



## ¿Eres un experto en ortografía?

**Ciclo:** Segundo

**Curso:** 3º E.S.O.

**Áreas Curriculares:** Lengua Castellana y Literatura

**Temporalización:** 1º/2º/3º trimestre

**Nº de sesiones:** 5 sesiones.



## Descripción del proyecto

Este proyecto consiste en una actividad interactiva de verdadero/falso sobre ortografía para estudiantes de segundo ciclo de secundaria, utilizando Arduino como plataforma tecnológica. Mediante sensores siguelíneas, los alumnos seleccionan si una frase o palabra mostrada en una pantalla LCD está escrita correctamente o no. Al final, un servomotor indica el porcentaje de aciertos. Esta actividad permite a los estudiantes practicar ortografía de forma lúdica e innovadora, promoviendo la participación activa y el aprendizaje significativo.





## Objetivos

Fomentar la mejora en la ortografía y comprensión de la lengua mediante actividades prácticas.

Desarrollar habilidades tecnológicas y de programación básicas usando Arduino.

Estimular el pensamiento crítico a través de la identificación correcta de errores ortográficos.

Promover la motivación y participación activa mediante el uso de herramientas interactivas.



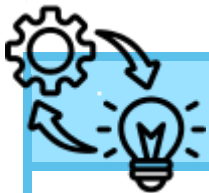
## Contenidos

Reglas básicas de ortografía y gramática.

Identificación de errores ortográficos comunes.

Uso de tecnologías para apoyar el aprendizaje lingüístico.

Manejo básico de Arduino y componentes electrónicos (sensores siguelíneas, zumbador, display LCD, servomotor).



## Metodología

En el desarrollo de este proyecto, se han seleccionado diversas metodologías didácticas con el propósito de crear un entorno educativo que sea tanto dinámico como efectivo. Estas han sido elegidas por su capacidad para involucrar activamente a los estudiantes y promover un aprendizaje significativo.



A continuación, se describen los enfoques metodológicos que se implementarán:

- **Aprender haciendo (Learning by doing):** esta metodología se basa en el construccionismo, donde los estudiantes aprenden mediante la creación de artefactos digitales. Al involucrarse activamente en la construcción de proyectos, los estudiantes desarrollan habilidades prácticas y comprenden mejor los conceptos teóricos.
- **Aprendizaje Basado en Retos (ABR):** en esta metodología, se plantean retos al estudiante que debe solucionar obteniendo un producto final. Este enfoque fomenta el pensamiento crítico, la creatividad y la capacidad de resolver problemas, ya que los estudiantes deben aplicar sus conocimientos para superar desafíos específicos.



## Preguntas para la gestión del aula

Para gestionar día a día mi trabajo, me planteo las siguientes preguntas:

- **¿Cómo puedo organizar a los alumnos para realizar esta actividad tecnológica en clase?**  
Si tienes un grupo grande, lo ideal es formar equipos de 3 a 5 alumnos con perfiles heterogéneos, mezclando distintos niveles y habilidades. Así, fomentas la colaboración y el aprendizaje entre iguales. Además, al trabajar en grupo, se facilita la resolución de problemas técnicos y la discusión sobre las respuestas lingüísticas.
- **¿Cuándo debo repartir el material tecnológico para la actividad?**  
Depende de tu estilo de enseñanza. Si consideras que es importante que los alumnos entiendan primero la dinámica y los objetivos, puedes explicar el proyecto antes de entregar el hardware. Si prefieres un enfoque más inductivo, reparte el material por fases, permitiendo que los alumnos experimenten y busquen soluciones mientras les das apoyo puntual.
- **¿Qué hago si algunos alumnos pierden interés o necesitan mucha ayuda durante la actividad?**  
Puedes crear un banco de recursos accesibles desde el aula virtual con tutoriales, instrucciones paso a paso y ejemplos. Esto ayudará a que los estudiantes trabajen con mayor autonomía y seguridad. Además,



fomentando el trabajo en equipo, los alumnos más avanzados pueden apoyar a sus compañeros, potenciando la tutoría entre iguales.

