# Portada

# Índice

# 

[Portada 1](#_b92vthnt0vn4)

[**Índice 1**](#_95xqmmuyvy96)

[**Introducción 2**](#_osp06tctfbux)

[Objetivo y descripción: 2](#_j4df1oi8yxsx)

[**Guía de usuario 4**](#_tgyedtvavbp8)

[CONOCIENDO LOS COMPONENTES DEL PROYECTO. 4](#_34yhem6xkkqj)

[● Arduino UNO o nano. 4](#_wolptz3wdw6i)

[● Placa solar. 5](#_x22z4yic5on3)

[● Fotorresistor/LDR. 6](#_d2q9e3zeyes4)

[● Servomotor. 8](#_ecvigjlmq8t)

[Mapa de flujo de la programación 10](#_qs1tuj5914a0)

[Código del proyecto 10](#_txtbjitbdowb)

[Montaje paso a paso de las piezas 10](#_bf2rjfhzw01p)

[Explicación del modelado 10](#_7yf80nj6l9u7)

[Propuestas de mejora 11](#_rxz9y5soysb7)

[**Guía de formador 12**](#_g0enb115z364)

[Introducción a los sensores 12](#_mn4qclyjwpyn)

[Mapa de flujo de la programación 12](#_fyaanql0l08s)

[Código del proyecto 12](#_5wjvf1vrxnnv)

[Montaje paso a paso de las piezas 12](#_htc6wiwcswj5)

[Explicación del modelado 12](#_poh2cx2wedsx)

[Explicación del modelado 12](#_8hh9qff6yvmr)

[Guía de usuario 14](#_6pha4q27zzar)

[Introducción a los sensores 14](#_ru3vd44hon2q)

[Mapa de flujo de la programación 14](#_2u7x3y8n8doo)

[Código del proyecto 14](#_1wsanw43q2ik)

[Montaje paso a paso de las piezas 14](#_3be3n83za5tr)

[Explicación del modelado 14](#_ewjud9jbkwmh)

[Guía de formador 15](#_dwdn1mn01i9j)

[Metodología de enseñanza 15](#_7jsiudhy3e5d)

[Procedimiento del proyecto 15](#_7s6v0chmikc)

[Fase 1: Corte láser de los archivos DWG 15](#_fpzvvqkpq5yv)

[Fase 2: Introducción a los sensores 15](#_rha14nbdhxk6)

[Fase 3: Código del proyecto 15](#_gbewtcv4hp1y)

[Fase 4: Montaje paso a paso de las piezas 15](#_dbhm11u3dqh8)

[Propuestas de mejora 16](#_1y2da4b6lc34)

# 

# 

# Introducción

## Objetivo y descripción:

En este proyecto se plantea la realización de una placa solar móvil. Se caracterizará por ser capaz de girar en busca de la mayor cantidad de luz solar con el objetivo de maximizar la energía solar que es capaz de adquirir.

Para ello en este proyecto se utilizará una placa solar que mediante distintos LDR (fotorresistores). El sistema sea capaz de decidir hacia donde hay mas luminosidad.

Tras la realización de este proyecto el alumno habrá adquirido nociones en…

| **Temática**  Impresión 3d  Corte láser  Programación básica  Electrónica básica | **Duración**: X h | |
| --- | --- | --- |
| **Materiales**   * POR DECIDIR | **Electrónica**   * Arduino * Placa solar (opcional) * 2x servo motores * Interruptor * Cables dupont * Protoboard * 4x resistencias * 4x LDR * volímetro |

Este proyecto consta de los siguientes elementos:

* Piezas impresas en 3D que supondrán un reto en la impresión 3d y el dominio de las máquinas.
* Piezas cortadas en metacrilato y madera que requerirán de conocimientos en corte láser en diversos materiales, potencias y velocidades.
* Utilización de servo motores y sensores LDR.
* Programación básica basada en bloques o Arduino, a través de diversas herramientas de programación.

