aprender, debido a la necesidad de autonomía para el desarrollo del mismo, sin olvidar la ampliación y mejora de la competencia digital.

En conclusión, el proyecto de desarrollo de un juego de robótica para la placa Echidna se presenta como una herramienta educativa innovadora y motivadora para los estudiantes de cuarto curso de ESO. Al combinar la programación en entorno gráfico, el uso de sensores y la aplicación de *machine learning*, se potencia el aprendizaje significativo y la adquisición de competencias fundamentales en la sociedad actual.

Este proyecto no solo busca enseñar conceptos técnicos, sino también promover habilidades como el trabajo en equipo, la comunicación y la resolución de problemas, aspectos esenciales para el desarrollo integral de los estudiantes. En definitiva, la integración de la tecnología en el aula a través de proyectos como este, contribuye a la formación de individuos preparados para afrontar los retos del siglo XXI en un mundo cada vez más digitalizado y tecnológico.

5.1. Recomendaciones y líneas futuras

1. Ampliar el alcance del proyecto: Considerar la posibilidad de expandir el proyecto a más cursos y niveles educativos para llegar a un mayor número de estudiantes y promover el interés en la programación y la robótica desde edades tempranas.
2. Crear una comunidad educativa: Fomentar la creación de una comunidad educativa en torno al uso de la placa Echidna y Scratch, donde docentes y estudiantes puedan compartir experiencias, recursos y buenas prácticas para enriquecer el aprendizaje. Además, se puede integrar esta comunidad con la comunidad Echidna Educación, lo que favorece, fomenta y enriquece el conocimiento.
3. Continuar la formación docente: Ofrecer formación continua a los docentes interesados en implementar proyectos de robótica y programación en el aula, brindando herramientas y recursos para potenciar sus habilidades pedagógicas en este ámbito.
4. Explorar nuevas aplicaciones educativas: Investigar y desarrollar nuevas aplicaciones educativas y proyectos interactivos que integren la placa Echidna y Scratch, con el objetivo de seguir motivando a los estudiantes y enriqueciendo su experiencia de aprendizaje.

En cuanto al código a desarrollar, lo más importante e inmediato una vez adquirido el conocimiento por parte del alumnado, sería crear una variable más para el uso de imágenes distintas al rojo o verde, de manera que modifiquemos el bloque de código principal (ver ilustración 10) incluyendo esta nueva variable. De esta manera, damos nuevas e infinitas posibilidades a los alumnos de introducir mejoras en el código como puedan ser:

1. Eliminar tiempos de espera de la cámara de manera que el código solo espere a recibir rojo o verde mientras de manera autónoma descarta el resto de imágenes o colores.
2. Evitar errores por confusión de colores o falta de entrenamiento y aprendizaje del propio código.
3. Agilizar el desarrollo del programa al poder eliminar retardos innecesarios.

En resumen, el proyecto de desarrollo de un juego de robótica para la placa Echidna Black en entorno scratch.echidna ha demostrado ser una iniciativa educativa valiosa que promueve el aprendizaje activo, la creatividad y el desarrollo de habilidades en ciencias, tecnología, ingeniería y Matemáticas, fomenta el aprendizaje autónomo y por tanto ser individuos autocríticos, lo que te compete personalmente y dado que este proyecto se enmarca en una

situación de aprendizaje enfocada al uso de las TICs, se fomentan las competencias digitales entre otras en los estudiantes.

Su continuación y expansión pueden tener un impacto positivo en la educación y en la formación de futuras generaciones de profesionales en el campo científico-humano-tecnológico.





6. CONTRIBUCIONES PRÁCTICAS

Durante la realización de este proyecto, se identificaron varias contribuciones prácticas que pueden enriquecer el ámbito educativo y tecnológico:

- Desarrollo de un juego de robótica educativo: La creación de un juego de robótica para la placa Echidna ha permitido integrar conceptos de programación visual, sensores y *machine learning* en un entorno educativo, brindando a los estudiantes una experiencia interactiva y motivadora para aprender sobre tecnología y robótica.
- Creación de recursos educativos: Se ha generado una guía didáctica interactiva que acompaña al juego de robótica, proporcionando a docentes y estudiantes un recurso detallado para comprender y aplicar los conceptos enseñados, fomentando así un aprendizaje autónomo y significativo.
- Organización de talleres y charlas educativas: La realización de talleres interactivos y charlas educativas para docentes ha permitido compartir conocimientos, experiencias y buenas prácticas en la enseñanza de la programación y la robótica, fortaleciendo la comunidad educativa y promoviendo la integración de la tecnología en el aula.
- Promoción de la competencia STEM: El proyecto Echidna Educación y la placa Echidna han contribuido a promover la competencia STEM (Matemáticas, Ciencia, Tecnología, y Arte) entre los estudiantes de primaria y secundaria, fomentando un enfoque interdisciplinario y creativo en el aprendizaje de las disciplinas STEM.