- **¿Cómo puedo aprovechar el aula virtual para complementar esta actividad?**

El aula virtual puede ser un espacio donde publiques las frases y palabras para la actividad, vídeos explicativos sobre ortografía y programación con Arduino, así como instrucciones y ejemplos del código. También puede incluir rúbricas para la autoevaluación y espacios para que los alumnos suban evidencias de su trabajo o resuelvan dudas.

- **¿Cómo gestionar el material y los proyectos entre sesiones si comparto el taller con otros profesores?**

Organiza cajas o contenedores etiquetados con el nombre de cada grupo o proyecto donde los alumnos guarden sus componentes y materiales al final de la clase. Así evitarás pérdidas y facilitarás la continuidad del trabajo en la siguiente sesión. Asegúrate de que los alumnos sean responsables de la recogida y orden del material.

- **¿Cómo puedo evaluar individualmente si trabajo en equipo?**

Asignando roles específicos dentro de cada grupo (programador, encargado del montaje, lector de frases, etc.) y rotándolos en cada sesión. Además, puedes complementar con pruebas individuales de ortografía o pequeñas actividades escritas que verifiquen la comprensión personal. La autoevaluación y coevaluación con rúbricas también son útiles para valorar el progreso individual y grupal.



## Recursos

Personales	Materiales	Digitales
<b>Docentes:</b> profesores y maestros que guían y facilitan el aprendizaje.	<b>Material didáctico:</b> Frases de verdadero/falso para repasar ortografía, esquemas y modelos	<b>Plataformas educativas:</b> Aula Virtual de Educamadrid, Google Classroom, etc.

**Estudiantes:**

compañeros de clase que colaboran y aprenden juntos.

visuales para el montaje de Arduino en caja de zapatos, mapas conceptuales sobre reglas ortográficas, rúbricas y listas de cotejo para autoevaluación y coevaluación.

**Aplicaciones y software educativo:** Arduino IDE

**Tecnología:** Kits Arduino y ordenadores o portátiles con Arduino IDE instalado.

**Material de oficina:** Tijeras, cúteres, pegamento, cinta adhesiva o cinta de doble cara, lápices, bolígrafos, papel, cartulina para etiquetas, regla para medir y marcar.



## Actividades

<b>Fase</b>	<b>1</b>
<b>Temporalización</b>	2 sesiones de 45 minutos
<b>Tipo de Actividad</b>	Introducción, presentación del proyecto y funcionamiento básico de los componentes necesarios (servomotor, siguelíneas, zumbador, Display LCD)
<b>Descripción</b>	Se presenta el objetivo de la actividad y se explica el funcionamiento básico del servomotor, de los sensores siguelíneas y del zumbador en una primera sesión y del Display LCD en una segunda sesión. Se forman grupos



heterogéneos y se explica el funcionamiento de los componentes, realizando programas sencillos con cada uno de ellos para aprender o recordar cómo se utilizan.

## **Programa 1: aprendemos/recordamos cómo utilizar un servo 180.**

Un SERVO 180° es un pequeño motor controlado electrónicamente que puede girar entre 0° y 180°. La ventaja que tiene es que puede ser posicionado en este rango en cualquier ángulo de manera precisa.

Utilizaremos la placa de expansión de motores conectando los tres cables del servo a la ranura específica para servos. El cable de señal queda conectado al pin 9.

Para programarlo necesitamos la **biblioteca Servo.h**.

Incluimos la biblioteca servo.h (`#include <Servo.h>`), creamos una instancia de la clase servo a la que llamaremos por ejemplo miServo (`Servo miServo;`) y en setup asignaremos el pin 9 al servo (`miServo.attach(9);`). Para que el servo se posicione en un ángulo determinado hay que utilizar la instrucción `miServo.write(ángulo);` donde ángulo es un número cualquiera entre 0 y 180.

El alumno debe hacer un programa en el que el servo al principio del programa se ponga en la posición de 0° y después, se posicione en distintos ángulos de su elección parando medio segundo entre cada posición.

