



## TRABAJO FIN DE MÁSTER

# ROBÓTICA PARA LA ESO CON PLACA ECHIDNA Y SUS SENSORES CON MARIONETA ACCIONADA POR SERVOS

Estudiante: Miguel Ángel García Lidón

Especialidad: Tecnología

Tutor/a: Juan Ramón Fernández González

Curso académico: 2023-24

## ÍNDICE

|   |    |
|---|----|
| 1. Resumen y palabras clave. ....                                     | 3  |
| 2. Introducción, planteamiento del problema y objetivos. ....         | 4  |
| 3. Revisión bibliográfica. ....                                       | 5  |
| 3.1 Estudios que apoyan la inclusión de la robótica en educación .... | 5  |
| 3.2 Origen e historia de la robótica educativa. ....                  | 7  |
| 3.3 Arduino ....  | 10 |
| 3.4 Raspberry PI. ....  | 14 |
| 3.5 Micro:bit ....  | 18 |
| 3.6 Echidna. ....   | 21 |
| 3.7 Tipos de Sensores y actuadores más importantes. ....              | 24 |
| 3.8 Scratch. ....   | 28 |
| 4. Propuesta. ....  | 30 |
| 4.1 Construcción marioneta. ....                                      | 31 |
| 4.2 Sensores y actuadores. Conexiones físicas. ....                   | 32 |
| 4.3 Programación ....   | 34 |
| 5. Conclusiones y futuros trabajos ....                               | 39 |
| 6. Bibliografía ....  | 42 |
| 7. Anexos ....  | 44 |

## 1. Resumen y palabras clave.

El trabajo que se propone se centra en introducir el pensamiento computacional y la robótica en la educación de los estudiantes de Tecnología de 4º curso de la ESO. El pensamiento computacional y la robótica es la competencia específica 4 en el decreto 107/2022 de 5 de agosto y un saber básico dentro del bloque 3 del mismo. Se llevará a cabo una revisión bibliográfica sobre la idoneidad de incluir la robótica en la educación, de las placas disponibles en el mercado, sus sensores, actuadores y su evolución a lo largo del tiempo. Posteriormente, se diseñará una propuesta que permita a los alumnos adquirir habilidades en programación y robótica, con el objetivo de desarrollar su lógica computacional desde una edad temprana.

Se decidirá la placa más adecuada y el lenguaje de programación para realizar la práctica donde los estudiantes trabajarán en equipos para construir una marioneta que, accionada por servos, reaccione al activarse los sensores de una de estas placas. Este proyecto, además de una experiencia práctica en programación y robótica, también fomentará el trabajo en equipo, la resolución de problemas y la creatividad. En resumen, se busca preparar a los estudiantes para resolver problemas en un mundo cada vez más digitalizado y tecnológico, dotándolos de habilidades importantes para su futuro.

Palabras clave: Pensamiento computacional, educación, scratch, controladora, equipo, habilidades, programación bloques.

The proposed work focuses on introducing computational thinking and robotics in the education of 4th year ESO Technology students. Computational thinking and robotics is specific competence 4 in decree 107/2022 of August 5 and basic knowledge within block 3 of the same. A bibliographic review will be carried out on the suitability of including robotics in education, the boards available on the market, their sensors, actuators and their evolution over time. Subsequently, a proposal will be designed that allows students to acquire skills in programming and robotics, with the aim of developing their computational logic from an early age.

The most appropriate board and programming language will be decided upon for the practice where students will work in teams to build a puppet that, driven by servos, reacts when the sensors on one of these boards are activated. This project, in addition to a practical experience in programming and robotics, will also encourage teamwork, problem solving and creativity. In short, it seeks to prepare students to solve problems in an increasingly digitized and technological world, equipping them with important skills for their future.

Keywords: Computational thinking, education, scratch, controller, equipment, skills, block programming.



## 2. Introducción, planteamiento del problema y objetivos.

En el ámbito de la asignatura de Tecnología de 4º de la ESO, se pueden encontrar entre sus saberes básicos y sus competencias específicas el pensamiento computacional y la robótica. Se considera que son herramientas fundamentales para el desarrollo académico y muy probablemente para el futuro profesional de los estudiantes.

En este entorno que nos rodea, cada vez más digitalizado y dependiente de la tecnología y la informática, se considera crucial adquirir estos conocimientos, despertar ese pensamiento lógico y llevarlo a cabo en aplicaciones prácticas que nos podemos encontrar en el día a día.

Pero hay que tener en cuenta que se va a trabajar con alumnos en edad relativamente temprana. Es por ello que antes de introducir a los estudiantes en el mundo de la lógica computacional, la aplicación de la programación y la robótica, se debe considerar cuáles son las herramientas más adecuadas para llevar a cabo esta inmersión (hablamos de una introducción a la robótica). De tal manera que se puedan encontrar esas herramientas idóneas para que el joven alumno pueda comprender y que sea capaz de desarrollar ese pensamiento computacional sin crear rechazo.

Además, las herramientas que se usen deben ser viables tanto a nivel económico, como de espacio en las aulas. El problema es que nos encontramos ante una variedad importante de tipos de placas para poder trabajar la robótica y varios entornos de programación. Por lo tanto, el objetivo que se plantea es el de recabar la máxima información posible sobre las placas existentes y decidir qué placa robótica y qué lenguaje de programación son los ideales para posteriormente llevar a cabo una propuesta de enseñanza. Una vez seleccionadas las herramientas, se debe proponer por lo tanto esta propuesta que lleve a la práctica los saberes que los alumnos van a adquirir.

¿Es el uso de la robótica un buen método de enseñanza a nivel de lógica computacional?

¿Cuál es la placa robótica ideal para que los alumnos comiencen su incursión en la robótica?

¿Qué entorno de programación se debe usar?

Una vez elegidas la placa y el entorno de programación ¿Qué propuesta de enseñanza se va a plantear?

Los objetivos principales se pueden resumir en estas cuestiones planteadas. La labor de este trabajo será dar respuesta a las mismas.

### **3. Revisión bibliográfica.**

La presente revisión bibliográfica tiene como objetivo realizar un repaso por la evolución de la robótica en la educación y profundizar en la comprensión de los diversos tipos de placas robóticas que hay disponibles en el mercado (Arduino, Raspberry Pi, micro:bit...). En el contexto educativo que estamos tratando, estas placas pueden ser herramientas fundamentales a la hora de la enseñanza de la robótica y el pensamiento computacional para los alumnos de 4º curso de la ESO. Se realizará un análisis de la literatura académica y técnica disponible, así como de los recursos en línea más relevantes. Se trabajarán los aspectos como las características técnicas, capacidades de programación o las aplicaciones educativas de cada placa. De esta manera se obtendrá una base sólida para decidir cuál es la placa que se debe elegir para llevar a cabo nuestra estrategia pedagógica.

#### **3.1 Estudios que apoyan la inclusión de la robótica en educación**

Diversos estudios señalan que la robótica educativa es una herramienta ideal para iniciar a los estudiantes en el pensamiento computacional. Estos estudios destacan que la integración de la robótica en el currículo escolar no solo incrementa el interés y el compromiso del alumnado, sino que también mejora su capacidad para resolver problemas y comprender conceptos tecnológicos complejos. Algunos de estos estudios respaldan la efectividad de la robótica en la educación secundaria, demostrando su impacto positivo en el desarrollo de habilidades STEM (ciencia, tecnología, ingeniería y matemáticas).

Por ejemplo, el enfoque basado en proyectos de robótica mejora la enseñanza de la programación electrónica en los estudiantes. A través del desarrollo de diversas actividades, se busca estimular y motivar a los alumnos, ayudándolos a montar circuitos electrónicos de manera más efectiva y a programar las placas con mayor precisión. Este método se centra en la práctica real mediante la creación de "artilugios" que pueden ser utilizados en el entorno educativo, facilitando así un aprendizaje más práctico y aplicado. (Blas Padilla & Jaén Martínez, 2018)

Otro estudio indica que la utilización de las placas en la educación media superior resulta ventajosa para el estudiantado, ya que les facilita el aprendizaje de habilidades prácticas en programación y electrónica. Además, promueve el pensamiento crítico y la toma de decisiones, mejora la capacidad de resolver problemas y fomenta el trabajo en equipo. Asimismo, el uso de placas robóticas puede contribuir a reducir la brecha de género en los campos de la tecnología y la ingeniería. (Pérez Tavera, 2023)

Además, las propuestas educativas en robótica pueden desarrollar competencias específicas tanto ciudadanas como laborales, elevando a los estudiantes más allá del simple ensamblaje siguiendo instrucciones. Este enfoque fomenta la toma de decisiones basadas en el conocimiento y la comprensión científica de los dispositivos creados, logrando un aprendizaje completo de la robótica y una formación científica y tecnológica integral. Para fortalecer una comunidad docente en robótica, es esencial sistematizar y compartir experiencias, ayudando a los nuevos docentes a incorporar actividades en el aula, reformar planes de estudio y adaptar el currículo, aprovechando esta motivación para el aprendizaje estudiantil. (López Ramírez & Andrade Sosa, 2013)

Otro estudio nos indica que, a pesar de ser una buena práctica, la robótica no está tan arraigada en los centros como debería, al menos por la baja cantidad de literatura publicada al respecto. “As indicated in the Introduction, Arduino is a didactic resource used at various educational levels. However, based on the scientific literature reviewed in this work, it could be stated that it is not a very widespread resource in the formal context of primary education, since according to the criteria applied, there are nine works that deal with the use of Arduino with students of this age”. (García-Tudela & Marín-Marín, 2023)

Debido a que debemos elegir un lenguaje para programar, también hemos realizado búsqueda de literatura sobre el programa Scratch, con las siguientes conclusiones: Scratch, basado en experiencias propias, es una herramienta eficaz para desarrollar el pensamiento lógico en niños y adolescentes. Su interfaz gráfica incentiva a los estudiantes a plantear soluciones a problemas interdisciplinarios, ofreciendo diversas alternativas y permitiendo pruebas de acierto y error para alcanzar la solución óptima. Además, Scratch complementa el proceso de enseñanza-aprendizaje, destacándose por su capacidad para fomentar y explotar el pensamiento algorítmico en individuos de todas las edades, desde niños y adolescentes hasta adultos. (Pérez-Tavera, 2019)

También se llega a la conclusión de que la implementación de una estrategia educativa utilizando el entorno de programación Scratch contribuye significativamente al desarrollo y fortalecimiento de las habilidades de pensamiento computacional mediante la enseñanza de conceptos básicos de programación dentro de la educación secundaria básica. (García Rodríguez, 2022)



### **3.2 Origen e historia de la robótica educativa**

La robótica en la educación se ha posicionado como uno de los ámbitos más destacados en la enseñanza contemporánea. No obstante, este sector ha experimentado una larga trayectoria de crecimiento y transformación a lo largo de los años.

La robótica educativa surge en la década de los 60, cuando un equipo de científicos e investigadores del MIT (Instituto Tecnológico de Massachusetts) propuso la creación de robots con propósitos pedagógicos. Entre los investigadores destacados se encontraban: Wally Feurzeig, Seymour Papert, Cynthia Solomon y Daniel Bobrow. El objetivo primordial era instruir a los niños y niñas en la interacción y programación de robots. Más adelante, se creó una colaboración con la empresa Lego para el desarrollo de robots con fines educativos.

#### Historia de la robótica educativa: Antecedentes

Los orígenes de los robots, concebidos como máquinas mecánicas destinadas a ejecutar acciones físicas, se remontan al siglo III a.C. Durante esta época, los babilonios construyeron "El Reloj de Agua".

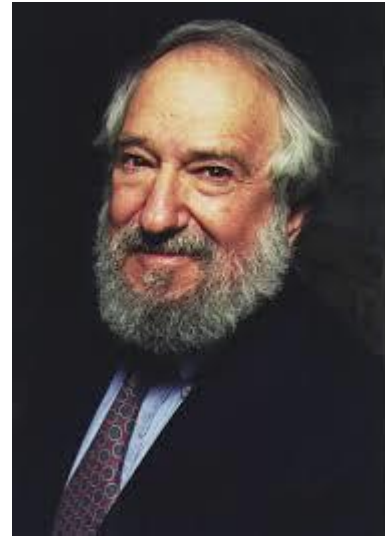
-En 1206, Al-Jazari creó diversos inventos, entre los cuales destacaba el primer robot autómatas. Con el tiempo, desarrolló otros modelos autómatas.

-En 1921, Karel Capek escribió la obra "Los robots universales de Rossum" (Rossum's Universal Robots), popularizando así el término "robot".

-En 1942, Isaac Asimov estableció las tres leyes de la robótica, las cuales tienen aplicación en el desarrollo ético de los robots.

-En 1948, George Devol diseñó el primer robot para tareas industriales, conocido como "Unimate". Este robot industrial marcó el inicio de la era moderna de la robótica.

El pionero de la robótica educativa podemos decir que es Seymour Papert. Reconocido como investigador, fundador y educador visionario, Papert fue el innovador detrás del primer robot educativo conocido como 'Turtle'. Papert abogaba por el aprendizaje constructor, una teoría educativa que enfatiza la participación activa de los estudiantes en la creación y construcción de sus propios proyectos y modelos. En 1968, publicó el influyente libro "Mindstorms: Children, Computers and Powerful Ideas", donde destacaba la importancia de integrar la robótica educativa en los programas de enseñanza. Por lo tanto, se atribuye a Papert el mérito de haber creado el campo de la robótica educativa.



*Seymour Papert*

### Lenguaje y primer robot

El nacimiento de la robótica educativa se inició con la creación del lenguaje de programación LOGO en 1967 por Wally Feurzeig, Seymour Papert y Cynthia Solomon. LOGO, derivado del término griego "logos" que significa "pensamiento", fue creado como un lenguaje de programación destinado a controlar robots. Este lenguaje fue diseñado con el propósito de enseñar los fundamentos y conceptos de la programación, utilizando una tortuga (Turtle) como modelo para la ejecución de comandos y la visualización de resultados.

El robot Turtle fue una de las primeras incursiones en la robótica educativa, desarrollado por Seymour Papert. Este robot, cuya creación se completó en 1969, tenía como objetivo principal la instrucción de adolescentes. Aunque el concepto inicial de los robots Turtle fue ideado por el matemático William Gray Walter en 1940, fue Papert quien refinó y perfeccionó este diseño para dar vida al primer robot educativo. Por tanto, los robots Turtle se destacan como los pioneros en dispositivos educativos STEM a nivel mundial.

### Desarrollo de la educación robótica

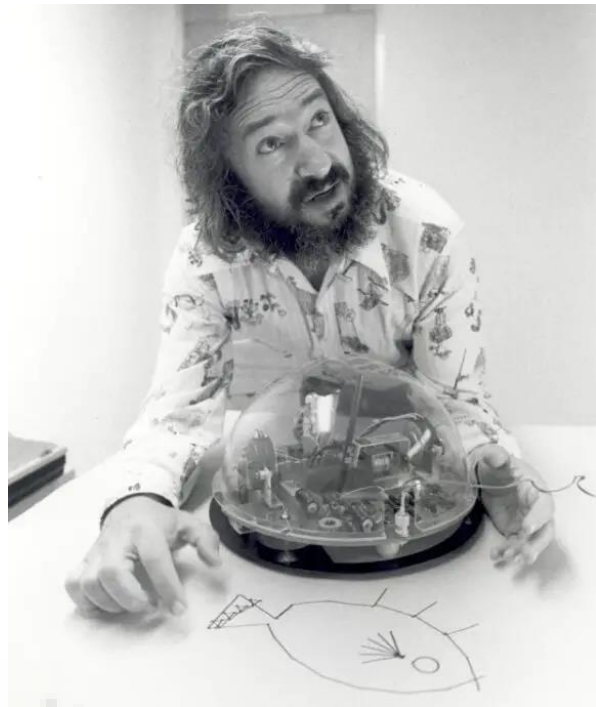
En 1980, la robótica educativa dio un salto hacia adelante al integrar el lenguaje Logo en los robots Turtle. Seymour Papert introdujo un robot Turtle equipado con ruedas, el cual era controlado mediante programación por computadora para su desplazamiento.



