



Actividades desarrolladas en equipo y en red.

1. Definición de la actividad.

Situación de aprendizaje
Título: Girasol electrónico
Descripción
<p>El alumnado debe dar solución a las instalaciones fotovoltaicas convencionales que suelen ser fijas y no aprovechan todo el ciclo solar con el máximo rendimiento. Teniendo como ejemplo la planta del girasol, que va girándose, mirando siempre perpendicularmente al sol, deben implementar un sistema para que las placas solares puedan realizar de la misma manera el mismo movimiento.</p> <p>Para esto, utilizarán un soporte realizado en 3D, un servo, unas resistencias LDR, una placa de control ESP32 y una pequeña placa solar.</p> <p>Para llegar al punto final, deberán realizar el montaje mecánico, el circuito electrónico y realizar varios programas de control para el servo, para las lecturas analógicas de las LDR, hasta llegar al montaje final.</p>

2. Actividades desarrolladas con trabajo en equipo.

Actividades
5.3.1 Montaje Mecánico
5.3.2 Montaje electrónico.
5.3.1 Iniciando la programación en uPython
5.3.2 Control de un servo
5.3.3 Medidas de magnitudes analógicas con upython
5.3.4 Control del girasol electrónico

3. Herramientas utilizadas

Herramientas de trabajo en equipo utilizadas	
Herramientas de comunicación	Correo de Outlook Plataforma Moodle



Herramientas de Planificación y seguimiento	Hoja de cálculo Excel
Herramientas para compartir documentación	Onedrive.
Herramientas para la creación de productos	MSWord 365
Otro tipo de herramientas	Thony

4. Dificultades y propuestas de mejora

Dificultades y propuestas de mejora
<p>No hemos tenido alumnado suficiente para tener grupos de 4 alumnos, por lo que los grupos se han formado con 2.</p> <p>Al desarrollar todo el trabajo en clase, los programas de comunicación tienen menos sentido, aunque creo que debemos orientar al alumnado a utilizar Microsoft Teams.</p>

5. Capturas del resultado del proceso.

Reparto de Roles

ROLES DEL GRUPO	
ROL	ALUMNADO
Jefe de Equipo	[Redacted]
Jefe de Recursos	[Redacted]
Jefe de Creatividad	[Redacted]
Jefe de Calidad	[Redacted]
FUNCIONES	
[Redacted]	
<p>Es el encargado de definir y organizar el trabajo del grupo, establecer los plazos, distribuir las actividades y facilitar la comunicación entre los integrantes. También debe resolver los posibles conflictos que surjan y motivar al equipo para lograr los resultados esperados.</p>	
[Redacted]	
<p>Es el responsable de compartir mediante la nube o herramientas digitales los documentos de trabajo con el resto del grupo. Debe comprobar que las fuentes utilizadas son confiables y están citadas correctamente. Una vez finalizado el trabajo se encargará de realizar la entrega de todos los documentos comunes del grupo.</p>	