Para que el servo se mueva no es suficiente con la alimentación del cable USB, se debe conectar la alimentación por medio del portapilas.

## **Programa 2.1 y 2.2: aprendemos/recordamos cómo utilizar el sensor siguelíneas.**

Un sensor siguelíneas es un dispositivo que se utiliza para detectar líneas negras sobre una superficie máscara. Emite luz infrarroja y detecta la cantidad que se refleja en la superficie. Es un **sensor digital**. Devuelve un **1 si debajo detecta blanco y un 0 si detecta negro**. El blanco reflejará la luz infrarroja y el negro no. En



nuestro caso no habrá ninguna superficie delante del sensor y por tanto no reflejará nada: detectará negro (0); si colocamos nuestra mano o una tarjeta delante del sensor se reflejará la luz infrarroja y detectará blanco (1).

Conectaremos el pin de señal del siguelíneas a un pin digital cualquiera por ejemplo al pin 5 de la placa Arduino (podemos utilizar `#define` para asignar el pin 5 a una constante que definamos de manera que sea sencillo cambiar el pin: `#define PinSL 5`). En nuestro programa utilizaremos la función `pinMode` para definir este pin de entrada: `pinMode (PinSL, INPUT)`; dentro del `setup`.

**Programa 2.1: vamos a detectar blanco.** El alumno debe hacer un programa en el que cuando se detecte blanco el LED integrado en la placa (pin 13) se encenderá y cuando no detecte blanco estará apagado.

Se debe definir con `pinMode` el pin 13 de salida (`pinMode(13, OUTPUT)`). Dentro del loop se debe utilizar la estructura `if(...){...} else {...}` para que en función de lo que se lea por el pin del siguelíneas, escriba en el pin 13 0 o 1 para encender o apagar el LED. Se tiene que usar `digitalRead` para obtener la señal que lee el siguelíneas (1 blanco, 0 negro) y `digitalWrite` para apagar o encender el LED integrado en la placa y conectado al pin 13.

**Programa 2.2. detectamos negro.** El alumno debe hacer un programa similar al anterior, pero en este caso detectando negro.

**Programa 3: aprendemos/recordamos cómo utilizar el zumbador.**

Usaremos el zumbador integrado en la placa de expansión, se encuentra conectado al pin 4. Para usarlo no hay que hacer ninguna conexión.

Podemos definir el pin 4 como el pin que vamos a usar con el zumbador (`#define PINZUMB 4`).

Usaremos la siguiente función para hacer que suene el zumbador: `tone (PINZUMB, freq, tiempo)`;

Las frecuencias de la escala básica de notas musicales



son: Do: 262 Hz - Re: 294 Hz - Mi: 330 Hz - Fa: 349 Hz - Sol: 392 Hz - La: 440 Hz - Si: 494 Hz - Do: 523 Hz.

El alumno debe escribir un programa para que suene la escala musical en el zumbador de Arduino, para ello debe buscar la frecuencia de las notas que deben sonar. Debe usarse la función delay(t) entre cada nota.

## **Programa 4.1 y 4.2: aprendemos/recordamos cómo utilizar el Display LCD.**

Un Display LCD con controlador I2C es una pantalla de cristal líquido que incluye un controlador para manejar la visualización de caracteres. El controlador simplifica la comunicación entre Arduino y la pantalla. El Display LCD se conecta a la placa Arduino conectando el cable SDA de datos al pin analógico A4 y el cable SCL del reloj al pin analógico A5 de la placa.

Para programarlo se debe instalar e incluir en el programa la **biblioteca LiquidCrystal\_I2C.h** (`#include <LiquidCrystal_I2C.h >`).