Más adelante, Papert amplió las capacidades del robot al añadir un bolígrafo, de esta manera se podía realizar dibujos a través de comandos programados.

Durante la década de los 90, la educación en robótica experimentó un crecimiento significativo a nivel global, marcando una nueva etapa en su desarrollo. En este período, surgieron dispositivos especialmente diseñados para fines educativos.

En Europa, se establecieron talleres, se desarrollaron materiales didácticos y se diseñaron actividades enfocadas en la enseñanza de la robótica en el nivel de educación primaria y secundaria. Este enfoque pedagógico buscaba integrar de manera efectiva el aprendizaje de la tecnología y la programación desde una edad temprana, fomentando así el pensamiento crítico y la resolución de problemas entre el estudiantado.



*Tortuga robot*

### Creación de Scratch

En 2007, la robótica educativa entró en una nueva era con el lanzamiento de Scratch. Desarrollado por el MIT, este lenguaje de programación visual fue diseñado específicamente para niños y niñas de entre 8 y 16 años, ofreciéndoles una forma innovadora de aprender a programar.

Scratch permitió a los jóvenes explorar conceptos de programación de manera creativa y accesible, a través de una interfaz gráfica intuitiva que les permitía arrastrar y soltar bloques de código para crear proyectos interactivos, juegos y animaciones. Esta herramienta no solo facilitó el aprendizaje de la programación, sino que también promovió la creatividad, el pensamiento lógico y la resolución de problemas entre los estudiantes de manera divertida y motivadora.



*Logo de Scratch*



Scratch permite a al estudiantado desarrollar programas de manera fácil sin la necesidad de tener grandes conocimientos de código. (Educación Robótica, 2023)

### Placas actuales

En la actualidad, el mercado ofrece una amplia variedad de kits robotizados o placas de programación destinadas tanto a estudiantes como a centros educativos. Con ellas se pueden realizar cantidad de proyectos y de propuestas para que el estudiantado desarrolle su creatividad y su pensamiento computacional. Esta diversidad de opciones brinda a los estudiantes la oportunidad única de experimentar con sus propias creaciones y explorar nuevos usos para la robótica educativa. A continuación, se realiza una recopilación de las más destacadas.

### **3.3 Arduino**

Arduino es algo más que una simple placa (de hecho, son varias, como se comprobará más adelante), es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso.

Esta placa contiene todos elementos necesarios para conectar periféricos a las entradas y salidas de un microcontrolador, y que puede ser programada tanto en Windows como macOS y GNU/Linux. Es un proyecto que promueve la filosofía de que la mejor manera de aprender es “cacharreando”.

La placa está basada en un microcontrolador ATMEL, de tal manera que en ellos se pueden grabar instrucciones. Estas instrucciones se escriben con el lenguaje de programación que se pueden utilizar en el entorno Arduino IDE. Por lo tanto, permiten crear programas que interactúan con los circuitos de la placa.

El microcontrolador de Arduino posee una interfaz de entrada, a la que se puede conectar en la placa diferentes tipos de periféricos. La información de estos periféricos conectados se trasladará al microcontrolador, el cual se encargará de procesar los datos que le lleguen a través de ellos.

El tipo de periféricos que se puede utilizar para enviar datos al microcontrolador es muy variado, y depende en gran medida de qué uso se le quiera dar. Se pueden usar cámaras para obtener imágenes, teclados para introducir datos, o muchos tipos de sensores.

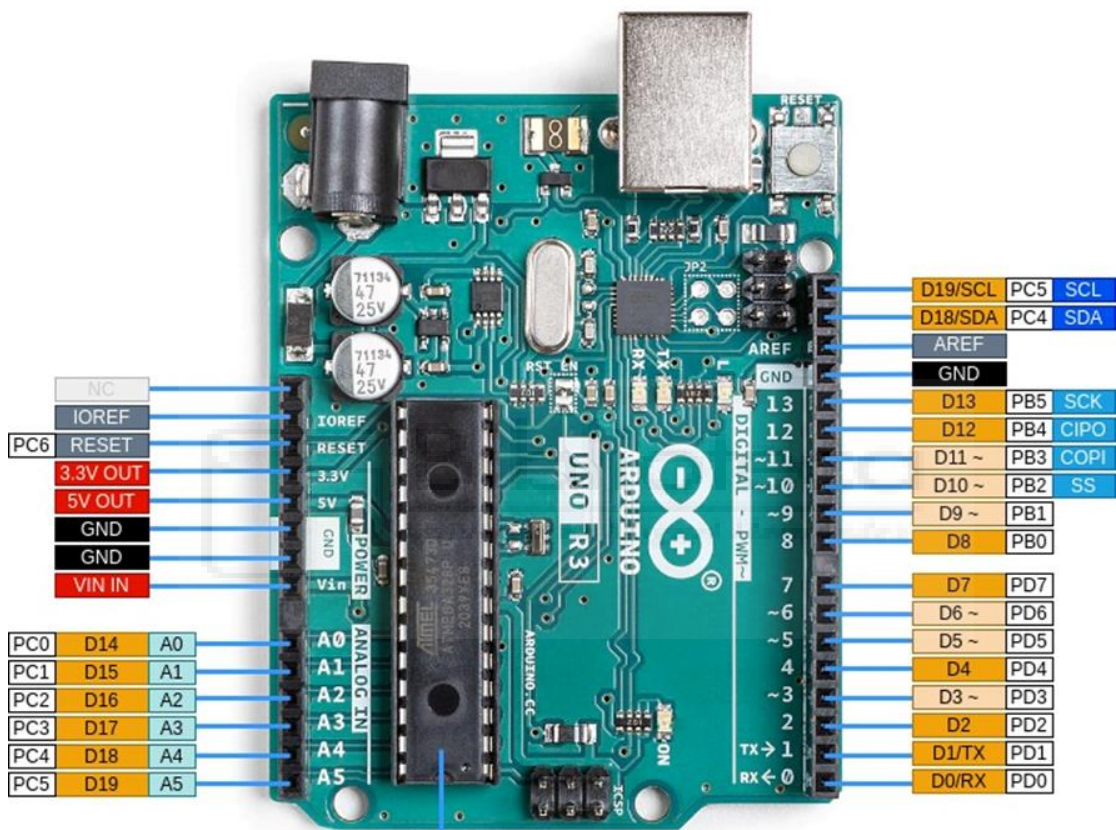
La placa también cuenta con una interfaz de salida, que se encarga de llevar la información que se ha procesado en el Arduino a otros periféricos. Estos

periféricos pueden ser pantallas o altavoces en los que reproducir los datos procesados, pero también pueden ser otras placas o controladores.

Como se ha comentado, Arduino tiene una gran variedad de placas, de distintas características según las necesidades del proyecto que se vaya a realizar. La más común y utilizada es la Arduino Uno, tanto es su modalidad de R3 como R4. A continuación, se muestra una tabla con las placas más utilizadas y sus distintas características. En este caso se verá más en profundidad el modelo Uno R3.

|                           | <b>UNO R3</b>  | <b>UNO R4</b>   | <b>LEONARDO</b>  | <b>ZERO</b>  |
|---------------------------|--|---|--|--|
| <b>Microcontrolador</b>   | ATmega328P   | Renesas RA4M1   | ATmega32u4   | ATSAMD21G18  |
| <b>Interfaz USB</b>       | USB-B  | USB-C   | Micro-USB  | Micro-USB  |
| <b>Pines</b>              | Digitales E/S: 14<br>PWM: 6<br>Analógicos entrada: 6   | Digitales E/S: 14<br>PWM: 6<br>Analógicos entrada: 6<br>DAC: 1                                    | Digitales E/S: 20<br>PWM: 7<br>Analógicos entrada: 12  | Digitales E/S: 20<br>PWM: 10<br>Analógicos entrada: 6<br>DAC: 1                                    |
| <b>Comunicación</b>       | UART<br>I2C<br>SPI   | UART<br>I2C<br>SPI<br>CAN   | UART<br>I2C<br>SPI   | UART<br>I2C<br>SPI   |
| <b>Alimentación</b>       | Voltaje de E/S: 5V<br>Voltaje nominal de entrada: 7-12V<br>CC por pin: 20mA<br>Conector CC coaxial | Voltaje de E/S: 5V<br>Voltaje nominal de entrada: 6-24V<br>CC por pin: 8mA<br>Conector CC coaxial | Voltaje de E/S: 5V<br>Voltaje nominal de entrada: 7-12V<br>CC por pin: 40mA<br>Conector CC coaxial | Voltaje de E/S: 3.3V<br>Voltaje nominal de entrada: 7-12V<br>CC por pin: 7mA<br>Conector CCcoaxial |
| <b>Velocidad de reloj</b> | Procesador principal: 16 MHz   | Procesador principal: 48 MHz  | Procesador principal: 16 MHz   | Procesador principal: 48 MHz   |
| <b>Memoria</b>            | 2KB SRAM<br>32KB FLASH<br>1KB EEPROM   | 32 kB RAM<br>256 kB FLASH   | 2.5KB SRAM<br>32KB FLASH<br>1KB EEPROM   | 32KB SRAM<br>256KB FLASH   |
| <b>Dimensiones</b>        | Peso: 25 g<br>Ancho: 53.4 mm<br>Longitud: 68,6 mm  | Peso: 25 g<br>Ancho: 68.85 mm<br>Longitud: 53.34 mm   | Peso: 25 g<br>Ancho: 68.6 mm<br>Longitud: 53.3 mm  | Peso: 12 g<br>Ancho: 53 mm<br>Longitud: 68 mm  |

En el siguiente esquema se pueden ver todos los pines de la placa Arduino UNO R3. Esta placa es la más popular entre las placas de Arduino, debido a su versatilidad y robustez, se utiliza ampliamente en proyectos electrónicos. Su uso y programación son relativamente fáciles de aprender y por ese motivo tiene mucho éxito en el ámbito educativo. Es una muy buena opción para comenzar en el mundo Arduino.



|                          |    |    |                        |
|--------------------------|----|----|------------------------|
| (PCINT14/RESET) PC6      | 1  | 28 | PC5 (ADC5/SCL/PCINT13) |
| (PCINT16/RXD) PD0        | 2  | 27 | PC4 (ADC4/SDA/PCINT12) |
| (PCINT17/TXD) PD1        | 3  | 26 | PC3 (ADC3/PCINT11)     |
| (PCINT18/INT0) PD2       | 4  | 25 | PC2 (ADC2/PCINT10)     |
| (PCINT19/OC2B/INT1) PD3  | 5  | 24 | PC1 (ADC1/PCINT9)      |
| (PCINT20/XCK/T0) PD4     | 6  | 23 | PC0 (ADC0/PCINT8)      |
| VCC                      | 7  | 22 | GND                    |
| GND                      | 8  | 21 | AREF                   |
| (PCINT6/XTAL1/TOSC1) PB6 | 9  | 20 | AVCC                   |
| (PCINT7/XTAL2/TOSC2) PB7 | 10 | 19 | PB5 (SCK/PCINT5)       |
| (PCINT21/OC0B/T1) PD5    | 11 | 18 | PB4 (MISO/PCINT4)      |
| (PCINT22/OC0A/AIN0) PD6  | 12 | 17 | PB3 (MOSI/OC2A/PCINT3) |
| (PCINT23/AIN1) PD7       | 13 | 16 | PB2 (SS/OC1B/PCINT2)   |
| (PCINT0/CLKO/ICP1) PB0   | 14 | 15 | PB1 (OC1A/PCINT1)      |

|                    |
|--------------------|
| ■ Digital          |
| ■ Digital PWM      |
| □ Microcontrolador |
| ■ SPI              |
| ■ I2C              |
| ■ Analógico        |
| ■ Power            |
| ■ Tierra           |

Componentes de la placa Arduino

Programación de la placa Arduino:

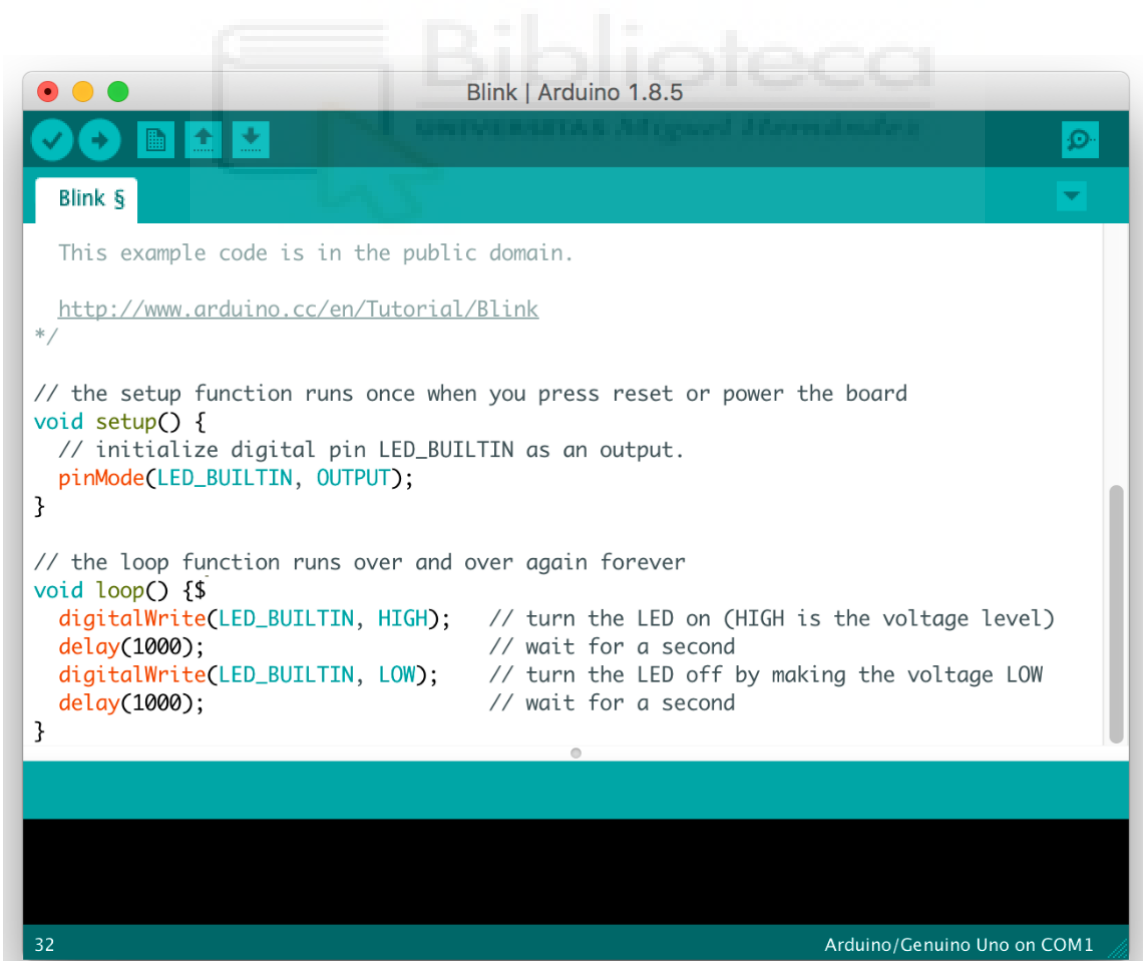
El Arduino UNO R3 se programa con el lenguaje de Arduino basado en C++ utilizando alguno de los IDEs de Arduino.