Definición de tareas

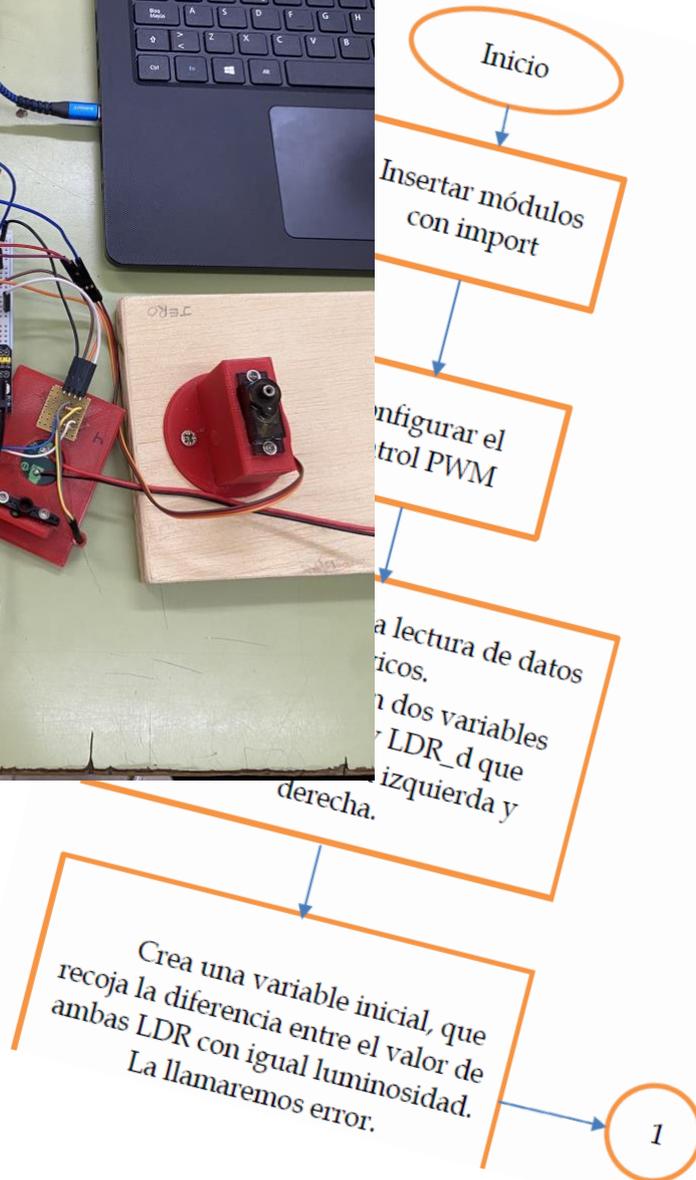
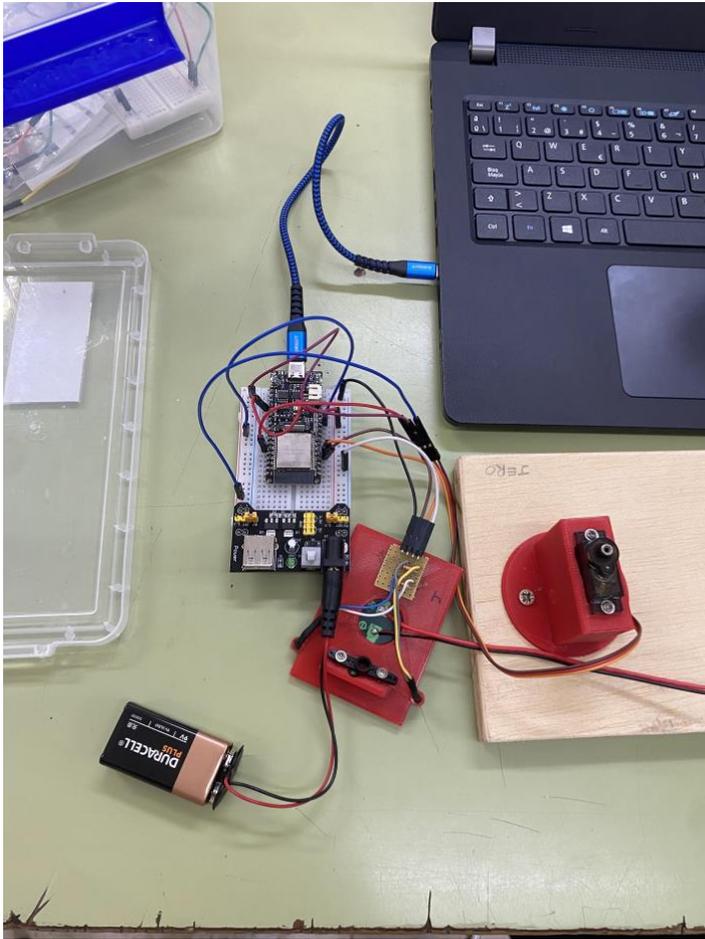
	DEFINICIÓN DE TAREAS
ACTIVIDAD 1	Repartir los roles
ACTIVIDAD 2	Compartir los documentos
ACTIVIDAD 3	Definir tareas
ACTIVIDAD 4	Medir y cortar madera
ACTIVIDAD 5	Mecanizado de las piezas 3D
ACTIVIDAD 6	Montar la placa solar en su soporte 3D
ACTIVIDAD 7	Atornillar el soporte a la madera
ACTIVIDAD 8	Realizar fotos a la tarea
ACTIVIDAD 9	Entregar la tarea
ACTIVIDAD 10	Montar un circuito electrónica en una placa Protoboard
ACTIVIDAD 11	Calcular valores de la LDR en distintas situaciones aplicando diferentes voltajes
ACTIVIDAD 12	Obtener la placa electrónica y los componentes.
ACTIVIDAD 13	Soldar los componentes.
ACTIVIDAD 14	Hacer las conexiones necesarias para montar el circuito.

Reparto de tareas y planificación (Diagrama de Gantt)





Productos elaborados.





5.3.4 Programa en Python para el control del girasol electrónico.

Nombre:



- Introduce una captura del programa. Añade en la tarea también un pequeño vídeo del funcionamiento. Esta actividad es grupal.

```
from machine import Pin, ADC, PWM
import time
ldr_i = ADC(Pin(36)) # LDR izquierda# Configuración de las entradas analógicas
ldr_d = ADC(Pin(39)) # LDR derecha
ldr_i.atten(ADC.ATTN_11DB) # Ajustar la atenuación para lectura de 0 a 3.3V
ldr_d.atten(ADC.ATTN_11DB)
# Configuración del servomotor
servo = PWM(Pin(14), freq=50) # Pin 14 para controlar el servo
# Calibración inicial: establecer el error en condiciones normales
error = abs(ldr_i.read() - ldr_d.read())
grados = 0
# Función para mover el servo
def set_servo(grados):
    angulo = int(27+(grados / 180) * 102) + 90 # Conversión de ángulo a ciclo de trabajo
    servo.duty(angulo)
# Posición inicial del servo
inicial = 0 # Posición media
set_servo(inicial)
# Bucle principal
while True:
    # Leer valores analógicos
    izquierda = ldr_i.read()
    derecha = ldr_d.read()
    # Calcular diferencia absoluta
    diferencia = abs(izquierda - derecha)
    if diferencia > error:
        # Determinar la dirección del movimiento
        if izquierda > derecha:
            grados = min(grados + 1, 180) # Mover hacia la izquierda
        elif izquierda < derecha:
            grados = max(grados - 1, 0) # Mover hacia la derecha
        set_servo (grados)
        time.sleep(0.05)
```