En resumen, las contribuciones prácticas derivadas de este proyecto han impactado positivamente en la educación tecnológica, promoviendo el aprendizaje activo, la creatividad y el desarrollo de habilidades clave en los estudiantes, y fortaleciendo la comunidad educativa en torno a la programación y la robótica.

Estas contribuciones representan un paso importante hacia la formación de futuras generaciones de profesionales en el campo de la tecnología y la innovación.



7. REFERENCIAS, BIBLIOGRAFÍA Y ENLACES WEBS

7.1. Referencias y bibliografía

- Boyle, M. (1999). The History of Mr. Papert. *Logo Exchange, Journal of the ISTE Special Interest Group for Logo-Using Educators.*, 17, 8-14.
- Feurzeig, W., Papert, S., Bloom, M., Grant, R., & Solomon, C. (1970). Programming-languages as a conceptual framework for teaching mathematics. *ACM SIGCUE Outlook*, 4(2), 13-17.
- Genalo, L. (2004, June). Piaget and engineering education. In *2004 Annual Conference* (pp. 9-988).
- Hernández-López, B. (2019). *Desarrollo e implementación de un dispositivo automático basado en microcontrolador* [Trabajo Final de Grado]. Escuela Politécnica Superior de la Universidad de Alicante.
- Magno, H. *Models of Learning*.
- Mancha, C. L., Valenciana, C., Madrid, C. D., Murcia, R. D., Navarra, C. F. D., & SEK, I. E. (2018). MECD (INTEF y CNIIE) Andalucía Balears, Illes Cantabria.
- Papert, S. (1980). LOGO's roots: Piaget and AI. *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Papert, S. (1993). The children's machine. *TECHNOLOGY REVIEW-MANCHESTER NH-*, 96, 28-28.
- Papert, S., & Harel, I. (2002). Situar el construccionismo. Alajuela: INCAE. Clares, M. (2021, 9 febrero). Inicio - Echidna Educacion. Echidna Educacion. <https://echidna.es/>
- Pérez, V. F. C., & Molano, E. G. (2023). Incorporación del enfoque stem con tictac en la construcción de un robot sembrador con arduino. *Aglaia*, 14(1), 127-141.
- Ruiz Vicente, F., Zapatera, A., Montes, N., & Rosillo, N. (2019, May). Proyectos STEM con LEGO Mindstorms para educación primaria en España. In *INNODOCT/18. International Conference on Innovation, Documentation and Education* (pp. 711-720). Editorial Universitat Politècnica de València.

Saldarriaga, G. E., Lasso, M. A. M., Sánchez, E. R. V., Badilla-Saxe, E., Quinn, M., Alzati, E. C., & Solórzano, C. M. V. (2020). *Constructores de conocimiento: Papert y su visión* (Vol. 1). Corporación Universitaria para el Desarrollo de Internet.

Sánchez Vera, M. D. M., González-Martínez, J., & Reina Herrera, M. (2019). Marta Reina Herrera [Entrevista].

White, B. Y. (1985). Thinking about learning about thinking: an interview with Seymour papert. *New ideas in psychology*, 3(3), 287-292.

7.2. Enlaces a webs de consulta

Características – Echidna. (s.f.). <https://echidna.microlog.es/caracteri/>

Colaboradores de Wikipedia. (2022b, agosto 17). *Wally Feurzeig*. Wikipedia, la Enciclopedia Libre. https://es.wikipedia.org/wiki/Wally_Feurzeig

Colaboradores de Wikipedia. (2024a, febrero 4). *Logo (lenguaje de programación)*. Wikipedia, la Enciclopedia Libre. [https://es.wikipedia.org/wiki/Logo_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Logo_(lenguaje_de_programaci%C3%B3n))

Colaboradores de Wikipedia. (2024b, marzo 22). *Seymour Papert*. Wikipedia, la Enciclopedia Libre. https://es.wikipedia.org/wiki/Seymour_Papert