Existen 3 IDEs con los que se puede programar Sketch para el Arduino UNO R3:

**Arduino IDE:** Es una aplicación de escritorio disponible para todos los sistemas operativos. Es muy fácil de instalar y sencillo de usar, con lo cual es ideal si se está comenzando a trabajar con esta plataforma

**Arduino CLI:** Es un cliente de línea de comandos. Esta opción es para usuarios un poco más avanzados que estén familiarizados con el uso de comandos del sistema operativo.

**Editor Web:** Es el IDE cloud de Arduino. Requiere realizar un registro con lo cual no sería el más indicado en educación. (Fernández Y. , 2022) (Para Arduino, 2024)



The image shows a screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.8.5". The main window displays the following code:

```
This example code is in the public domain.

http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {$
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

At the bottom left, the page number "32" is visible. At the bottom right, the text "Arduino/Genuino Uno on COM1" is displayed.

Programación con Arduino

### 3.4 Raspberry Pi

La Raspberry Pi es un ordenador muy económico y con un tamaño compacto y pequeño, que cabe prácticamente en una mano. Puede ser conectado a un monitor de ordenador o a una pantalla de TV, y puede usarse con un ratón y teclado estándar igual que un ordenador normal. Este pequeño ordenador funciona con el sistema operativo Linux y permitirlas a sus usuarios de todas las edades explorar la informática y aprender a programar lenguajes como Scratch y Python.

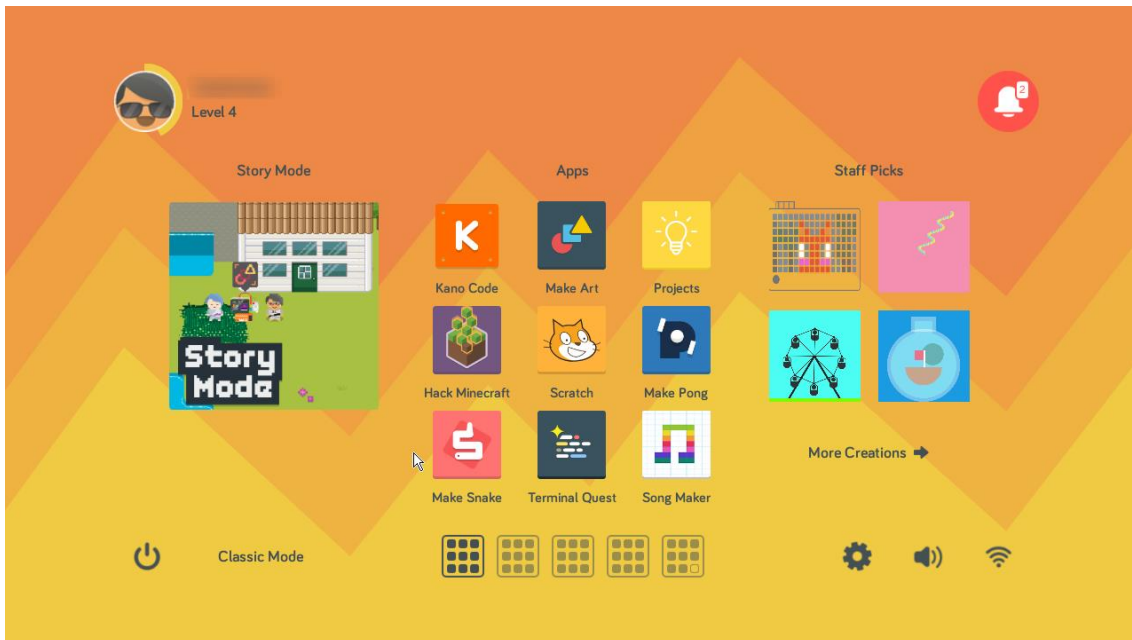
Es capaz de realizar la mayoría de las tareas típicas de un ordenador de escritorio, desde navegar en internet, reproducir videos en alta resolución, manipular documentos de ofimática, o incluso reproducir juegos. Además, y aquí entra quizá la parte más interesante para nuestro estudio, la Raspberry Pi tiene es capaz de interactuar con el mundo exterior, puede ser usada en una amplia variedad de proyectos digitales, desde reproductores de música y video, estaciones meteorológicas hasta cajas de aves con cámaras infrarrojas. Existen infinidad de proyectos educativos que se pueden realizar con esta herramienta. Con lo que la convierte en una gran opción para aprender a programar, entender cómo funcionan los ordenadores e incluso iniciarse en proyectos robóticos.

La Raspberry Pi fue creada en febrero del 2012 por la Raspberry Pi Foundation, en principio para promover en los institutos y universidades del Reino Unido el uso y la enseñanza de las ciencias de la informática. Al poco tiempo ya se había creado una comunidad de miles de personas desarrollando y experimentando con proyectos tras haber adquirido una.

En definitiva, nos encontramos quizá ante el ordenador perfecto para la educación: económico, versátil, y con cientos de opciones para desarrollar proyectos. A todo ello podemos sumar la opción de instalar KANO OS.

KANO OS es un sistema creado para Raspberry Pi especialmente enfocado a educación. Se puede definir como un sistema operativo de código abierto para la exploración, la creación y el juego.

Cuando se instala y se arranca KANO OS por primera vez nos encontraremos ante un sistema operativo especialmente diseñado para alumnos jóvenes, con una interfaz atractiva basada en el color y muy sencilla de utilizar. Contiene una importante base de gamificación para llamar la atención del alumnado y así descubrir sus posibilidades.



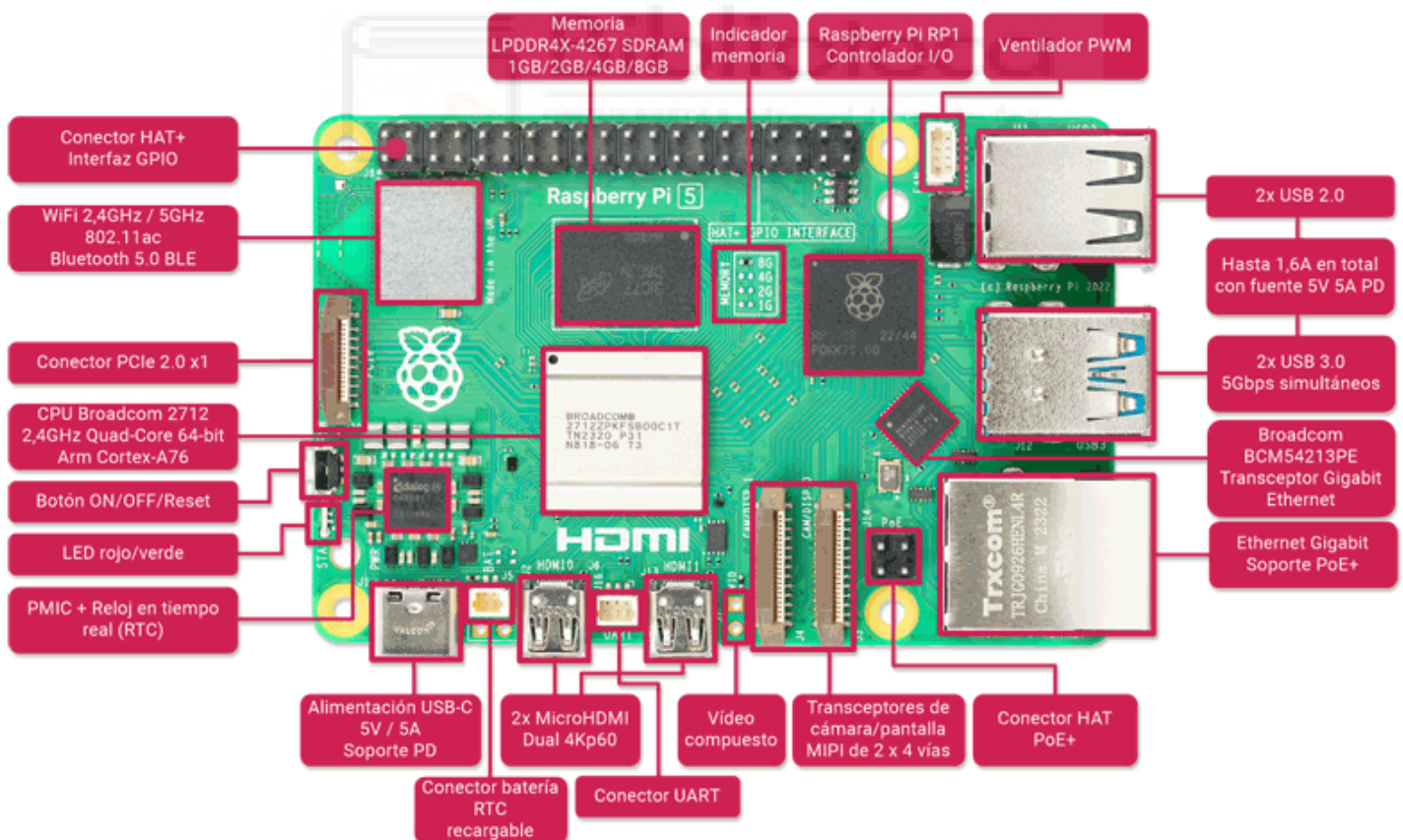
Sistema Operativo KANO OS

Igual que ocurre con las placas Arduino, Raspberry Pi también tiene varias versiones, en la actualidad podemos encontrar estas.

- **Raspberry Pi 5.** Es el modelo más potente y actual e incluye un procesador Broadcom BCM2712, 4 u 8 GB de memoria RAM, doble salida HDMI, Bluetooth 5.0, WiFi 802.11ac de doble banda, 4 puertos USB (2 USB 3.0 y 2 USB 2.0) y Ethernet (con adaptador USB).
- **Raspberry Pi 4 Model B.** Cuenta con un procesador Broadcom BCM2711 y ofrece versiones de 2, 4 y 8 GB de memoria RAM. Tiene dos puertos USB 3.0, dos USB 2.0 y dos micro-HDMI que soportan 4K, entre otras características.
- **Raspberry Pi 3 Model A+.** Procesador Broadcom BCM2837B0 y 1 GB de RAM LPDDR2 que el Modelo B+. Un puerto USB 2.
- **Raspberry Pi 3 Model B+.** Procesador Broadcom BCM2837B0, 1 GB de memoria RAM, WiFi 802.11ac de doble banda, Bluetooth 4.2, HDMI, cuatro puertos USB 2.0 y Gigabit Ethernet (a través del USB 2.0).
- **Raspberry Pi 3 Model B.** Procesador Broadcom BCM2837, 1 GB de memoria RAM, conectividad WiFi 802.11n, Bluetooth 4.1, HDMI y cuatro puertos USB 2.0.

- **Raspberry Pi Zero y Zero W.** Son las versiones más pequeñas y económicas: el Zero tiene un procesador de un solo núcleo y no tiene conectividad inalámbrica, mientras que el Zero W añade WiFi y Bluetooth. Ambos tienen 512 MB de RAM.
- **Raspberry Pi 400.** Es una Raspberry Pi 4 integrada en un teclado compacto. Tiene especificaciones similares a dicha placa, pero viene integrada en un teclado para crear una solución todo en uno a la que sólo le falta conectar un monitor.
- **Raspberry Pi Pico:** Es una placa de microcontrolador basada en el chip RP2040 de Raspberry Pi. Está diseñada para proyectos de electrónica.

En este caso vamos a profundizar sobre el último modelo, la **Raspberry Pi 5**, mostrando sus partes principales y sus características. (Educación 3.0, sf) (Linares, 2023) (Pastor, 2023)



*Características de una Raspberry Pi*



| <b>Raspberry Pi 5</b> |   |
|-----------------------|---|
| <b>CPU</b>            | Broadcom BCM2712<br>Quad-core Arm Cortex-A76 a 2,4 GHz  |
| <b>GPU</b>            | VideoCore VII<br>Soporta OpenGL ES 3.1 y Vulkan 1.2   |
| <b>RAM</b>            | 4 / 8 GB LPDDR4X  |
| <b>CONECTIVIDAD</b>   | Wi-Fi 5<br>Bluetooth 5.0 / BLE  |
| <b>PUERTOS</b>        | 2 x micro HDMI (hasta 2 x 4K 60Hz simultáneas)<br>2 x USB 3.0<br>2 x USB 2.0<br>1 x Gigabit Ethernet con PoE opcional<br>2 x MIPI de 4 pistas<br>1 x PCIe 2.0 x1<br>1 x GPIO 40 pines |
| <b>ALMACENAMIENTO</b> | Ranura microSD<br>Opción para unidades SSD M.2 (vía HAT opcional)   |
| <b>OTROS</b>          | Botón de encendido y apagado<br>RTC (Real Time Clock)<br>Accesorios opcionales diversos   |

### 3.5 Micro:bit

Micro:bit es una pequeña placa programable con unas dimensiones que no llegan a los 4x5 cm por lo que es fácilmente transportable.

Está diseñada pensando en facilitar tanto su programación como las conexiones de las que dispone, pudiendo usar latiguillos cocodrilo, y un lenguaje de programación por bloques. Por lo tanto, se convierte en una herramienta muy atractiva para acercar la robótica y la programación al alumnado incluso desde la educación primaria.

Debido a la amplia gama de sensores y placas adicionales de fácil conexión que existen, permiten la creación de cientos de proyectos abiertos a la imaginación y creatividad del alumnado.

Micro:bit forma parte de la comunidad open source y open hardware, por lo que es un material en continua evolución con una amplia comunidad online donde encontrar ideas para trabajar.