Y su funcionamiento es el siguiente:

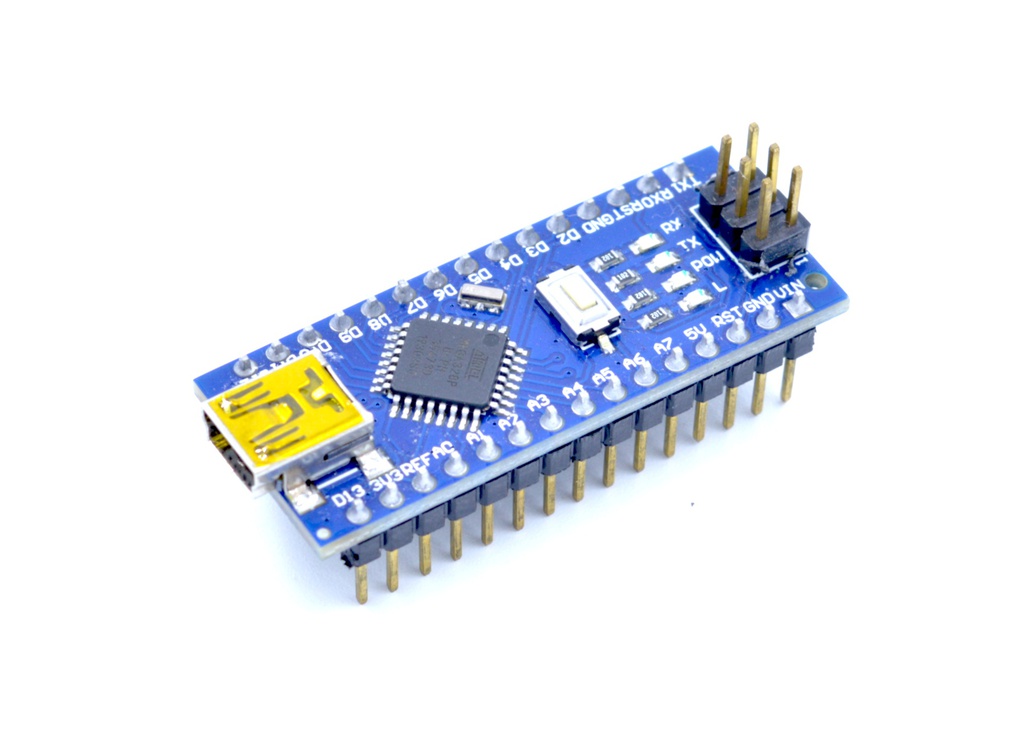
# 

# 

# Guía de usuario

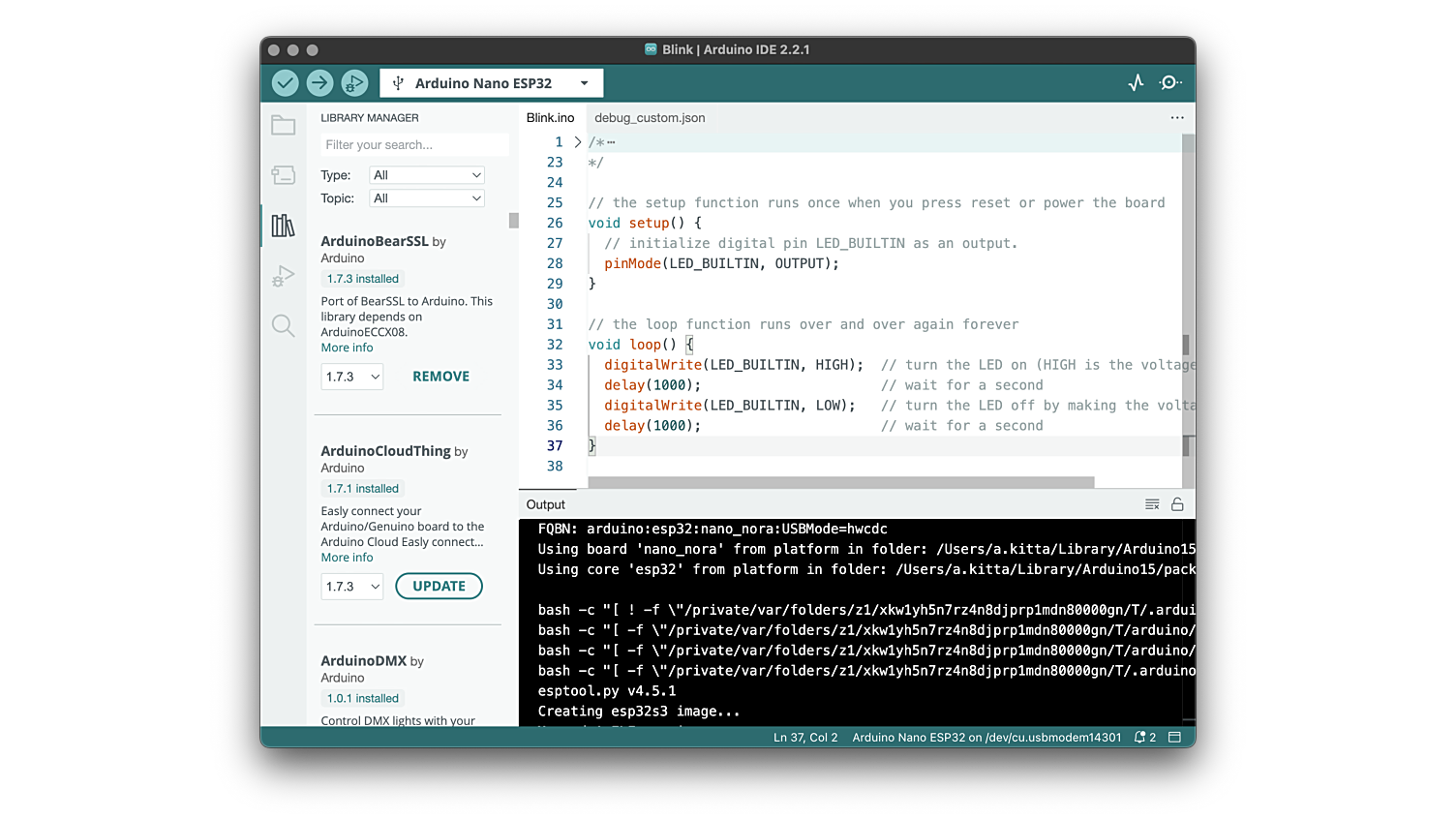
## CONOCIENDO LOS COMPONENTES DEL PROYECTO.

### Arduino UNO o nano.



Arduino es una plataforma de hardware y software de código abierto diseñada para facilitar el desarrollo de proyectos electrónicos interactivos. Consiste en una placa de circuito impreso con un microcontrolador y un entorno de desarrollo integrado (IDE) que permite programar el microcontrolador de manera sencilla.

Para la realización de este proyecto no importa utilizar la versión UNO o NANO, ya que ambas versiones poseen los pines necesarios para el proyecto. Cabe destacar que la versión NANO, como su nombre indica es una versión más pequeña de arduino con un consumo energético y potencia menor respecto a us versión UNO.

Para programar estos dispositivos debemos utilizar el arduino IDE. Se trata de un software capaz de transformar código entendible para los humanos en un código binario que posee instrucciones para Arduino. Además de realizar esta conversión de código se encargará de introducir en la memoria de nuestro arduino el código. 

