

## Situación de Aprendizaje: Proyecto Bastón Inteligente para Personas Ciegas

**Nivel:** 1º Bachillerato Tecnológico

**Materia:** Robótica

**Duración estimada:** 8 sesiones

**Metodología:** Aprendizaje Basado en Proyectos (ABP)

---

### 1. Contextualización y Justificación

El Proyecto Bastón Inteligente propone al alumnado diseñar y construir un prototipo funcional que permita a personas con discapacidad visual detectar obstáculos mediante sensores ultrasónicos y motores de vibración. La experiencia se contextualiza en un caso real: un alumno con discapacidad visual de la clase probará el dispositivo. La iniciativa fomenta el aprendizaje significativo, la inclusión, la conciencia social y el pensamiento diseñador.

---

### 2. Competencias Específicas (Decreto 171/2022)

- **CE1:** Diseñar prototipos funcionales y creativos mediante proyectos colaborativos.
  - **CE2:** Configurar y programar sensores y actuadores con lógica computacional.
  - **CE3:** Evaluar y mejorar el diseño a partir de pruebas reales.
- 

### 3. Objetivos de Aprendizaje

- Aplicar conocimientos de electrónica, programación y diseño 3D.
  - Integrar sensores ultrasónicos y motores de vibración mediante una placa Arduino Nano.
  - Desarrollar un prototipo funcional con impacto social.
  - Evaluar la eficacia del prototipo mediante pruebas con un usuario real.
- 

### 4. Activación del Conocimiento Previo

**Actividad:** Reflexión inicial sobre accesibilidad y tecnología: "¿Cómo puede la robótica mejorar la vida de las personas con discapacidad?"

## 5. Planteamiento del Reto

¿Podemos crear un bastón inteligente que detecte obstáculos y ayude a una persona ciega a orientarse con mayor seguridad?

---

## 6. Desarrollo del Proyecto (Sesiones)

### S1: Introducción y formación de equipos

- Presentación del reto.
- Investigación sobre tecnologías para ciegos.
- Roles de equipo: programador, diseñador 3D, integrador.

### S2: Práctica 1 - Sensor ultrasónico HC-SR04

- Lectura de distancia con un sensor.
- Código básico + visualización en Serial Monitor.

```
#include <NewPing.h>
NewPing sonar(7, 6, 200);

void setup() {
  Serial.begin(9600);
}

void loop() {
  delay(50);
  Serial.print("Distancia: ");
  Serial.println(sonar.ping_cm());
}
```

### S3: Práctica 2 - Motor de vibración

- Activación de un motor con digitalWrite.

```
int motor = 3;

void setup() {
  pinMode(motor, OUTPUT);
}

void loop() {
  digitalWrite(motor, HIGH);
  delay(1000);
  digitalWrite(motor, LOW);
  delay(1000);
}
```

#### **S4: Integración 1 sensor + 1 motor**

- Vibración si la distancia es menor a 100 cm.

#### **S5: Sistema completo (3 sensores y 3 motores)**

- Sensor central, derecho e izquierdo.
- Vibrador central, derecho e izquierdo.

#### **S6: Diseño e impresión 3D**

- Bastón: compartimento inferior para sensores y Arduino Nano.
- Mango superior: cavidades para motores.

#### **S7: Ensamblado y pruebas**

- Montaje completo del prototipo.
- Pruebas en circuito simulado.

#### **S8: Prueba con usuario real y evaluación**

- Feedback del alumno ciego.
- Propuestas de mejora.

---

## 7. Producto Final

Bastón inteligente funcional, con sensores y motores integrados en diseño impreso en 3D, probado por un usuario real con discapacidad visual.

---

## 8. Evaluación

### **Formativa:**

- Rúbrica de trabajo en equipo y seguimiento por sesiones.
- Dianas de auto y coevaluación.

### **Final:**

- Evaluación funcional del prototipo con criterios CE1-CE3.
- Portafolio digital de desarrollo.
- Reflexión individual.