Tiene un entorno de programación gráfico propio: MakeCode de Microsoft, un sencillo editor gráfico online muy potente y gratuito que posibilita introducirnos en el mundo de la programación de forma intuitiva a través del lenguaje de programación visual o de bloques. Con él se puede aprender a pensar como un programador sin caer en los molestos errores de sintaxis.

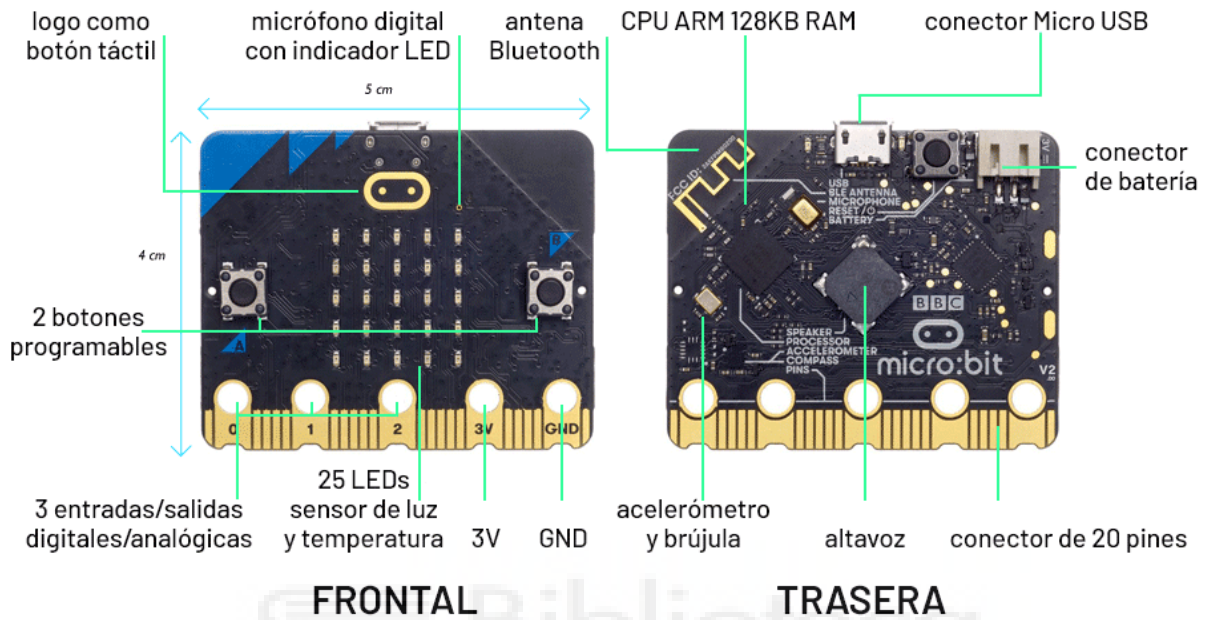
Pero para un alumnado más avanzado Micro:bit también puede ser una gran herramienta, ya que se puede programar con JavaScript, Python y Scratch (añadiendo una extensión).

En 2015 se creó la primera versión de Micro:bit, pero vamos a centrarnos en el segundo modelo mejorado, el cual incorpora micrófono, altavoz, sensor táctil y un procesador más potente.

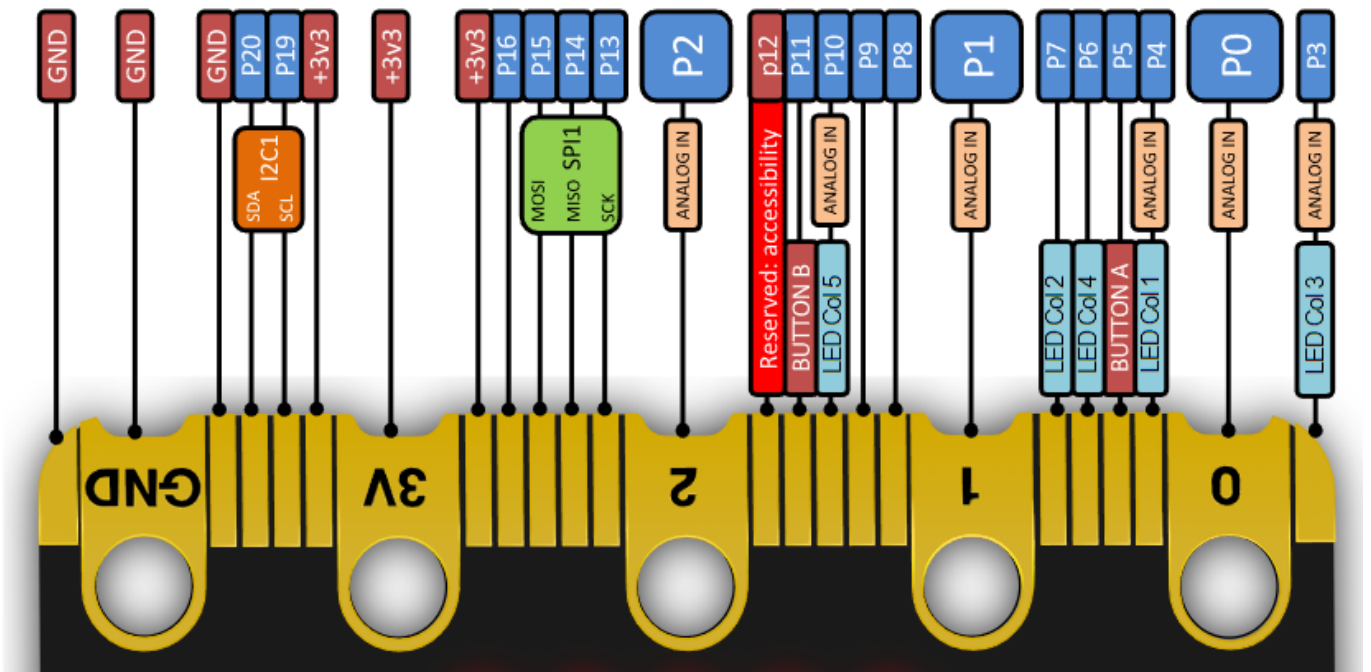
En este caso, la tarjeta micro:bit dispone de:

- 25 LEDs programables individualmente
- 3 botones programables
- Altavoz
- Micrófono
- Pines de entrada y salida
- Sensor de Luz y Temperatura
- Sensores de movimiento (acelerómetro y brújula)
- Comunicación inalámbrica, vía Radio y Bluetooth
- USB y Conector para batería externa

La placa también contiene 25 LEDs programables individualmente que permiten mostrar texto, números e imágenes.

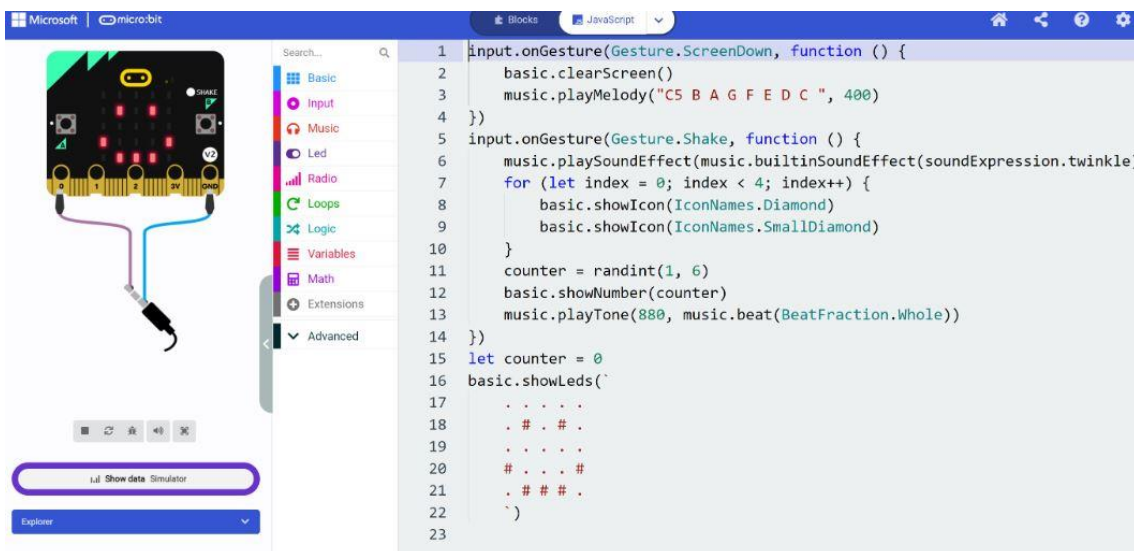
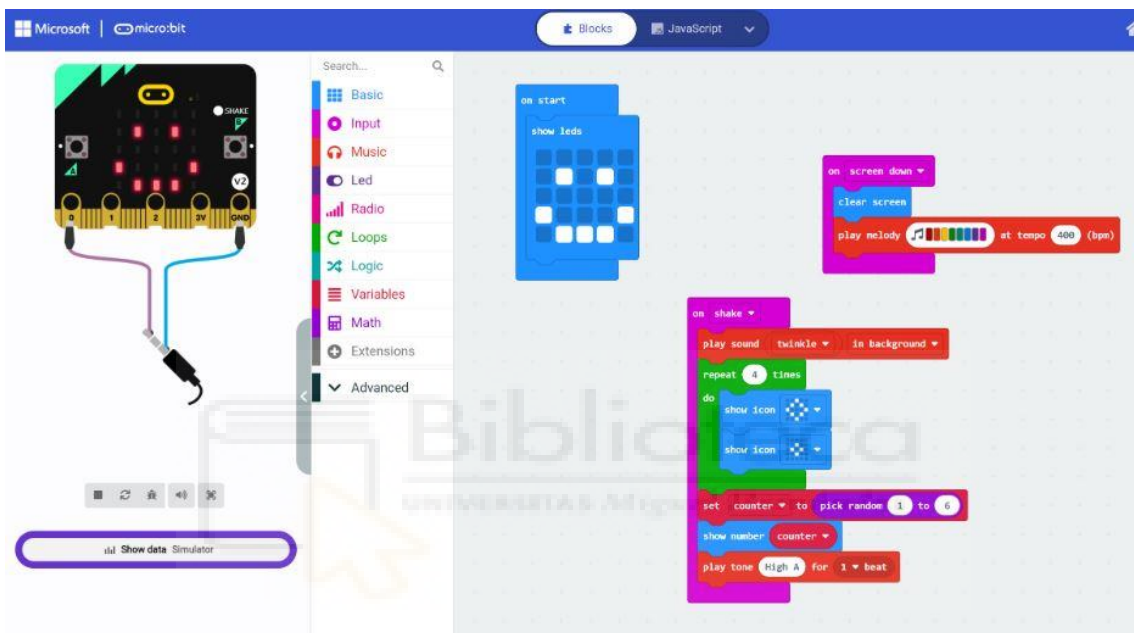


Las conexiones de expansión de la Micro:bit están dispuestas de la siguiente manera:



Conexiones de expansión de la placa Micro:bit

Como se ha comentado, con esta placa se puede programar de diversos modos, pero el editor MakeCode de Microsoft es la forma perfecta de comenzar a practicar y crear con Micro:bit. Los bloques con código de colores resultan familiares para cualquiera que esté familiarizado con Scratch, y lo suficientemente potentes como para acceder a todas las funciones de este pequeño ordenador. Además, permite cambiar a modo JavaScript para ver el código de texto equivalente a los bloques.



Programación por bloques y por comandos

(micro:es, s.f.) (micro log, s.f.)

### 3.6 Echidna

Es una placa 100% autónoma basada en Arduino, pero con unas ventajas específicas que se podrán comprobar más adelante. Nace como proyecto de la asociación Echidna Educación, que es una asociación sin ánimo de lucro que tiene como objetivos:

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

Echidna nace de la idea de 5 docentes españoles que deciden crear su propia herramienta para trabajar con robótica en el aula usando hardware y software libre. La placa programable tanto con Scratch (incluso personalizado para esta placa: EchidnaScratch) o con Snap4Arduino.

- 3 Leds (verde, rojo, amarillo)
- 1 Led RGB
- 2 Pulsadores
- 1 Zumbador
- 1 Joystick
- 1 LDR
- 1 Sensor de temperatura
- 1 Acelerómetro
- 1 Micrófono
- 8 Conexiones Makekey makekey
- 1 Conexión bluetooth
- 3 E/S para motores y sensores
- 1 E/S analógica

Complementos: Estos son algunos de los componentes que se pueden usar para complementar las funcionalidades de Echidna, para conectarlos se disponen de las I/O y la entrada analógica IN

- Servomotor posición
- Servomotor continuo
- Infrarrojos distancia

En esta imagen podemos encontrar los componentes de Echidna.



*Componentes de la placa Echidna*

(micro log, s.f.)

Esta placa tiene una gran ventaja en relación con Scratch con respecto a las demás, y es que sus creadores diseñaron EchidnaScratch. Esta plataforma es una modificación de Scratch que proporciona nuevos bloques para programar las placas Echidna (servos, entradas digitales, Joystick, sensores, micrófonos...) y nos permite trabajar con un módulo de inteligencia artificial. Se accede a través de un [enlace](#) concreto (no desde el Scratch original).



*Bloques que añade Echidna a Scratch*

En la imagen de la izquierda podemos apreciar el bloque que se genera en Scratch cuando accedemos a través de EchidnaScratch. También se generan bloques por si se desea trabajar con Machine Learning, que es la inteligencia artificial creada por el proyecto Echidna, pero es este caso no profundizaremos en ella.

Nos permite trabajar de manera directa sobre los leds rojo, verde y amarillo, sobre los pines digitales, los servos que podamos conectar a la placa, las diversas entradas analógicas y digitales, el acelerómetro, el Joystick, el sensor de luz, de temperatura, lectura del micrófono y las entradas Makey Makey.

Esta adaptación de Scratch que nos ofrecen los creadores de la placa Echidna la hacen muy apetecible a la hora de seleccionarla para poder trabajar con el alumnado, ya que reduce considerablemente las labores tanto a nivel de hardware como de software.

(Echidna Educación, 2024)

### 3.7 Tipos de Sensores y actuadores más importantes.

Tanto los sensores como los actuadores desempeñan un papel muy importante en el funcionamiento y la utilidad de las placas robóticas en el ámbito educativo. Estos componentes permiten al estudiantado interactuar con su entorno, recopilar datos y realizar acciones físicas, lo que contribuye a una comprensión más profunda de los conceptos de robótica y pensamiento computacional.

Se puede definir que los sensores actúan como los "sentidos" de un supuesto robot, permitiéndole percibir y responder a su entorno, mientras que los actuadores podrían ser los "músculos" u "órganos", que ejecutan las acciones físicas basadas en las decisiones tomadas por el sistema. En este sentido, comprender la función y la aplicación de sensores y actuadores es básico para el diseño, la programación y la operación efectiva de robots y otros dispositivos electrónicos controlados por placas robóticas.

Por lo tanto, vale la pena presentar algunos de los sensores y actuadores más importantes utilizados en el ámbito educativo, destacando sus consideraciones más relevantes para su integración en actividades de enseñanza y aprendizaje.

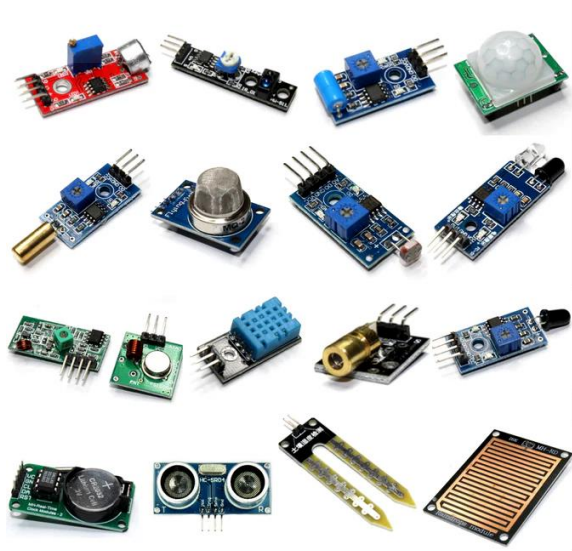
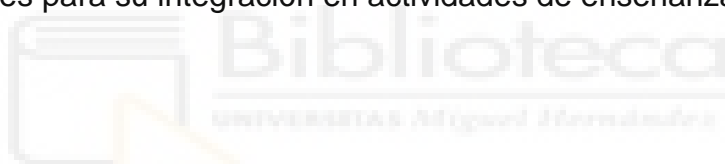
#### Sensores

Un sensor es un dispositivo tecnológico que tiene la capacidad de percibir ciertos estímulos del exterior y transformarlos en impulsos eléctricos, que pueden ser interpretados por ordenadores u otras máquinas.

Básicamente, se puede definir un sensor como un traductor que puede explicar en un lenguaje común para las máquinas cualquier propiedad del ambiente (física, química, sonora, lumínica...).

Las lecturas de un sensor se pueden usar para medir variaciones en las condiciones de una determinada área, accionando respuestas automáticas de otros dispositivos (actuadores) como consecuencia: activar o desactivar

luces/sonidos, puertas, controles de riego, etc.



*Tipos de sensores*



## **Tipos de sensores y sus características:**

Según el tipo de variable que pueda percibir e interpretar un sensor, se pueden definir varios tipos. Cada uno de ellos tiene propiedades únicas que los hacen útiles en escenarios específicos. A continuación, citamos los más destacados.