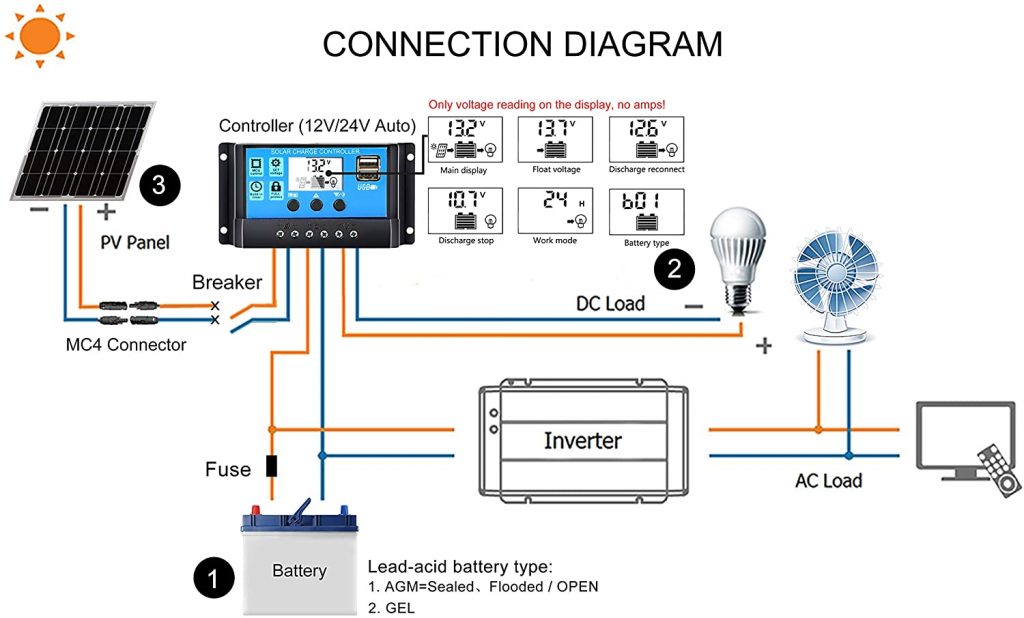
### Placa solar.

Como ya sabréis y debido a la fama que ha logrado en los últimos años, se trata de un dispositivo capaz de transformar la energía solar en energía eléctrica.

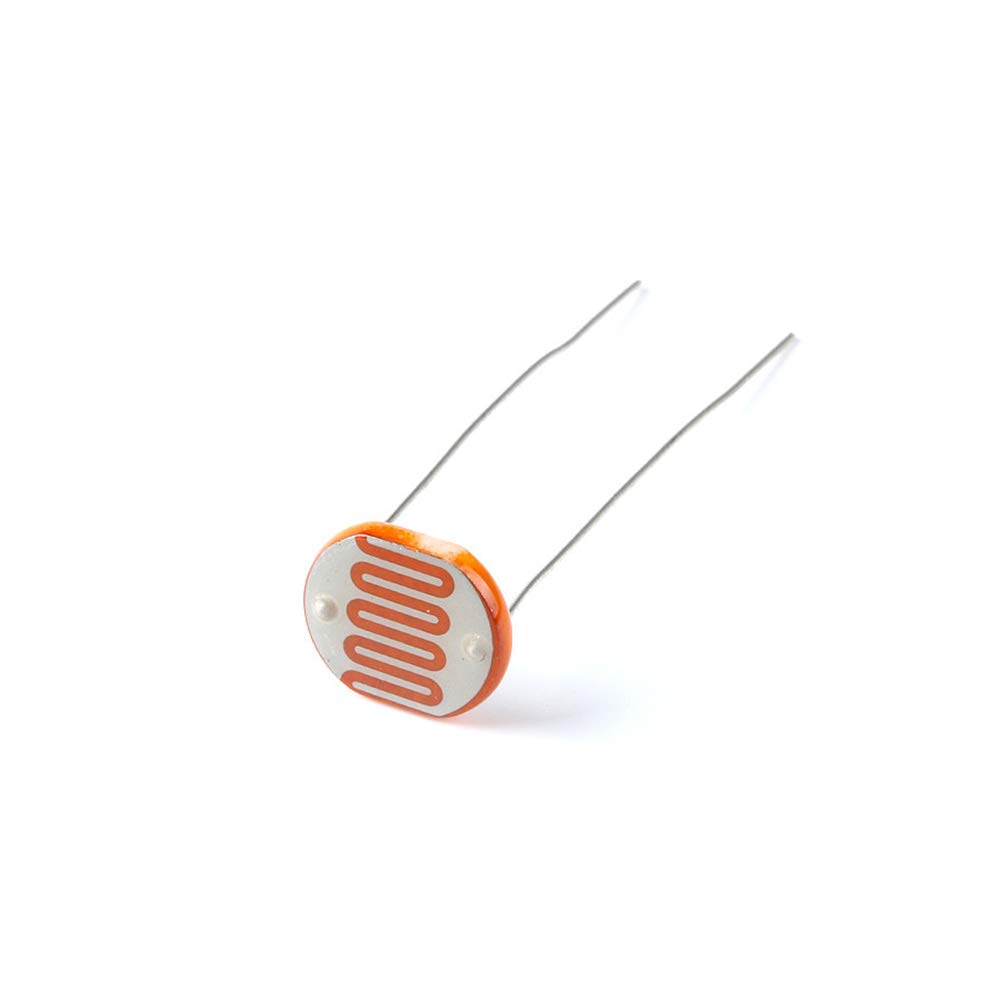
Es importante conocer ciertas características de las placas solares si queremos utilizarlas para distintos proyectos:

Por un lado, las placas solares están diseñadas para colocarse orientadas al sur por la posición del sol en el hemisferio norte y orientadas al norte en el hemisferio sur.

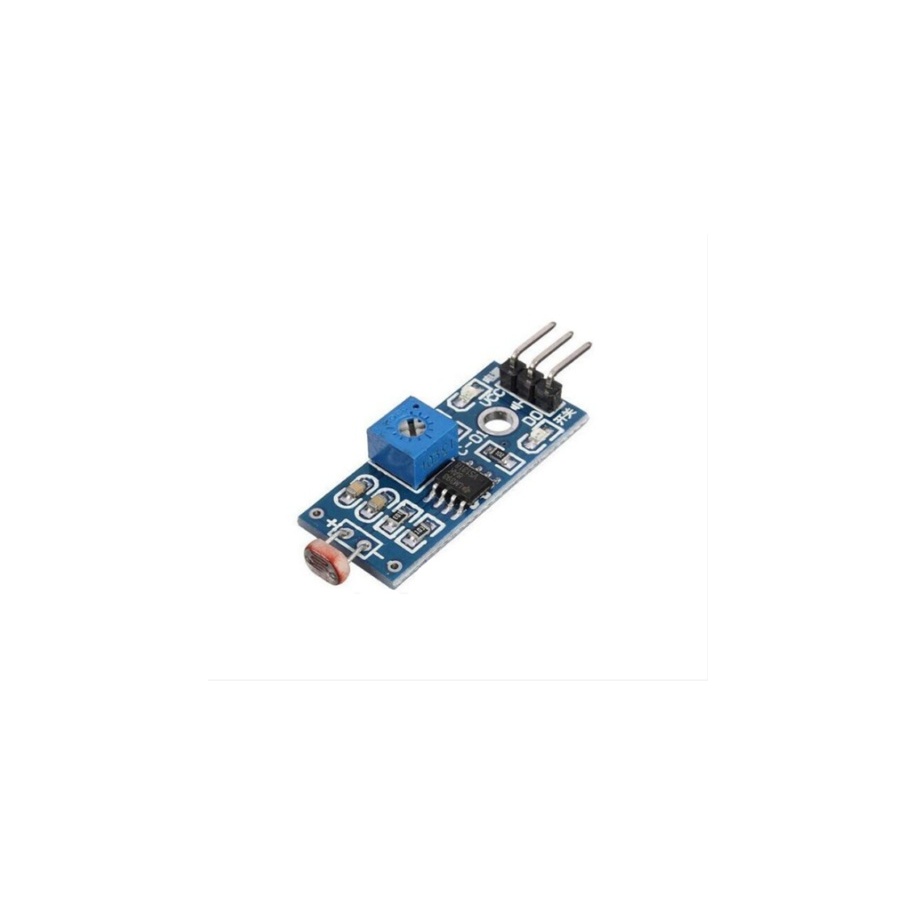
Por lo general, si queremos utilizar una placa solar como alimentación para nuestro proyecto es importante tener en cuenta que una placa solar de 12V no proporciona constantemente 12V. Dependiendo de ciertos factores (la hora, la climatología…) es muy probable que la placa solar suministre un voltaje menor del que debería. Por ello es muy recomendable utilizar la placa solar acompañada de una batería y un regulador de carga solar para alimentar nuestros proyectos.