## 9. Recursos y Materiales

- Arduino Nano
- 3 sensores ultrasónicos HC-SR04
- 3 motores de vibración
- Resistencias, cables, protoboard
- Impresora 3D y filamento
- Ordenador con Arduino IDE

## 10. Adaptaciones y Diversificación

- Guías visuales paso a paso y esquemas ilustrados para TDAH/dislexia.
- Mejora del prototipo para altas capacidades: incorporar zumbadores o sensores adicionales.
- Alternativa: presentación audiovisual del proceso.

## 11. Estrategias de Motivación

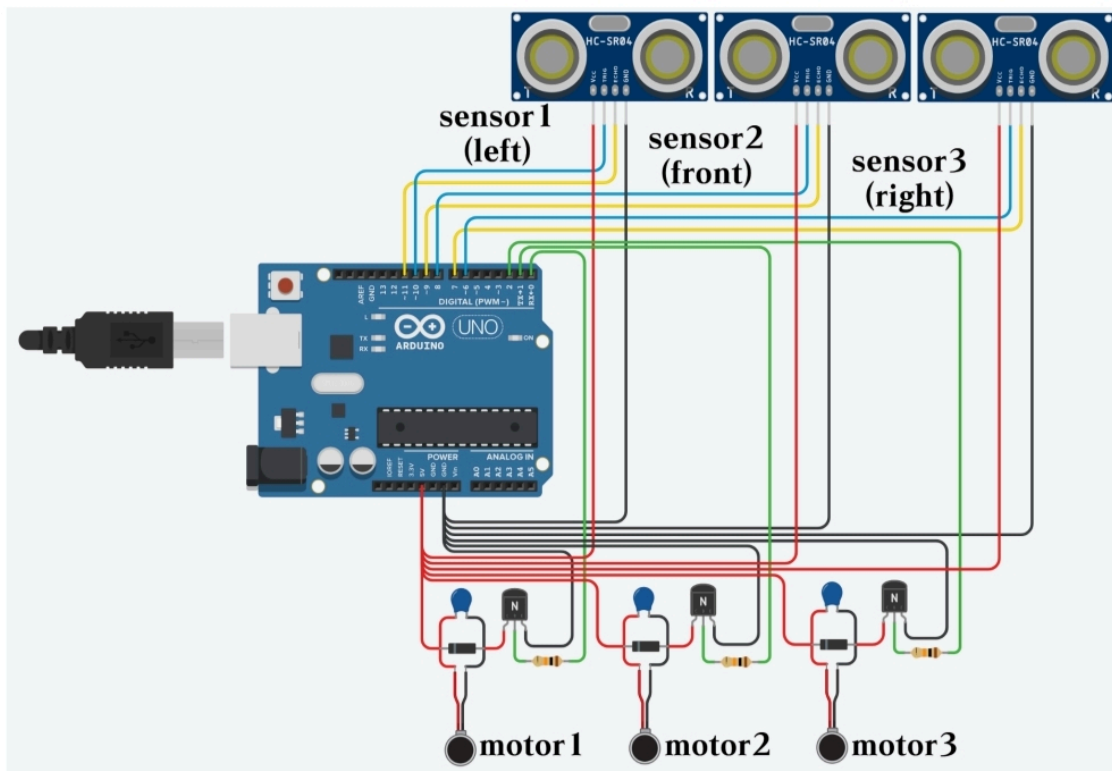
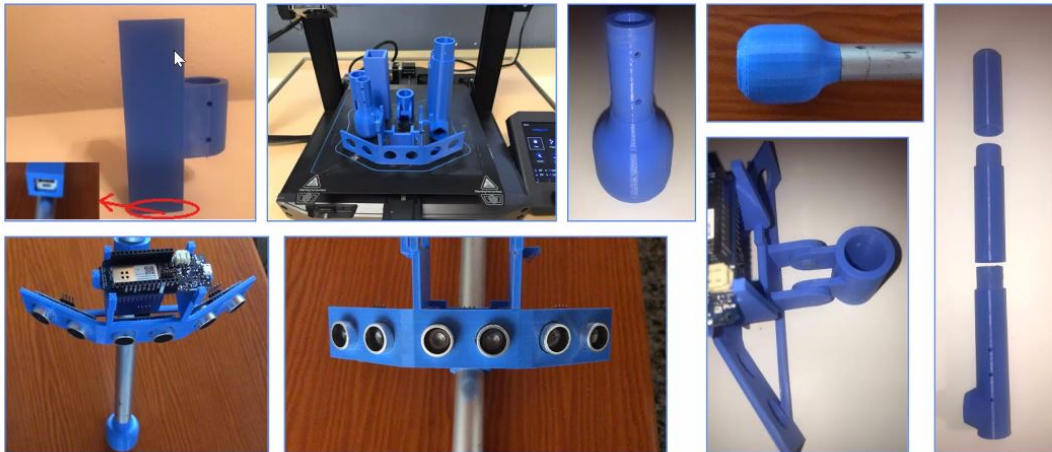
- Reto con impacto social directo.
- Protagonismo del alumno con discapacidad.
- Presentación final como "Feria de Prototipos para la Accesibilidad".