### Sensores de distancia/proximidad

Los sensores de proximidad son dispositivos que facilitan la detección de la presencia de un objeto y su distancia respecto a un punto de referencia, generalmente el propio sensor. Por lo general, operan mediante un par de componentes: un emisor y un receptor. El emisor emite una señal periódicamente, mientras que el receptor detecta el rebote de la señal, lo que indica la proximidad del objeto.

Estos sensores tienen la capacidad de proporcionar una amplia gama de datos relevantes sobre los objetos que detectan dentro de su área de cobertura.

### Sensores de sonido

Los dispositivos de detección de sonido captan las vibraciones acústicas presentes en el entorno, las cuales son el resultado de las fluctuaciones de presión del aire. Dependiendo de los niveles de intensidad para los que estén configurados, transforman estas perturbaciones en señales eléctricas. Estos sensores suelen ser compactos en comparación con otros tipos y tienen diversas aplicaciones prácticas en la tecnología moderna, como reconocimiento de voz, sistemas de vigilancia inteligente y medición de la potencia de trabajo de equipos.

### Sensores de luz

Un sensor de luz es un dispositivo capaz de percibir la luz ambiental (o la que se origina de un punto en concreto) y luego reaccionar a ella con un impulso eléctrico que varía dependiendo de la intensidad de la luz que ha detectado. A mayor lectura de luz, mayor intensidad en la respuesta eléctrica.

Este tipo de sensores también pueden estabilizarse para reaccionar ante determinadas intensidades, por lo que se les conocen también como sensores de luminosidad, dado que reaccionan a las variaciones de luz ambiental.

Estos sensores son especialmente útiles para la regulación de consumo de energía en espacios controlados porque pueden apagar o encender las luces según sea necesario, por ejemplo.

### Sensores de temperatura

Los dispositivos de medición de temperatura son instrumentos que permiten evaluar la disparidad de calor entre un punto establecido como referencia y el entorno que se está analizando. Estos dispositivos transforman esta información en señales eléctricas, lo que facilita la toma de decisiones basadas en los niveles de temperatura que se consideren adecuados.

Por lo general, los sensores de temperatura utilizan resistencias en su funcionamiento, aunque existen diversas metodologías para su operación.

### Sensores de humedad

Los dispositivos de detección de humedad posibilitan la evaluación tanto de la temperatura como de la cantidad de humedad relativa presente en el aire en un espacio específico. Estos datos se convierten habitualmente en señales eléctricas que luego pueden ser utilizadas para activar otros sistemas.

Además, en el ámbito educativo, son comunes los sensores de humedad del suelo, los cuales permiten llevar a cabo proyectos como estaciones de riego automáticas. En este caso, según el nivel de humedad detectado, se puede activar automáticamente el riego de las plantas.

### **Actuadores**

Los actuadores son componentes mecánicos que convierten la energía en movimiento. Básicamente, los actuadores son los motores que permiten a los robots moverse y realizar acciones tras recibir las órdenes pertinentes.

### **Tipos de actuadores y sus características:**

Existe una gran variedad de actuadores, pero vamos a reducirlos a tres tipos de actuación:

- Neumático
- Hidráulico
- Eléctrico

Se va a destacar los actuadores que consideramos más interesantes y fáciles de implementar en educación, sin profundizar en todos los tipos de actuadores que hay. Se destacan los siguientes.



*Tipos de actuadores*

Los servomotores son ampliamente utilizados en robótica debido a sus excelentes características de control de posición y velocidad, así como su capacidad para proporcionar un alto torque en un tamaño compacto.

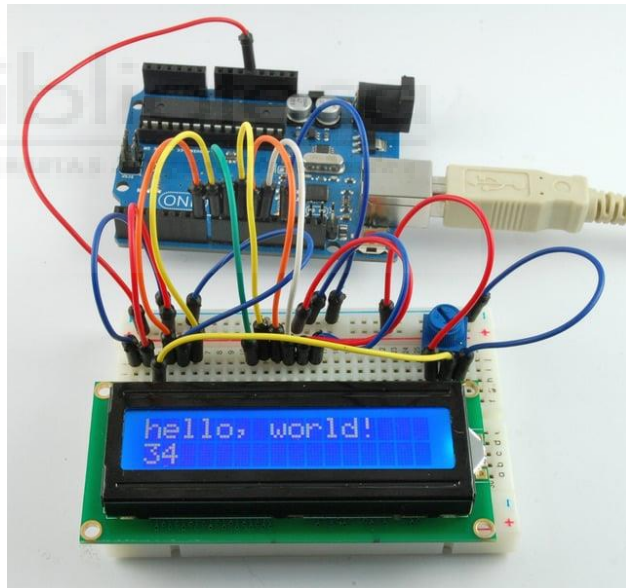
Los motores eléctricos son actuadores electromecánicos que producen movimiento al convertir la energía eléctrica en energía mecánica. Estos actuadores son la opción preferida en robótica

## Periféricos

No se debe dejar en el olvido otros componentes como son los periféricos. Podemos definirlos como el dispositivo auxiliar e independiente conectado a la unidad central de procesamiento o en este caso a nuestra placa. Se consideran periféricos a las unidades o dispositivos de hardware a través de los cuales la placa se comunica con el exterior, y también a los sistemas que almacenan o archivan la información, sirviendo de memoria auxiliar de la memoria principal. Pueden ser de entrada o salida

Ejemplos de periféricos:

- Pantallas LCD
- Teclados
- Memorias externas
- Cámaras
- Micrófonos
- Impresoras
- Pantalla táctil
- Displays numéricos
- Zumbadores
- Altavoces
- Indicadores luminosos



*Periféricos*

(EDS Robotics, 2022) (Crespo, 2020)

### 3.8 Scratch

#### ¿Qué es?

Scratch es un entorno gráfico de programación con fines educativos. Permite crear simulaciones, juegos o animaciones de forma sencilla y accesible gracias al ensamblaje de bloques. Scratch es un lenguaje de programación, un entorno de desarrollo y un sitio web.

Como primera comunidad de programación gratuita para niños, Scratch es una potente herramienta para enseñar el mundo de la informática. También ayuda a desarrollar la lógica y la reflexión del alumnado en aspectos básicos de programación como bucles, pruebas y tareas.

Para programar con Scratch se usa su software, su entorno y su lenguaje de programación visual. A través de su interfaz en línea o sin conexión, permite a los usuarios crear un algoritmo ensamblando bloques para crear una simulación o una historia interactiva.

#### Ventajas

Las principales ventajas del lenguaje y entorno Scratch son su facilidad de uso y su accesibilidad. Realiza una gran labor de enseñanza y educación de la lógica y la reflexión que hay que tener para programar, y es completamente gratuito.

La herramienta funciona a través de un editor visual disponible en más de 70 idiomas, incluido el español. Con ladrillos de colores, implementa conceptos básicos de programación como bucles, condiciones, pruebas y asignaciones.

Además, Scratch es una forma divertida y educativa para que los docentes desarrollen la lógica del alumnado y transmitan conocimientos sobre programación. El aspecto gráfico lo hace atractivo de utilizar. El programa también permite transformar los proyectos en aplicaciones web, para poder utilizarlos en línea.



*Ejemplo de bloques en Scratch*

### Uso en educación

Aunque su uso va más allá de los colegios, Scratch es utilizado por una gran comunidad de docencia y alumnado. La codificación y la programación forman ya parte de algunos planes de estudios de secundaria. Por tanto, el profesorado puede utilizar esta plataforma para sus clases.

Este lenguaje de programación está diseñado específicamente para jóvenes de 8 a 16 años. Por lo tanto, el alumnado puede aprender en todos los niveles, desde primaria hasta prácticamente la universidad. Ayuda en diferentes asignaturas como matemáticas o tecnología. Los usuarios aprenden conceptos y lógica para resolver problemas, diseñar proyectos y dar vida a su creatividad e imaginación.