Se debe crear una instancia de la pantalla LCD en nuestro caso de 16 x 2 caracteres: `LiquidCrystal_I2C MiLCD (0x27, 16, 2);`

En setup se inicia la pantalla y se habilita la retroalimentación de la misma:

```
MiLCD.init();
```

```
MiLCD.backlight();
```

Para mostrar texto en el Display LCD se utilizan las siguientes funciones:

```
MiLCD.clear(); - borra el contenido de la pantalla
```

```
MiLCD.setCursor(x, y); - colocar el cursor en la posición x, y; donde x varía entre 0 y 15; e y valdrá 0 o 1 (0 para la primera línea, 1 para la segunda línea).
```

```
MiLCD.print("texto"); - muestra "texto" en el Display.
```

El alumno debe hacer un programa que realice lo



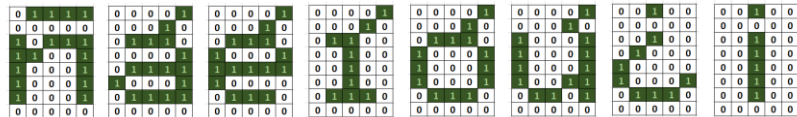
siguiente: en primer lugar, borra la pantalla. Después, en la primera línea a partir del segundo carácter se escribe "Bienvenido", espera 1 segundo, después sitúa el cursor en la segunda línea, en la posición quinta y escribe "Hasta pronto". Espera 1 segundo y que vuelva a empezar todo.

**Advertencia:** en cada línea puedes escribir hasta 16 caracteres.

**IMPORTANTE:** No admite vocales acentuadas, ñ, vocal con diéresis, ni signos de interrogación o acentuación iniciales. Para poder escribirlos tienes que definir un carácter y después llamarlo. Veamos cómo se hace:

Se pueden crear hasta 8 caracteres personalizados. Nosotros crearemos ñ, á, é, í, ó, ú, ç, j

Hay que tener en cuenta que cada carácter está en una cuadrícula de 5x8. Cuando queramos que uno de los cuadrados esté en negro pondremos 1, si queremos blanco un 0. Aquí tienes varios ejemplos:



Se deben definir cada uno de los caracteres que se deseen usar (un máximo de 8 caracteres especiales) de la siguiente manera (ejemplo de la ñ anterior):

```
byte gn [8] {
0b01111,
0b00000,
0b10111,
0b11001,
0b10001,
0b10001,
0b10001,
0b00000
};
```



Una vez que se hayan definido los caracteres especiales que se desean usar, se debe hacer que el Display los cree para que pueda usarlos: `MiLCD.createChar(i, gn);` donde *i* es un número entre 0 y 7. Aquí creamos el carácter especial gn (ñ) en mi display LCD.

Para que los muestre en la pantalla se usa la función write: `MiLCD.write(byte(i));` donde *i* es un número entre 0 y 7, que mostrará el carácter asignado a ese número. Por ejemplo, el 0 es la ñ.

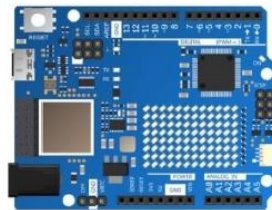
El alumno debe hacer un programa donde se muestran los 8 caracteres siguientes: ñ, á, é, í, ó, ú, ç, j

Además de lo anterior, se muestra al alumnado cómo montar los componentes en la caja de cartón. Se introducen brevemente las reglas ortográficas que se trabajarán durante el proyecto.

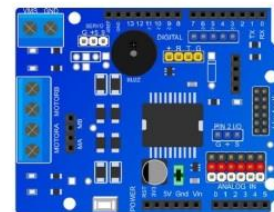
## Recursos

### Componentes.

Arduino UNO R4 Wifi



Placa de expansión de control de motores



Portapilas x5 AA con adaptador Jack



Cables dupont



2 Sensores Sigue líneas



Servomotor 180º

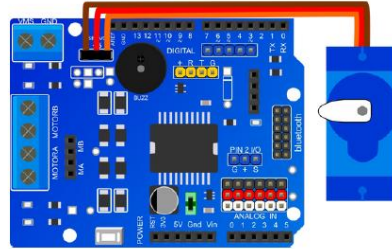


Display LCD con controlador I2C





## PROGRAMA 1: Servo

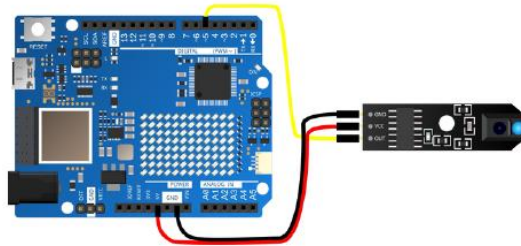


### Mapa de pines

Los tres cables del servo se conectan a la ranura específica para servos. El cable de señal queda conectado al pin 9.