### Fotorresistor/LDR.

Un LDR, o "Resistor Dependiente de la Luz" en español, es un componente electrónico que cambia su resistencia eléctrica en respuesta a la intensidad de la luz incidente. Su nombre completo en inglés es "Light-Dependent Resistor". Este dispositivo tiene la capacidad de variar su resistencia en función de la cantidad de luz que incide sobre él.

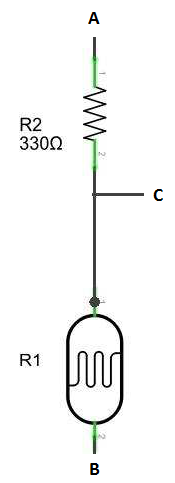
Cuando se expone a una mayor cantidad de luz, la resistencia del LDR disminuye, permitiendo que pase más corriente a través de él. Por el contrario, en condiciones de poca luz, la resistencia del LDR aumenta, reduciendo el flujo de corriente. Esto hace que el LDR sea útil en circuitos electrónicos para detectar o medir la luz ambiental.

En este proyecto utilizaremos 4 LDR que se encargarán de medir la luz solar incidente en 4 posiciones distintas y nuestro microcontrolador decidirá hacia donde se debe mover la placa solar para poder alcanzar la posición de máxima incidencia de luz, es decir la posición en el que los 4 LDR en este caso posean el mismo valor de intensidad lumínica.

**IMPORTANTE:** Para la realización de este proyecto vamos a leer valores analógicos, es por eso, que no podemos utilizar los módulos LDR que poseen salida digital. Ya que estos, solo interpretaran dos posibles estados de luminosidad (0 o 1). Por el contrario si el módulo posee 4 pines (GND, VCC, DO , AO) si podría utilizarse, ya que este posee salida analógica (capaz de interpretar valores contínuos entre 3,3V y 5V).

A continuación se explica el principio de utilización para un LDR.

Para poder utilizar un LDR, necesitamos aplicar lo que se conoce en electrónica como un divisor de tensión.

Un divisor de tensión es un circuito eléctrico compuesto por dos resistencias conectadas en serie. La tensión de salida se toma en el punto de conexión entre las dos resistencias (lugar donde realizaremos la medición analógica). La relación entre las resistencias determina la proporción en la que se divide la tensión de entrada.

En el caso de la lectura de valores de un fotorresistor, se utiliza un divisor de tensión para adaptar el rango de voltajes al rango de operación del dispositivo de lectura, nuestro arduino. Los fotorresistores cambian su resistencia en función de la intensidad luminosa. Al utilizar un divisor de tensión, se puede obtener una señal de voltaje proporcional al cambio en la resistencia del fotorresistor, sin exceder los límites de voltaje admisibles del dispositivo de lectura.

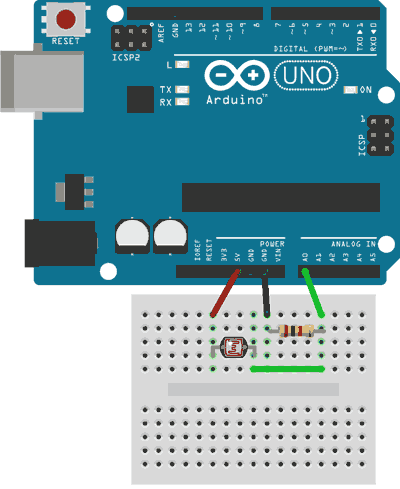
Lo que debemos hacer es conectar en serie una resistencia con nuestro fotorresistor. Tal y como se muestra en la imagen.

Para nuestro sistema es común la utilización de resistencias entre 5 y 10 kohm. Es importante que los fotorresistores para el proyecto sean los 4 iguales ya que cada fotorresistor tendrá un valor de resistencia ante un mismo valor de luminosidad.

Es importante que los LDR de distintos fabricantes proporcionan un valor de resistencia en ohmios distintos en condiciones luminosas.