En definitiva, es una herramienta ideal para iniciarse en la programación en edades tempranas, ayuda a los usuarios a desarrollar las habilidades lógicas y de pensamiento computacional necesarias para construir proyectos.

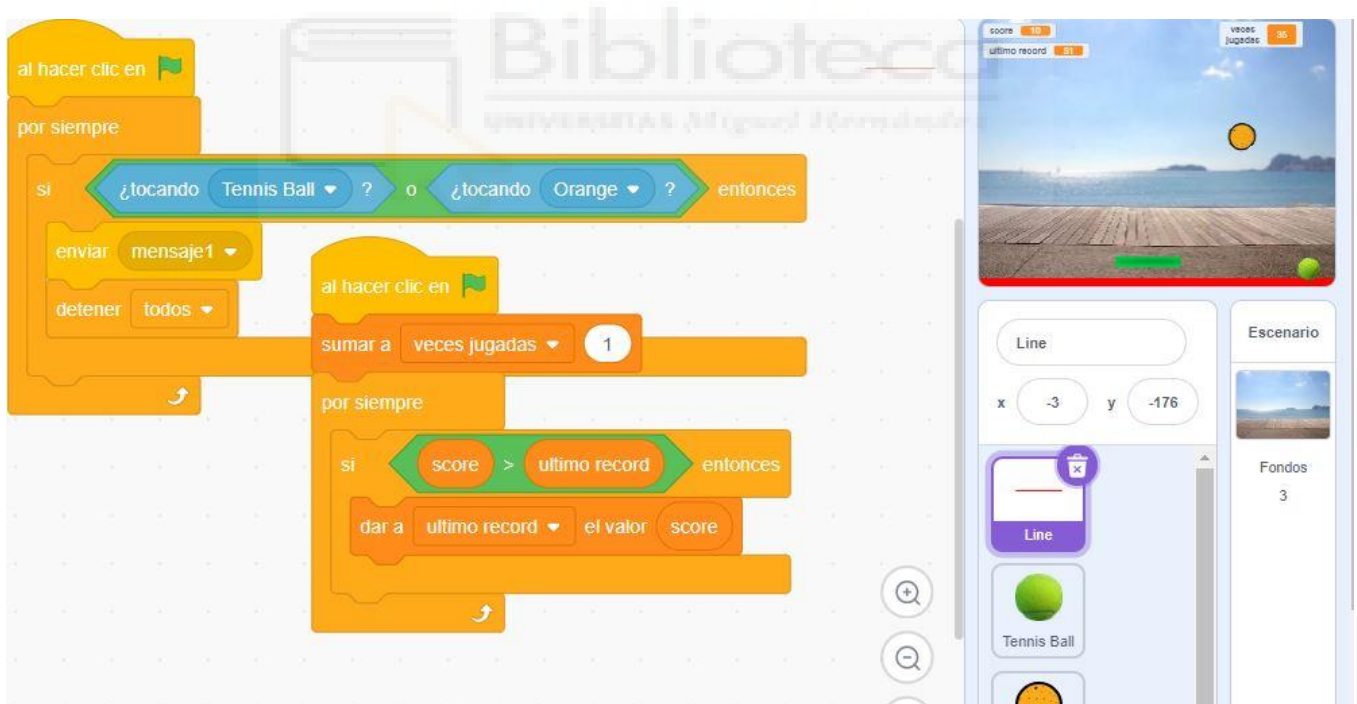


Imagen de Scratch

(DataScientest, 2024)

#### **4. Propuesta.**

Una vez demostrado que la robótica es un campo ideal para introducirlo en nuestro currículo de Tecnología y realizado el análisis de las placas robóticas más relevantes para la educación, hemos tomado la decisión de utilizar la placa Echidna en combinación con el lenguaje de programación Scratch para nuestro trabajo con el alumnado. Esta elección se fundamenta en la convicción de que esta combinación ofrece el entorno ideal para implementar la robótica de manera efectiva en el proceso de aprendizaje de los estudiantes.

La placa Echidna presenta características versátiles que la hacen adecuada para una amplia gama de proyectos educativos. Además, el hecho de que sus desarrolladores hayan creado un entorno en Scratch exclusivo para Echidna, permite trabajar con nuevos bloques y usar los sensores y actuadores de la propia placa, lo cual es una gran ventaja.

Como no olvidamos que el trabajo está enfocado en un aula de tecnología, se le va a dar a la propuesta de aprendizaje un contexto donde el alumnado pueda crear, manipular y crear también con sus propias manos. La idea es que construyan una marioneta, y sobre ella se colocará la placa como “cerebro”, recibiendo esta misma las interacciones en los sensores y que harán reaccionar a la marioneta gracias a los actuadores.

En concreto, vamos a trabajar con los sensores de luz, proximidad y con el micrófono del ordenador haciendo las veces de sensor de sonido. Como actuadores para hacer que la marioneta reaccione cuando los sensores lo ordenen, usaremos 3 servomotores.

Se realizará el montaje manual de la marioneta, a la cual se le conectará un servomotor a cada uno de los brazos y el tercero a la cintura mediante hilos. Los servomotores estarán colocados en la cruceta superior que sustenta la marioneta. Cuando los servos se activen, girarán un determinado número de grados para hacer que la marioneta levante los brazos/cintura como reacción.

En la parte superior de la marioneta, en el centro de la cruceta, se colocará la placa Echidna, a la cual se le conectará el sensor de proximidad, se usará el sensor de luz que trae la propia placa y se usará el micrófono del propio ordenador para detectar el sonido. Se programará la placa con scratch de tal manera que tanto al detectar un objeto a una distancia determinada, una luz, o un sonido de determinada intensidad, los servomotores se activen y la marioneta reaccione moviéndose de una determinada manera. Además de la reacción del movimiento de la marioneta, también se puede programar que se emita algún sonido/grabación cuando se activen los servomotores y añadir sonido a ese movimiento.

#### 4.1 Construcción marioneta

Como se ha comentado en la propuesta, se va a contextualizar la actividad sobre una marioneta que hará las veces de humano. Recibirá ciertos estímulos a través de los sensores y se activarán los servos para que mueva partes de su cuerpo. Para poder llevar a cabo el movimiento, las partes de la marioneta deben pesar lo menos posible, y debe ser un material muy económico.

En ese aspecto los tubos de espuma de polietileno reticulado (churros de piscinas) son un material ideal debido a su precio económico, su poco peso y su fácil manipulación ya que se corta con facilidad. Para simular la cabeza de la marioneta, el material ideal es una esfera de poliestireno de unos 20 cm de diámetro.

Se cortan y se unen las partes/extremidades con cinta adhesiva/americana, dando forma final a la marioneta.

Para sustentarla, se crea una cruz con maderas, de la cual saldrá un hilo hacia la cabeza la de la misma como fijación principal, y tres hilos que conectados desde los servos a brazos/cadera para darle el movimiento. La cruz a su vez estará sostenida por un trípode en la parte trasera del montaje.



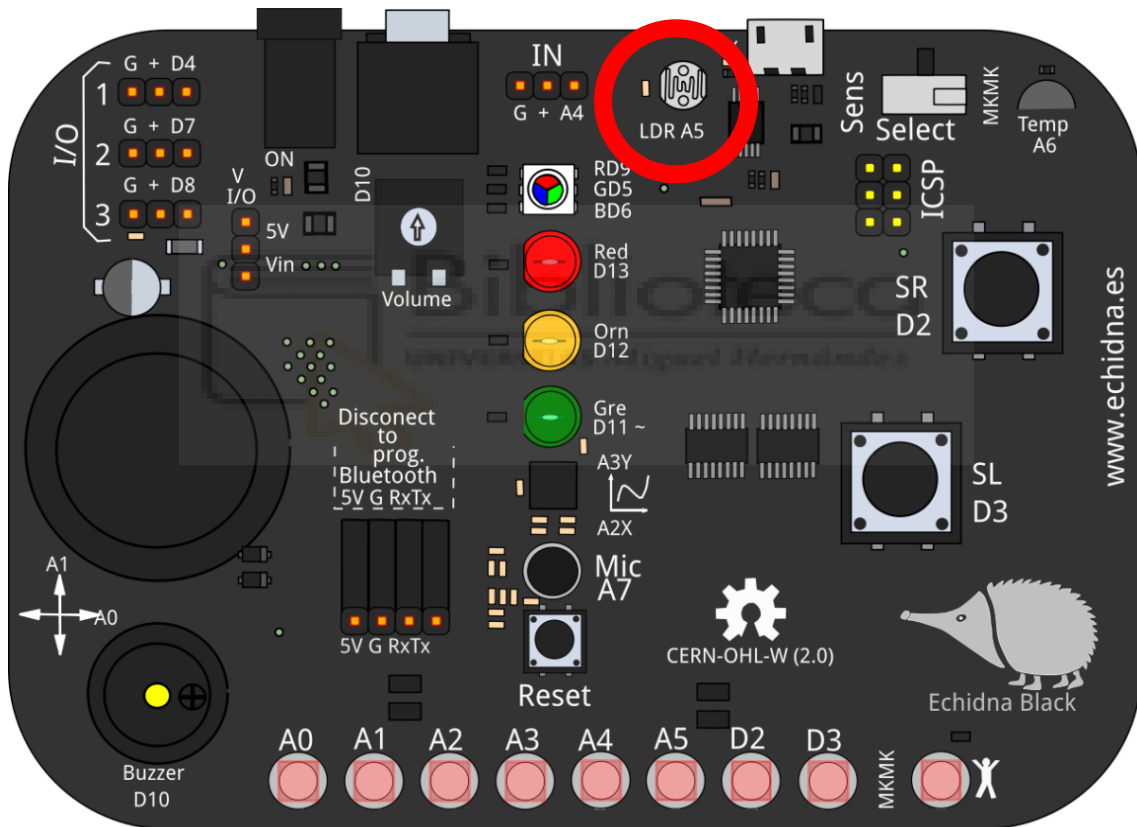
*Montaje de la marioneta*

(Fernández J. R., 2024)

#### 4.2 Sensores y actuadores. Conexiones físicas.

Como se ha comentado anteriormente, vamos a dotar al sistema de tres receptores: de proximidad, de luz y de sonido, y de tres servos para que la marioneta se mueva al accionarse los mismos.

La placa Echidna trae integrada un sensor de luz (LDR). La LDR proporciona valores altos de resistencia (en torno al  $M\Omega$ ) con poca luz y valores bajos con mucha luz. Gracias a conectarla con una resistencia en serie forma un divisor de tensión que consigue invertir la lógica del funcionamiento, de forma que con mucha luz proporciona valores altos de tensión  $4,9V = 999$  (de lectura analógica) y valores bajos con poca luz ( $0,0V = 000$ ).

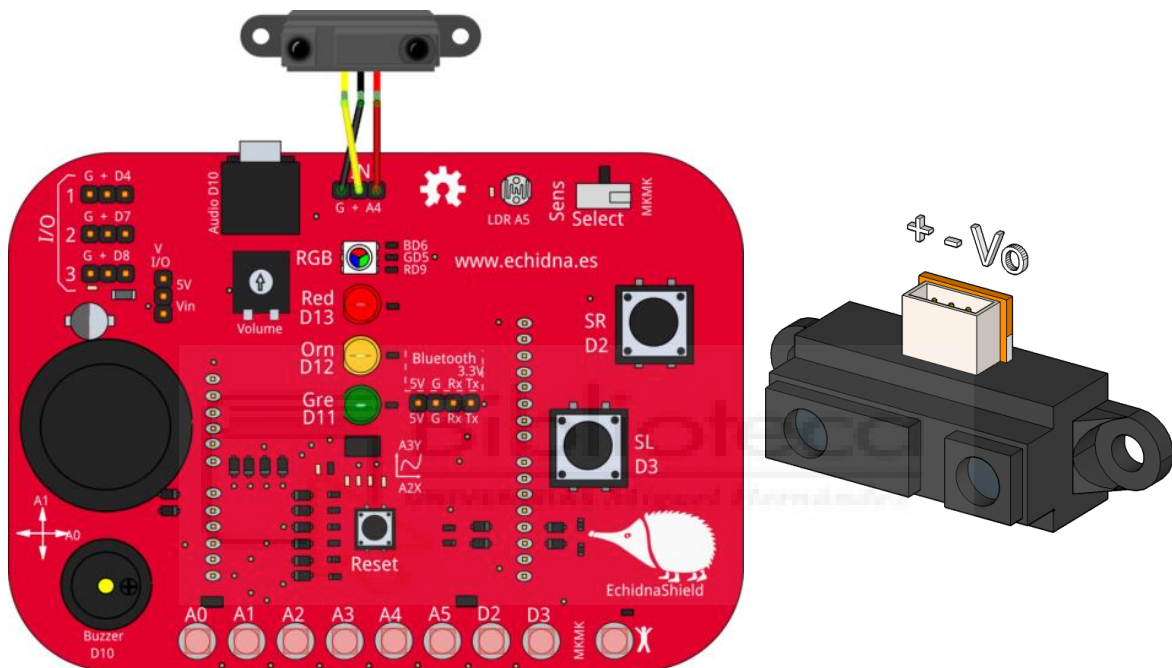


Sensor LDR en placa Echidna



El sensor de distancia se debe adquirir, ya que es externo, y se conecta a la placa según las instrucciones que nos da la web de Echidna.

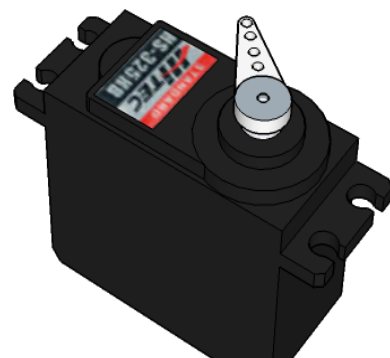
Es un sensor que proporciona una tensión según la cantidad de infrarrojo que rebota en una superficie. Se conecta directamente a 5v, GND y A4. Proporciona una salida no lineal. Según el modelo que se use tiene diferentes rangos de medida. Se ven afectados por la luz. Se deben conectar los cables del Sensor a un conector hembra de 3 pines (2,54 mm)



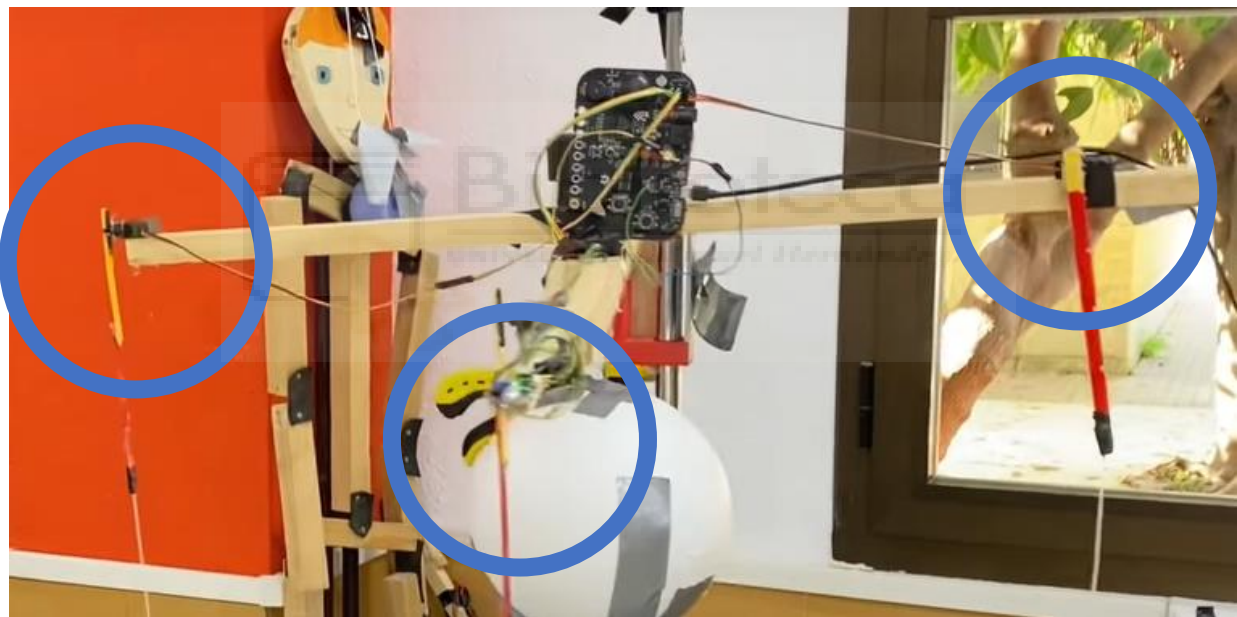
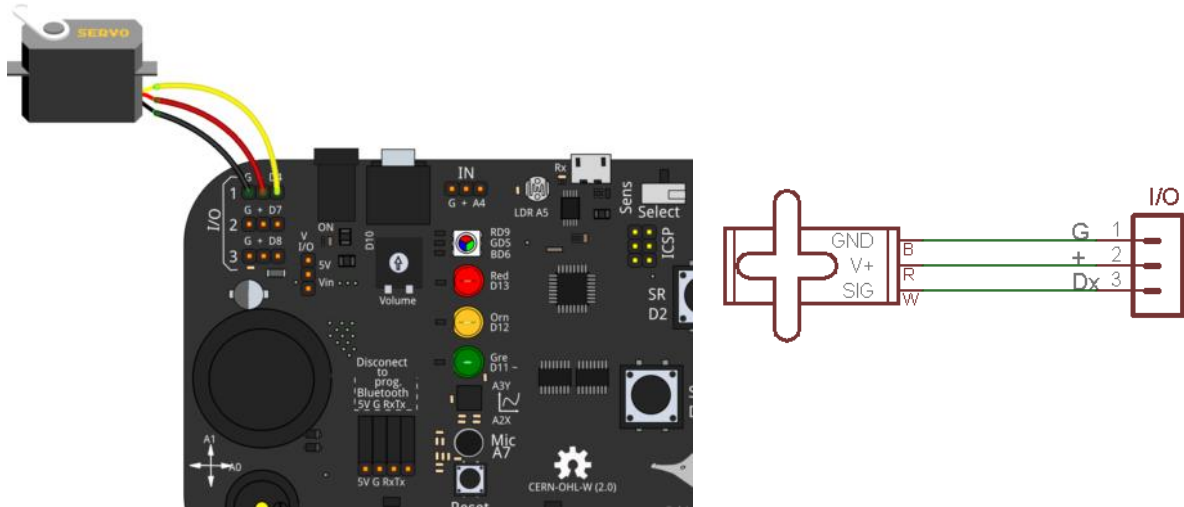
*Sensor de distancia en placa Echidna*

Y para recibir el sonido, se usa directamente el sensor de volumen de sonido que nos aporta el programa Scratch, y que detectará el nivel desde el micrófono de nuestro ordenador.

En el caso de los actuadores, se adquieren tres servomotores de posición, que son motores de corriente continua con una reductora y electrónica de control que permiten posicionarlo en un ángulo entre 0 y 180°. Se controlan mediante pulsos de 1-2 ms en periodos de 20 ms. Se conectan directamente con sus cables y se aconseja colocar el jumper de alimentación en Vin y alimentar la placa mediante el jack de alimentación. Se colocarán en las puntas de la cruz de la marioneta para poder conectar a ellos los hilos que tiren de los brazos y cadera para generar el movimiento.



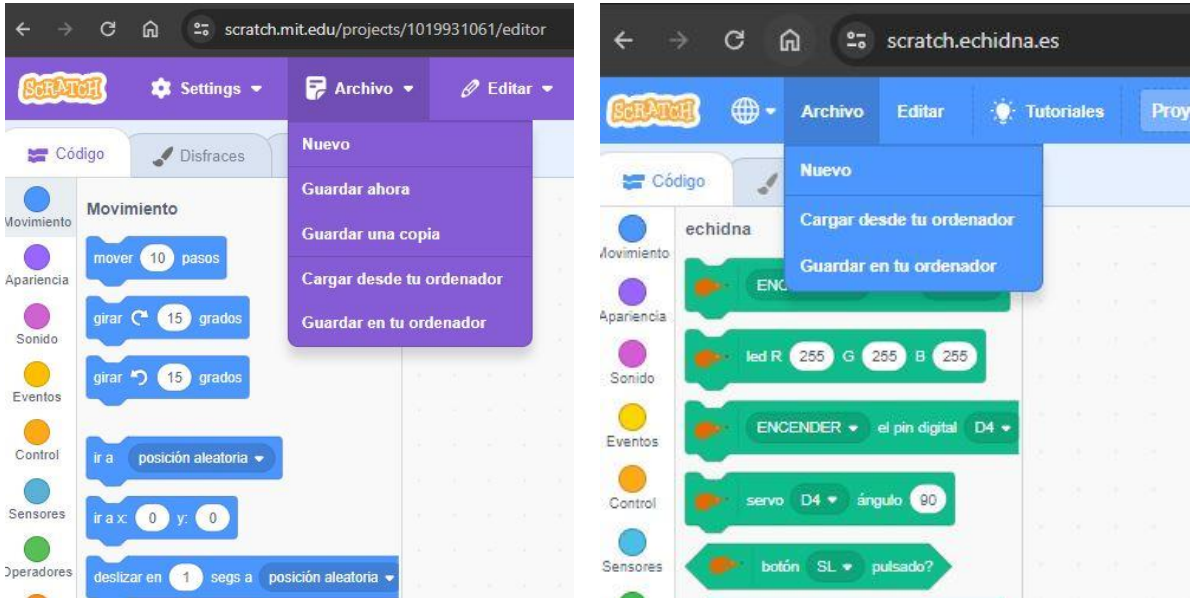
*Servomotor*



*Instalación de los servomotores en la marioneta*

### 4.3 Programación

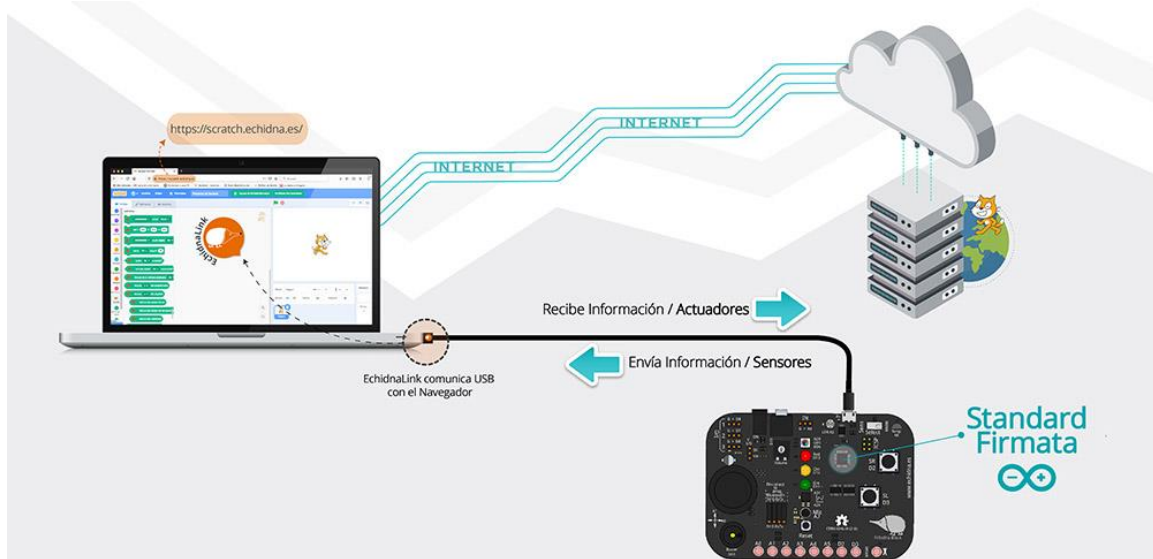
Antes de iniciar la explicación de cómo programar el sistema, debemos diferenciar entre la interfaz de [Scratch normal](#), y la de EchidnaScrath. Como se ha comentado anteriormente, el proyecto Echidna creó una plataforma de Scratch modificada que proporciona nuevos bloques para programar las placas Echidna. Se puede iniciar la programación perfectamente en la plataforma habitual de Scratch, guardar el archivo generado, y cargarlo en la plataforma de Echidna. A esta nueva plataforma, denominada [EchidnaScrath](#) se accede desde otro enlace web y se debe cargar el archivo, o bien iniciar el proyecto desde el inicio en esta plataforma.