Programa 1\_Servo.ino ([documento descargable](#))

## PROGRAMA 2: Siguelíneas (Programas detecta blanco y detecta negro)



### Mapa de pines:

El pin de señal del siguelíneas a un pin digital cualquiera, por ejemplo, al pin 5.

Programa 2\_1\_SigueLineas\_Blanco.ino ([documento descargable](#))

Programa 2\_2\_SigueLineas\_Negro.ino ([documento descargable](#))

## PROGRAMA 3: Zumbador

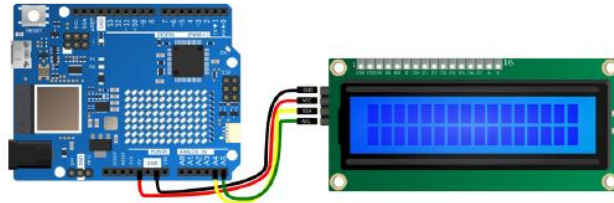
### Mapa de pines:



No hay que hacer ninguna conexión.

Programa 3\_Zumbador.ino ([documento descargable](#))

**PROGRAMA 4: Display LCD**



**Mapa de pines:**

El cable SDA de datos del display LCD se conecta al pin analógico A4 y el cable SCL del reloj al pin analógico A5 de la placa.

Programa 4\_1\_LCD.ino ([documento descargable](#))

Programa 4\_2\_LCD\_car\_esp.ino ([documento descargable](#))

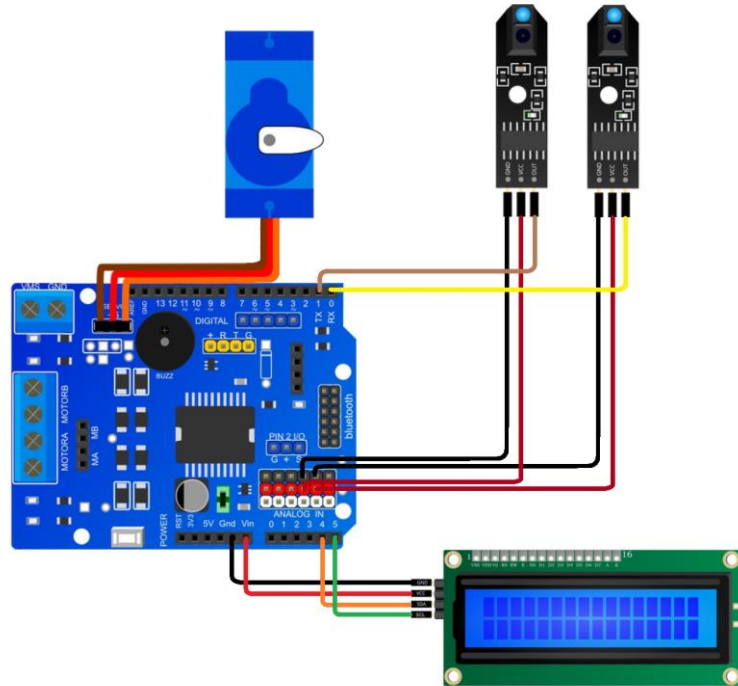
caracteres.h ([documento descargable](#))

<b>Fase</b>	<b>2</b>
<b>Temporalización</b>	45 min
<b>Tipo de Actividad</b>	Montaje de la estructura y conexiones básicas
<b>Descripción</b>	Los alumnos comienzan a montar la placa Arduino, sensores, servomotor y display LCD en la caja de cartón, usando el material de oficina para fijar los componentes. Se realizan las conexiones básicas y se familiarizan con el hardware. El profesor supervisa y guía el trabajo en equipo.