Echidna. (s.f.). <https://echidna.microlog.es/>

<https://ceice.gva.es/documents/162640733/162655319/Tecnolog%C3%ADa.pdf/f1cfb52d-6819-4c33-983a-556885b0479f?t=1461853077221>

<https://ceice.gva.es/es/web/ordenacion-academica/secundaria/curriculo>

<https://educagob.educacionfpydeportes.gob.es/curriculo/curriculo-lomloe/menu-curriculos-basicos/ed-primaria/competencias-clave/personal-social.html>

<https://educagob.educacionfpydeportes.gob.es/curriculo/curriculo-lomloe/menu-curriculos-basicos/bachillerato/competencias-clave/mat-ciencia-tec.html>

<https://educagob.educacionfpydeportes.gob.es/curriculo/curriculo-lomloe/menu-curriculos-basicos/ed-secundaria-obligatoria/materias/tecnodigitali/competencias-especificas.html>

SOFTWARE – Echidna. (s.f.). <https://echidna.microlog.es/programs/>

The LEGO® Group. (s.f.). *The LEGO® Group*. <https://www.lego.com/es-es/categories/coding-for-kids>

Wikipedia contributors. (2023, 12 octubre). *Cynthia Solomon*. Wikipedia.
https://en.wikipedia.org/wiki/Cynthia_Solomon





8. ANEXOS

8.1. Situación de aprendizaje



CURSO ACADÉMICO: 2023/2024				ÁREA/MATERIA: TECNOLOGÍA Y DIGITALIZACIÓN				NIVEL Y GRUPO: 4º				NÚMERO DE SESIONES: 12							
TEMPORALIZACIÓN																			
Septiembre		Octubre		Noviembre		Diciembre		Enero		Febrero		Marzo		Abril		Mayo		junio	
SITUACIÓN DE APRENDIZAJE NÚMERO 4	TÍTULO Programación con Scratch en Echidna Black.							CONTEXTO Los estudiantes de este nivel educativo se encuentran en una etapa de desarrollo cognitivo y habilidades tecnológicas donde se busca fomentar su creatividad, conocimiento y destrezas en el ámbito de las Tecnologías de la Información y la Comunicación (TIC). Partimos de la introducción a la informática, el uso de las TICs y las nuevas tecnologías aplicadas al entorno educativo y fuera de él.											
	DESCRIPCIÓN Y JUSTIFICACIÓN. <p>Las situaciones de aprendizaje integran todos los elementos que constituyen el proceso de enseñanza y aprendizaje competencial encaminadas al logro de los objetivos. Se plantean tareas en las que el alumnado moviliza un conjunto de recursos y saberes con el fin de afrontar dichas situaciones, articulando los saberes básicos mediante tareas significativas y relevantes, resolviendo retos de manera autónoma y creativa. Se pretende así, promover la transferencia de lo aprendido a situaciones cercanas a la vida real, siempre desde procesos pedagógicos flexibles y accesibles, ajustados a las NEE del alumnado, analizando el contexto en el que dar sentido a los aprendizajes y estimularlos y en función de la propuesta planteada, siendo así posible reflexionar sobre el grado de conexión de la situación planteada con el mundo real.</p> <p>Es recomendable, además, que el conjunto de la información, datos y soluciones generadas mediante la construcción de prototipos, el análisis de objetos o el diseño de experimentos, permitan la argumentación y fundamentación a la hora de tomar decisiones, siendo otra de las claves el ajuste del nivel de autonomía del alumnado, que comenzará con el desarrollo de propuestas casi completamente guiadas hasta una actuación totalmente autónoma a medida que se alcancen mayores niveles de desarrollo competencial. A continuación, se relacionan los principales objetivos a alcanzar:</p> <ul style="list-style-type: none"> • <i>Mejorar el aprendizaje</i> • <i>Fomentar competencias digitales</i> • <i>Personalización del aprendizaje</i> • <i>Promover la colaboración</i> • <i>Desarrollar habilidades críticas</i> • <i>Relevancia tecnológica</i> • <i>Acceso a información</i> • <i>Interactividad y motivación</i> • <i>Preparación para el futuro</i> • <i>Inclusión educativa</i> 							RELACIÓN CON LOS RETOS DEL S. XXI Y CON LOS ODS <ul style="list-style-type: none"> • Digitalización y Transformación Digital. Prepara a los estudiantes para vivir y trabajar en un mundo cada vez más digitalizado. Esto es crucial en un siglo XXI donde la tecnología está en el centro de casi todas las actividades humanas. • Competencias para el Empleo del Futuro. Las TIC en la educación aseguran que los estudiantes adquieran competencias necesarias para la adaptabilidad y el éxito profesional en un mercado laboral en constante cambio. • Aprendizaje Continuo y Adaptativo. Esencial en un mundo donde el conocimiento y las habilidades deben actualizarse constantemente. • Desigualdad y Acceso a la Educación: La tecnología puede ayudar a reducir las desigualdades en el acceso a la educación. • Pensamiento Crítico y Resolución de Problemas Promover el desarrollo de habilidades críticas, creativas y de resolución de problemas, que son fundamentales para enfrentar los complejos desafíos globales del siglo XXI. • ODS 4: Educación de Calidad: El uso de las TIC en la ESO contribuye directamente a mejorar la calidad de la educación, promoviendo el acceso a una enseñanza inclusiva y equitativa. Facilita el aprendizaje a lo largo de toda la vida, una meta clave de este objetivo. • ODS 5: Igualdad de Género: Las TIC pueden ser una herramienta poderosa para empoderar a las niñas y mujeres, proporcionándoles acceso a la educación y reduciendo la brecha de género en el acceso a recursos educativos. • ODS 8: Trabajo Decente y Crecimiento Económico: Al equipar a los estudiantes con habilidades digitales, las TIC preparan a los jóvenes para empleos de calidad y promueven el crecimiento económico inclusivo. • ODS 9: Industria, Innovación e Infraestructura: La educación en TIC fomenta la innovación y el desarrollo de infraestructuras tecnológicas, esenciales para el desarrollo sostenible. 											