Scratch en modo normal y modo Echidna

Para que la plataforma de programación EchidnaScrath pueda comunicarse con la placa Echidna, necesita que se establezca un enlace entre el navegador web donde se ejecuta EchidnaScrath y la propia placa. Ese enlace es proporcionado por la aplicación Echidnalink. De manera que antes de poder programar con EchidnaScrath, es necesario:

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



Conexiones de Echidna con el PC

Una vez descargado el programa Echidnalink, conectamos a nuestro ordenador la placa a través de un cable USB, y ejecutamos el programa Echidna. Una vez que el ordenador haya reconocido la placa, ya podemos acceder a la plataforma EchidnaScratch para programar desde cero o cargar algún programa ya iniciado previamente desde Scratch habitual.

En este caso vamos a trabajar directamente todo desde la plataforma EchidnaScratch. Se realizarán tres programas distintos, uno para cada sensor, con sus respectivas reacciones. De esta se observará mejor en la interfaz los bloques de código, aunque perfectamente se pueden unir para trabajar los sensores de forma simultánea.



### Programa Sensor de sonido.

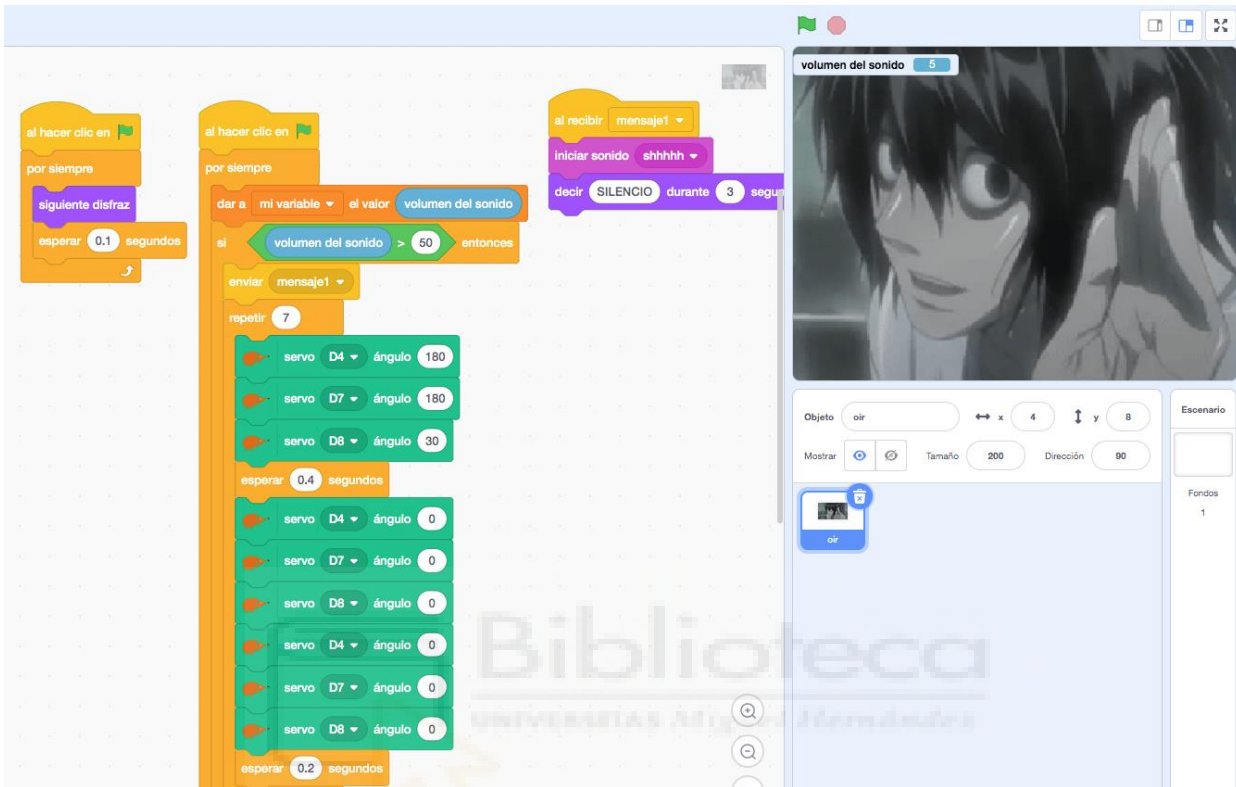
Como se ha comentado, en este caso usaremos el propio micrófono del ordenador, y el sensor de volumen de sonido que incorpora Scratch. Se creará un bucle de tal manera que se esté constantemente comprobando el nivel de sonido que entra por el micrófono. Cuando supere el umbral asignado, hará que se activen los tres servos, dos por brazo y uno por la cadera. Gracias al movimiento de los hilos conectados a los servos, la marioneta iniciará una serie de 7 movimientos.

A los servos de los brazos se le asignará un movimiento de 180 grados, y al de la cadera, un movimiento menos marcado, de 30 grados. De tal manera que la marioneta levante y baje los brazos, y mueva la cadera de una forma más ligera. Recordemos que, para realizar el movimiento, el servo debe ir a la posición de 180 grados, pero también volver a la posición inicial de 0 grados, para realizar estos cambios le daremos un tiempo de espera de 0.4 segundos. Una vez realizados los 7 movimientos, y mientras el nivel de sonido no supere el parámetro asignado, los servos permanecerán en 0 grados, sin movimiento.

Recordemos que gracias a la plataforma EchidnaScratch, tenemos los bloques de programación de los servos perfectamente accesibles en la interfaz de programación, facilitando mucho el proceso.

Aprovechando que Scratch nos ofrece en su plataforma más reacciones (salida de sonido, imagen, texto...) ante el desencadenamiento de eventos, también se puede programar que, al saltar nuestra variable de sonido, se reproduzca un audio y se muestre un texto. En este caso serán un sonido y un texto demandando "silencio". Para darle más personalización, cuando se inicia el programa se muestra un gif con una persona en modo escucha en el escenario, esperando que llegue el sonido que active tanto el movimiento de la marioneta como las reacciones en Scratch.

En el Anexo I encontramos el código completo, pero aquí adjuntamos una previsualización de cómo podría quedar el programa.



Programa sensor audio

### Programa Sensor de distancia.

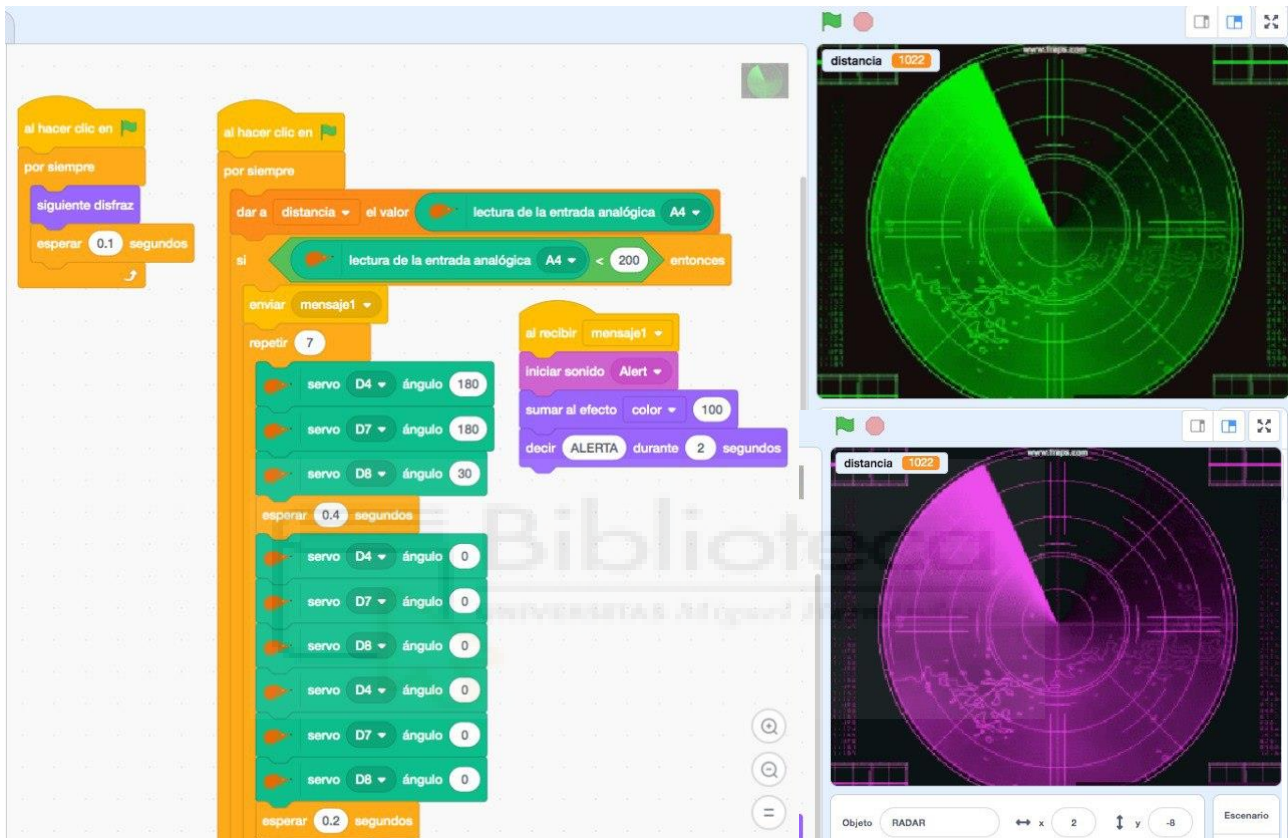
En este caso el sensor no está integrado en la placa Echidna, se deberá conectar físicamente como se indica en las instrucciones de la placa, en su entrada analógica A4. Una vez más, gracias a trabajar con la plataforma EchidnaScratch, se puede usar el bloque de lectura de la entrada analógica A4 para configurar la lectura, recibir los datos y decidir a partir de qué distancia se activen los servomotores que hagan moverse a nuestra marioneta.

Se debe programar de nuevo la manera de monitorizar constantemente los valores del sensor, de tal manera que cuando detecte presencia a menos de la distancia programada, se vuelvan a activar nuestros servomotores, de la misma manera que en el ejemplo anterior.

De nuevo, y para darle más uso a la plataforma Scratch, además del movimiento de la marioneta, se puede ejecutar también la salida de un audio, y mostrar algún texto. En este caso se ha programado que cuando se detecte presencia, el gif

que simula un radar, cambie de color, se emita un sonido de alarma, y también salte un texto en el programa alertando de la presencia detectada

En el Anexo II se pueden consultar con más detalle las características del programa.



*Programa sensor distancia*

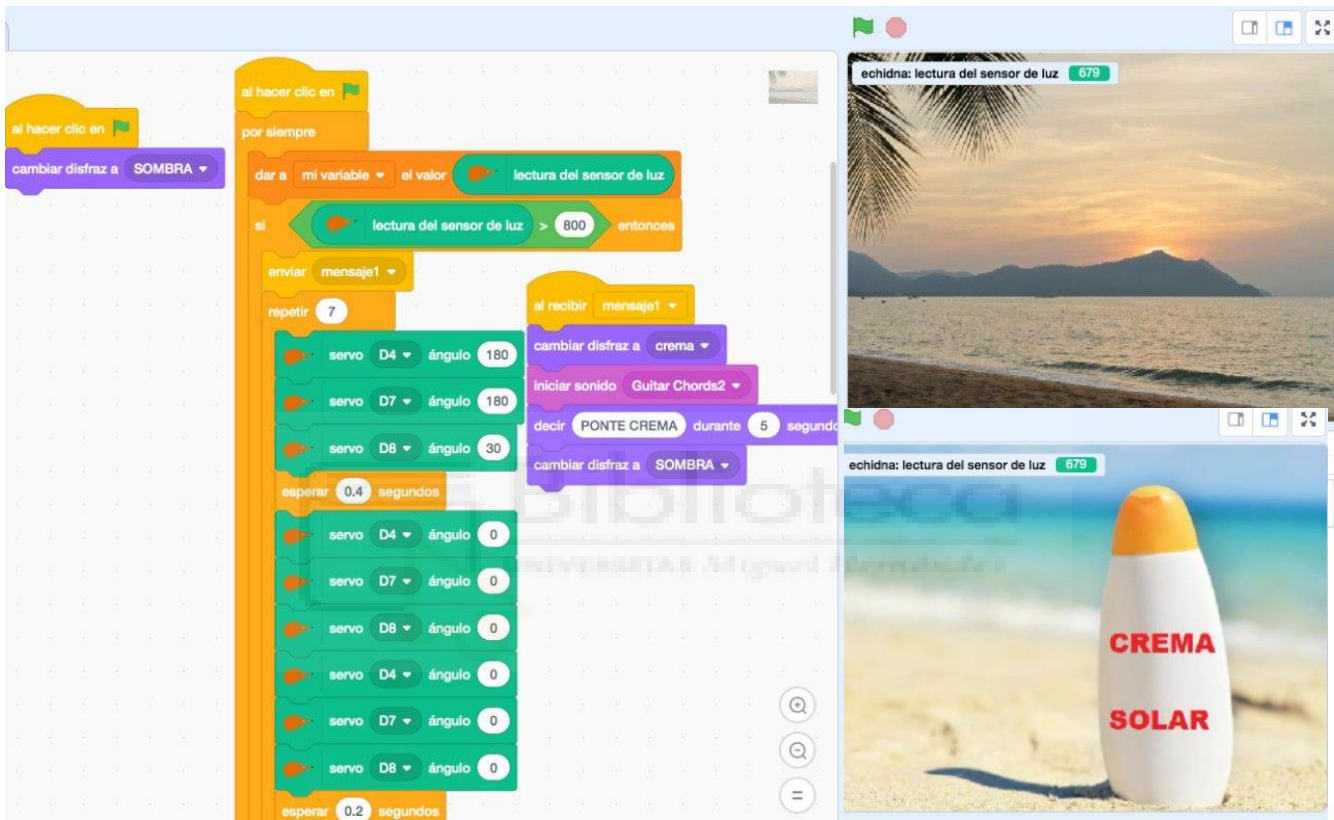
### Programa Sensor de luz.

Como se comprobó en las características de la placa Echidna, esta incorpora de manera integrada un sensor LDR. Para que EchidnaScratch proceda a realizar la lectura de la cantidad de luz que recibe, se debe usar el bloque que nos ofrece, denominado “lectura del sensor de luz”. El objetivo es que cuando se detecte una cantidad de luz superior al valor establecido, de nuevo se activen nuestros servos para que la marioneta se mueva como en los ejemplos anteriores.

Para que el alumnado siga trabajando con la plataforma también se propone que al recibir la orden se ejecute un cambio de escenario, la reproducción de un audio y mostrar un texto. En este caso para darle contexto al programa, se plantea que se recibe luz solar, y el escenario pasa de una playa anocheciendo a una playa

con Sol, en la que se muestra una crema solar y se mostraría el texto “ponte crema”.

En el Anexo III se pueden consultar con más detalle las características del programa.



Programa sensor LDR

## 5. Conclusiones y futuros trabajos

Gracias a los estudios revisados llegamos pronto a la conclusión de que la robótica sí es un campo perfecto para introducir en nuestra aula. Y consideramos que la elección de la placa Echidna y la plataforma de programación Scratch ha sido acertada para el propósito de nuestro proyecto de aprendizaje en robótica y lógica computacional para alumnos de Tecnología de 4º de la ESO.

Las herramientas elegidas han resultado idóneas: respecto a la placa Echidna, se ha comprobado que es versátil y accesible, permitiendo al alumnado experimentar con una amplia gama de sensores y actuadores facilitando así la comprensión de conceptos fundamentales de robótica.

Por otra parte, Scratch es un lenguaje de programación intuitivo y amigable que elimina las barreras de entrada, permitiendo al estudiantado desarrollar habilidades de programación de manera creativa y divertida. Además, al trabajar con la interfaz de EchidnaScratch, que está personalizada para Echidna, ha facilitado la manera de programar ciertos componentes de la placa, ya que poseen su bloque específico en la interfaz.

Con esta propuesta, el alumnado trabajará con los elementos básicos de la programación, aplicándose bucles de control, variables, operadores condicionales, "llamadas"... proporcionando una base importante en programación. Estos fundamentos son esenciales para el desarrollo de habilidades más avanzadas en el campo de la tecnología y la informática. Al comprender y practicar estos conceptos desde una etapa temprana, los estudiantes están mejor preparados para abordar desafíos más complejos pensando en su futuro.