## Recursos

## Esquema de conexiones



### Mapa de pines:

Los tres cables del servo se conectan a la ranura específica para servos. El cable de señal queda conectado al pin 9.

Los dos sensores siguelíneas se conectan a dos pines digitales, por ejemplo, al 0 y al 1.

El cable SDA de datos del display LCD se conecta al pin analógico A4 y el cable SCL del reloj al pin analógico A5 de la placa.

Fase

3

Temporalización

45 min



# Proyecto didáctico



<b>Tipo de Actividad</b>	Programación inicial y pruebas básicas
<b>Descripción</b>	Los alumnos cargan un programa básico (Programa de ayuda) para mostrar 3 frases en la pantalla LCD y activar los sensores. Se realizan pruebas para comprobar el funcionamiento correcto de los sensores y la pantalla, con ajustes guiados por el profesor. Se modifica el programa para mostrar más frases seleccionadas por los alumnos. Se pueden añadir mejoras como un mensaje de bienvenida.
<b>Recursos</b>	<p><b>Programa de ayuda:</b></p> <p><b>Programa Final_ayuda.ino (<a href="#">documento descargable</a>)</b></p> <p><b>Programa con las mejoras:</b></p> <p><b>Programa Final.ino (<a href="#">documento descargable</a>)</b></p> <p><b>Programa mejorado que incluye caracteres especiales:</b></p> <p><b>Programa Final_car_esp.ino (<a href="#">documento descargable</a>)</b></p> <p><b>caracteres.h (<a href="#">documento descargable</a>)</b></p>

<b>Fase</b>	<b>4</b>
<b>Temporalización</b>	45 min
<b>Tipo de Actividad</b>	Actividad de evaluación y retroalimentación
<b>Descripción</b>	Los alumnos usan su dispositivo para realizar el juego de verdadero/falso con las frases propuestas. Se recoge el porcentaje de aciertos mostrado por el servomotor. Se realiza una puesta en común para discutir los errores ortográficos más comunes y se realiza la evaluación.



## ¿Sabías qué?

- Estamos trabajando con un siguelíneas. Explicamos cómo funcionan estos sensores y qué es la luz infrarroja.
- Estamos trabajando con un LED. Explicamos cómo se conecta un LED que es la resistencia de protección y por qué es importante cuando conectamos un LED utilizarla.
- Estamos trabajando con un zumbador. Explicamos el concepto de frecuencia. Explicamos la frecuencia de las notas.
- Estamos trabajando con un servomotor. Explicamos la diferencia entre un motor DC y un servomotor. Explicamos cómo se controla un servomotor a través de pulsos PWM.
- Estamos trabajando con un Display LCD que usa el código ASCII estándar. Explicamos qué es el código ASCII estándar y qué caracteres incluye y cuáles no.



## Evaluación

Para evaluar adecuadamente este proyecto didáctico, se han establecido procedimientos, actividades de evaluación e instrumentos que reflejan fielmente los objetivos y competencias planteados. La evaluación no solo permite medir el progreso y los logros de los estudiantes, sino que también proporciona información valiosa para ajustar y mejorar el proceso de enseñanza. A continuación, se detallan estos aspectos.

Procedimientos	Actividades de Evaluación	Instrumentos
Observación directa Intercambios orales Producciones del alumnado Autoevaluación Co-evaluación	Archivo digital Participación diaria Asamblea y puesta en común Pruebas escritas Actividades	Rúbricas Listas de cotejo Portafolio Cuaderno de clase



## Criterios de evaluación

### Lengua Castellana y Literatura

1. **Aplica las normas ortográficas de manera correcta** al escribir las palabras que forman parte del juego de verdadero o falso, demostrando dominio de la escritura correcta y cuidando la presentación del texto.
2. **Revisa y corrige de manera adecuada** los textos creados para el juego, identificando y explicando los errores ortográficos cometidos y mejorando su precisión en el uso de la lengua.