SABERES BÁSICOS	Los saberes básicos exigidos para la adquisición y desarrollo de las competencias específicas se organizan en los siguientes bloques: Proceso de resolución de problemas; Digitalización del entorno personal de aprendizaje; Pensamiento computacional, programación, control y robótica; Materiales, productos y soluciones tecnológicas; Creación, expresión y comunicación; Tecnología sostenible.				
ORGANIZACIÓN	SECUENCIACIÓN DE ACTIVIDADES / DISTRIBUCIÓN DEL TIEMPO (indicar el número de sesiones de cada actividad)	EVALUACIÓN <ul style="list-style-type: none"> • Escalas de evaluación con criterios que puedan usarse y adaptarse por parte del docente o el alumnado • Co-evaluación y autoevaluación • Propuesta de portfolio de aprendizaje con diario de control de versiones y modificaciones sobre el proyecto 	MEDIDAS DE RESPUESTA PARA LA INCLUSIÓN: NIVEL II <ul style="list-style-type: none"> • Metodologías • Agrupamientos • Recursos materiales • Recursos organizativos • Organización de los espacios 	MEDIDAS DE RESPUESTA PARA LA INCLUSIÓN: NIVEL III Recursos materiales y personales adicionales ordinarios	MEDIDAS DE RESPUESTA PARA LA INCLUSIÓN: NIVEL IV Recursos materiales y personales adicionales especializados
	Aprendizaje accesible. Considerar el grado de accesibilidad de los aprendizajes para que éstos se adapten a las necesidades, características y diferentes ritmos de aprendizaje del alumnado, como por ejemplo la accesibilidad física, cognitiva, emocional y la accesibilidad sensorial, todas para proporcionar y fomentar la mejor situación para el aprendizaje.				
	Motivación Motivar al alumnado mediante diferentes estrategias que promuevan la implicación y participación en todas y cada una de las situaciones de aprendizaje con el fin de crear personas totalmente competentes, seguras y funcionales.	Representación Mediante la adecuación a las NEE se facilitará la accesibilidad física, sensorial y cognitiva presentando una información clara y concisa, que favorezca la reflexión y el procesamiento de la información. Así mismo se facilitará y proporcionará al alumnado para la autoevaluación, diferentes modelos, pautas de técnicas de autoevaluación	Acción y expresión Facilitar el uso de diferentes medios de expresión del conocimiento a través de aplicaciones informáticas de software empleando el uso de las TICs, llevando un seguimiento de manera individualizada del alumnado.		







MASTERPROF UMH
UNIVERSITAS *Miguel Hernández*

**MÁSTER UNIVERSITARIO EN FORMACIÓN DEL PROFESORADO
ESO Y BACHILLERATO, FP Y ENSEÑANZAS DE IDIOMAS**



Biblioteca
UNIVERSITAS *Miguel Hernández*