VIDEO DISPONIBLE ONLINE EN: <https://drfperez.github.io/baston>

IMÁGENES DEL PROYECTO







```

1 int trigPin1 = 10;
2 int echoPin1 = 11;
3 const int motor1 = 0;
4 int sensor1 = 0;
5
6 int trigPin2 = 8;
7 int echoPin2 = 9;
8 const int motor2 = 1;
9 int sensor2 = 0;
10
11 int trigPin3 = 6;
12 int echoPin3 = 7;
13 const int motor3 = 2;
14 int sensor3 = 0;
15
16 void setup()
17 {
18     Serial.begin(9600);
19
20     pinMode(trigPin1, OUTPUT);
21     pinMode(echoPin1, INPUT);
22
23     pinMode(trigPin2, OUTPUT);
24     pinMode(echoPin2, INPUT);
25
26     pinMode(trigPin3, OUTPUT);
27     pinMode(echoPin3, INPUT);
28
29     pinMode(motor1, OUTPUT);
30     pinMode(motor2, OUTPUT);
31     pinMode(motor3, OUTPUT);
32 }
33
34 void loop()
35 {
36     long duration1, distance1;
37     digitalWrite(trigPin1, LOW);
38     delayMicroseconds(2);
39     digitalWrite(trigPin1, HIGH);
40     delayMicroseconds(10);
41     digitalWrite(trigPin1, LOW);
42     duration1 = pulseIn(echoPin1, HIGH);
43     distance1 = (duration1/2) / 29.1;
44
45     if (distance1 <= 200 && distance1 >= 0)
46     {
47         sensor1 = map(distance1,0,200,255,0);
48         analogWrite(motor1,sensor1);
49     }
50
51     else if (distance1 >= 200 || distance1 <= 0)
52     {
53         sensor1 = 0;
54         analogWrite(motor1,0);
55     }
56
57     long duration2, distance2;
58     digitalWrite(trigPin2, LOW);
59     delayMicroseconds(2);
60     digitalWrite(trigPin2, HIGH);
61     delayMicroseconds(10);
62     digitalWrite(trigPin2, LOW);
63     duration2 = pulseIn(echoPin2, HIGH);
64     distance2 = (duration2/2) / 29.1;
65
66     if (distance2 <= 200 && distance2 >= 0)
67     {
68         sensor2 = map(distance2,0,200,255,0);
69         analogWrite(motor2,sensor2);
70     }
71
72     else if (distance2 >= 200 || distance2 <= 0)
73     {
74         sensor2 = 0;
75         analogWrite(motor2,0);
76     }
77
78     long duration3, distance3;
79     digitalWrite(trigPin3, LOW);
80     delayMicroseconds(2);
81     digitalWrite(trigPin3, HIGH);
82     delayMicroseconds(10);
83     digitalWrite(trigPin3, LOW);
84     duration3 = pulseIn(echoPin3, HIGH);
85     distance3 = (duration3/2) / 29.1;
86
87     if (distance3 <= 200 && distance3 >= 0)
88     {
89         sensor3 = map(distance3,0,200,255,0);
90         analogWrite(motor3,sensor3);
91     }
92
93     else if (distance3 >= 200 || distance3 <= 0)
94     {
95         sensor3 = 0;
96         analogWrite(motor3,0);
97     }
98
99     delay(200);
100 }

```