### Tecnología y Digitalización

3. **Realiza la programación de manera correcta** en la placa Arduino para que el sistema funcione adecuadamente (lectura de sensores, cálculo de aciertos y movimiento del servomotor).
4. **Usa el material y construye la maqueta de manera adecuada y correcta**, montando la estructura con los componentes electrónicos (sensores, pantalla LCD, servomotor) y materiales reciclados (como la caja de zapatos), asegurando su estabilidad y buen funcionamiento.

## Rúbrica de evaluación para el docente ([documento descargable](#))

	Excelente	Satisfactorio	Mejorable	Insuficiente
<b>1. Aplica las normas ortográficas de manera</b>	Escribe todas las palabras del juego sin errores	Comete muy pocos errores ortográficos y muestra buen	Presenta varios errores ortográficos,	Comete numerosos errores que dificultan la



# Proyecto didáctico



<b>correcta</b>	ortográficos. Demuestra dominio total de las reglas y cuida la presentación del texto.	dominio de las reglas.	aunque el texto es comprensible.	comprensión del texto.
<b>2. Revisa y corrige de manera adecuada los textos creados para el juego</b>	Revisa y corrige todos los textos con autonomía, justificando sus correcciones de forma argumentada.	Revisa y corrige la mayoría de los errores, mostrando comprensión de las normas ortográficas.	Corrige solo algunos errores cuando se le indica, con cierta dificultad para justificar los cambios.	No realiza correcciones o no comprende los errores cometidos.
<b>3. Realiza la programación de manera correcta en la placa Arduino</b>	Programa el sistema completo sin errores, integrando correctamente sensores, servomotor y pantalla LCD. Explica claramente el funcionamiento del código.	Realiza la programación con pequeños fallos que corrige de forma autónoma. El programa funciona casi por completo.	Programa parcialmente el sistema, pero requiere ayuda para resolver errores.	Tiene dificultades para programar y necesita apoyo constante.
<b>4. Usa el material y construye la maqueta de manera adecuada y correcta</b>	Construye la maqueta de forma limpia, ordenada y funcional. Utiliza los materiales con	La maqueta es funcional y está bien montada, aunque presenta leves fallos de	La maqueta cumple parcialmente su función, con errores de montaje o	La maqueta no funciona o está mal montada por uso inadecuado de



"Programa financiado por el Ministerio de Educación, Formación Profesional y Deportes"





	precisión y cuidado.	acabado o estabilidad.	desorden.	los materiales.
--	----------------------	------------------------	-----------	-----------------



## Atención a las diferencias del alumnado

Como docente comprometido con la inclusión y el éxito de todos los estudiantes, es fundamental adaptar las tareas y actividades para atender la diversidad en el aula. Siguiendo los principios del **Diseño Universal para el Aprendizaje (DUA)**, se pueden implementar estrategias flexibles y personalizadas que respondan a las necesidades individuales de cada alumno.

A continuación, se detallan las pautas y medidas que se va a aplicar para fomentar un entorno de aprendizaje inclusivo y efectivo:

- **Ubicación o agrupación del alumnado en el aula:** los estudiantes que necesitan más apoyo se sientan cerca del profesor para recibir instrucciones adicionales. Los estudiantes que trabajan mejor en grupo se agrupan en mesas colaborativas para fomentar la cooperación
- **Reconsideración de ítems en las rúbricas para su evaluación:** la rúbrica de evaluación se adapta para incluir criterios específicos adaptados a las necesidades del estudiante.
- **Variación de la ponderación de los criterios de calificación:** los criterios de calificación se ajustan según las capacidades individuales. Por ejemplo, para un estudiante con dificultades en la expresión escrita, se da más peso a la parte oral de la presentación.
- **Refuerzo de saberes básicos:** se proporcionan materiales adicionales y sesiones de refuerzo para estudiantes que necesitan consolidar conceptos fundamentales. Esto incluye videos educativos y actividades prácticas adicionales.
- **Reconsideración del grado de exigencia de los saberes básicos:** para facilitar el aprendizaje, se ajustan las expectativas según las capacidades individuales. Por ejemplo, un estudiante con necesidades educativas especiales puede centrarse en explicar solo las partes principales del ciclo del agua, mientras que otros estudiantes pueden profundizar en detalles adicionales.