También es importante subrayar que el proyecto ha puesto énfasis en el trabajo con elementos fundamentales de robótica, incluyendo sensores, actuadores y la propia placa. Estos componentes forman la base de cualquier proyecto robótico y proporcionan a los estudiantes una comprensión de los principios básicos de propia robótica. Asimismo, al interactuar con sensores para percibir el entorno y actuadores para realizar acciones físicas, el alumnado puede experimentar de primera mano cómo funciona la tecnología robótica en la práctica.

Además, es fundamental destacar la relevancia del trabajo manual en el ámbito de la tecnología. Durante la ejecución del proyecto, el estudiantado no solo se involucra en la programación y el diseño digital, sino que también tiene la oportunidad de aplicar habilidades prácticas al construir una marioneta de gran tamaño que se mueve al activar los servomotores. De tal modo que no solo trabajan la teoría, sino que se obtiene una apreciación más profunda de la importancia del trabajo manual en la materialización de ideas tecnológicas. Se fomentan las habilidades motoras, la resolución de problemas y la creatividad, complementando así de manera integral el enfoque educativo de nuestro proyecto.

En definitiva, entendemos que es una buena propuesta de aprendizaje ya que se desarrollan conocimientos básicos de la robótica, pensamiento computacional, y además se trabajan las habilidades prácticas construyendo la marioneta.

Como recomendaciones para futuros trabajos, se puede plantear:

Añadir más sensores: Además de los sensores de LDR, sonido y distancia, se podría considerar agregar otros sensores como un sensor de temperatura, sensor de movimiento (PIR), sensor de color, etc.





**Integrar más movimientos:** Se podría experimentar con la cantidad de movimientos que realiza la marioneta. La opción de integrar más servomotores para permitir movimientos más complejos, como girar la cabeza, mover los brazos o las piernas en múltiples direcciones.

**Implementar control remoto:** Añadir un módulo de comunicación inalámbrica, como Bluetooth o Wi-Fi, para controlar la marioneta de forma remota a través de un smartphone, tablet o computadora. Esto podría abrir posibilidades para realizar actuaciones interactivas o juegos.

**Programación más avanzada:** Usar técnicas de programación más avanzadas para hacer que la marioneta reaccione de manera más inteligente a los estímulos de los sensores. Se podría implementar algoritmos de aprendizaje automático para que la marioneta aprenda y adapte su comportamiento con el tiempo.



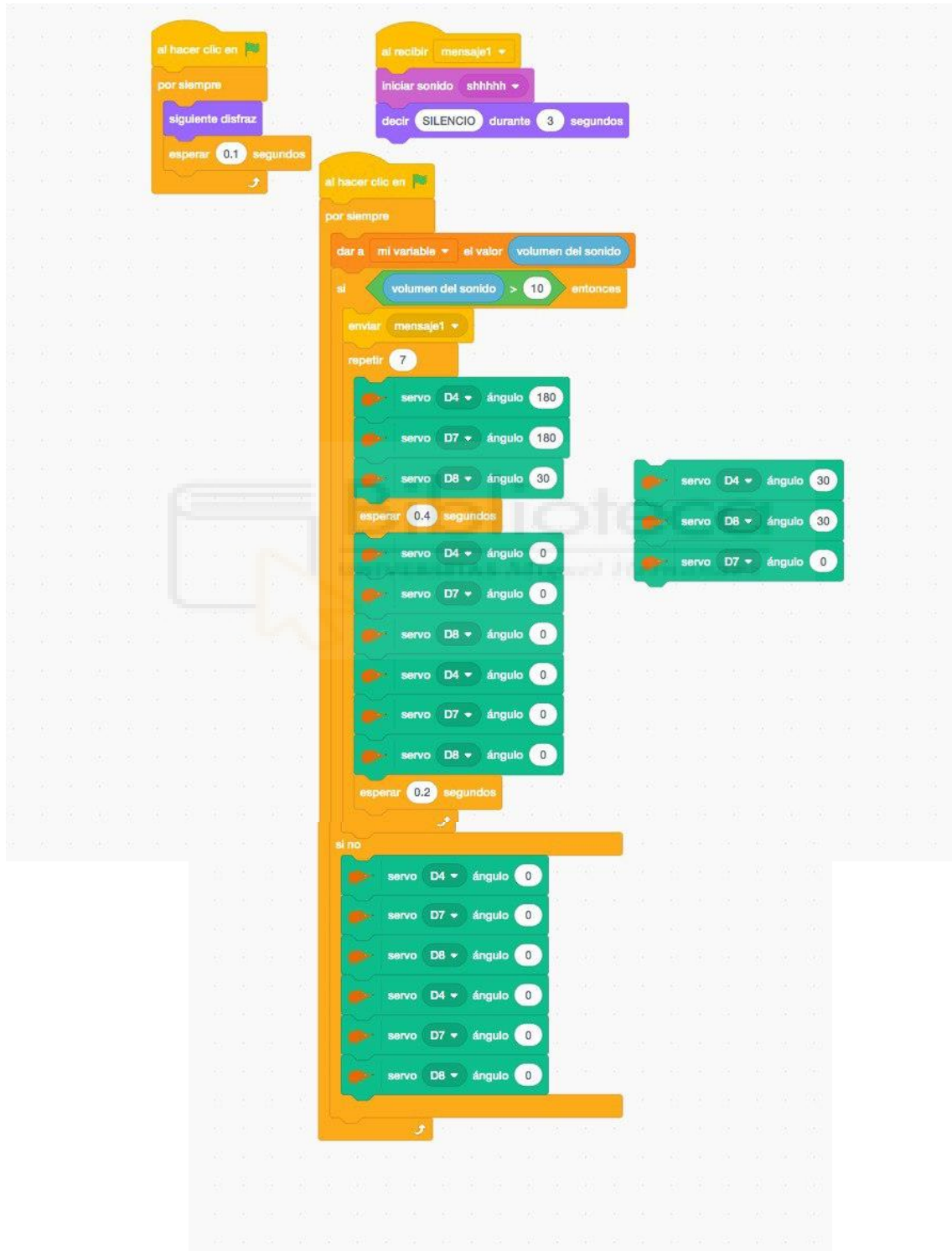
## 6. Bibliografía

- Blas Padilla, D., & Jaén Martínez, A. (2018). Experiencia didáctica con Arduino. El aprendizaje basado en proyectos como metodología de trabajo en el aula de secundaria. *Hekademos: Revista Educativa Digital*, ISSN-e 1989-3558, N.º. 25, 2018, Págs. 73-82, 25(25).
- Crespo, E. (26 de abril de 2020). *Aprendiendo Arduino*. Obtenido de Aprendiendo a manejar Arduino en profundidad:  
<https://aprendiendoarduino.wordpress.com/2016/12/18/sensores-y-actuadores/>
- DataScientest. (27 de febrero de 2024). *DataScientest*. Obtenido de  
<https://datascientest.com/es/scratch-el-lenguaje-de-programacion-para-iniciarse>
- Echidna Educación. (2024). *Echidna Educación*. Obtenido de <https://echidna.es/>
- EDS Robotics. (21 de febrero de 2022). *Los 12 tipos de sensores más usados: características y funciones*. Obtenido de  
<https://www.edsrobotics.com/blog/tipos-sensores-mas-usados/>
- Educación 3.0. (sf). *Educación 3.0*. Obtenido de  
<https://www.educaciontrespuntocero.com/tecnologia/placas-de-programacion/>
- Educación Robótica. (10 de febrero de 2023). *El Origen y la Historia de la Robótica Educativa*. Obtenido de <https://educacionrobotica.com/historia-y-evolucion-robotica-educativa/>
- Fernández, J. R. (2024). *Tecnoelxblog*. Obtenido de tecnoelxblog:  
<https://tecnoelxblog.wordpress.com/>
- Fernández, Y. (23 de septiembre de 2022). *Xataka*. Obtenido de  
<https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
- García Rodríguez, A. (2022). Enseñanza de la programación a través de Scratch para el desarrollo del pensamiento computacional en educación básica secundaria. *Academia y Virtualidad*, 15(1). <https://doi.org/10.18359/ravi.5883>
- García-Tudela, P. A., & Marín-Marín, J. A. (2023). Use of Arduino in Primary Education: A Systematic Review. In *Education Sciences* (Vol. 13, Issue 2).  
<https://doi.org/10.3390/educsci13020134>

- Linares, I. (14 de diciembre de 2023). *Hype: gadgets y mucho más*. Obtenido de <https://thehype.substack.com/p/guia-para-elegir-tu-primera-raspberry>
- López Ramírez, P. A., & Andrade Sosa, H. (2013). Aprendizaje de y con robótica, algunas experiencias. *Revista Educación*, 37(1).  
<https://doi.org/10.15517/revedu.v37i1.10628>
- micro log. (s.f.). *micro log*. Obtenido de <https://microlog.es/microbit/3560-microbit-v2.html>
- micro log. (s.f.). *micro log*. Obtenido de <https://echidna.microlog.es/caracteri/>
- micro:es. (s.f.). *micro:es*. Obtenido de La nueva versión de la tarjeta micro:bit:  
<http://microes.org/caracteristicas.php>
- Para Arduino. (17 de enero de 2024). *Para Arduino*. Obtenido de <https://paraarduino.com/placas/>
- Pastor, J. (28 de septiembre de 2023). *Xataka*. Obtenido de Llega la nueva Raspberry Pi 5: <https://www.xataka.com/ordenadores/raspberry-pi-5-caracteristicas-precio-ficha-tecnica>
- Pérez Tavera, I. H. (2023). Arduino en el nivel medio superior. *Vida Científica Boletín Científico de La Escuela Preparatoria No. 4*, 11(22).  
<https://doi.org/10.29057/prepa4.v11i22.10962>
- Pérez-Tavera, I. H. (2019). Scratch en la educación. *Vida Científica Boletín Científico de La Escuela Preparatoria No. 4*, 7(13).
- roboticoss. (2024). *roboticoss*. Obtenido de <https://roboticoss.com/actuadores-para-robotica/>
- Wikipedia. (2024). Obtenido de <https://es.wikipedia.org/wiki/Arduino>

## 7. Anexos

### Anexo I: Programación de bloques para el sensor de audio



**Anexo II: Programación de bloques para el sensor de distancia**

```

    al hacer clic en
    por siempre
    siguiente disfraz
    esperar 0.1 segundos

    al recibir mensaje1
    iniciar sonido Alert
    sumar al efecto color 100
    decir ALERTA durante 2 segundos

    al hacer clic en
    por siempre
    dar a distancia el valor lectura de la entrada analógica A4
    si lectura de la entrada analógica A4 < 200 entonces
    enviar mensaje1
    repetir 7
    servo D4 ángulo 180
    servo D7 ángulo 180
    servo D8 ángulo 30
    esperar 0.4 segundos
    servo D4 ángulo 0
    servo D7 ángulo 0
    servo D8 ángulo 0
    servo D4 ángulo 0
    servo D7 ángulo 0
    servo D8 ángulo 0
    esperar 0.2 segundos
    si no
    servo D4 ángulo 0
    servo D7 ángulo 0
    servo D8 ángulo 0
    servo D4 ángulo 0
    servo D7 ángulo 0
    servo D8 ángulo 0
  
```

### Anexo III: Programación de bloques para el sensor LDR