EJERCICIO 1. Realizar la lectura de un LDR y mostrar por serial sus valores.

* Montaje.



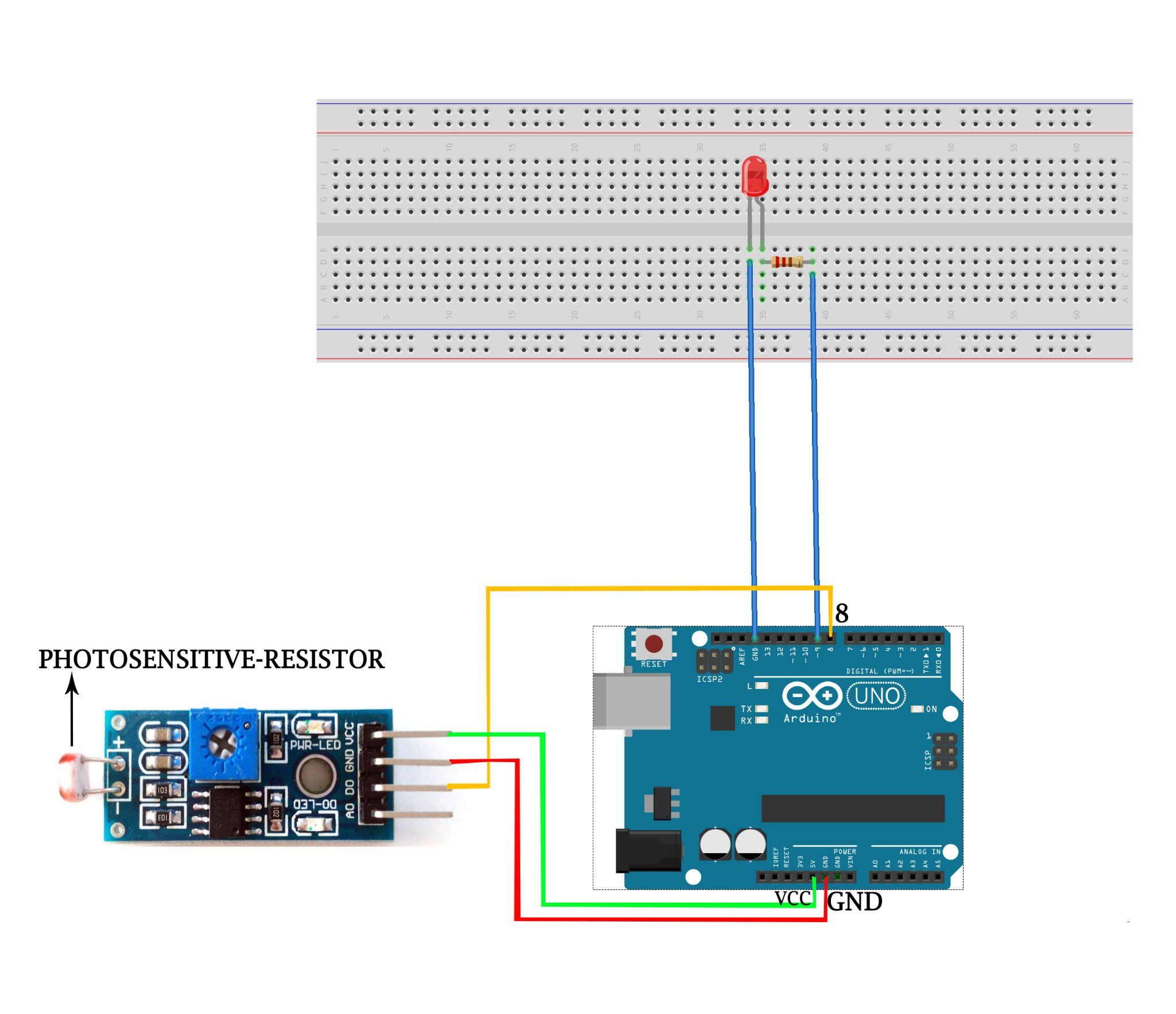
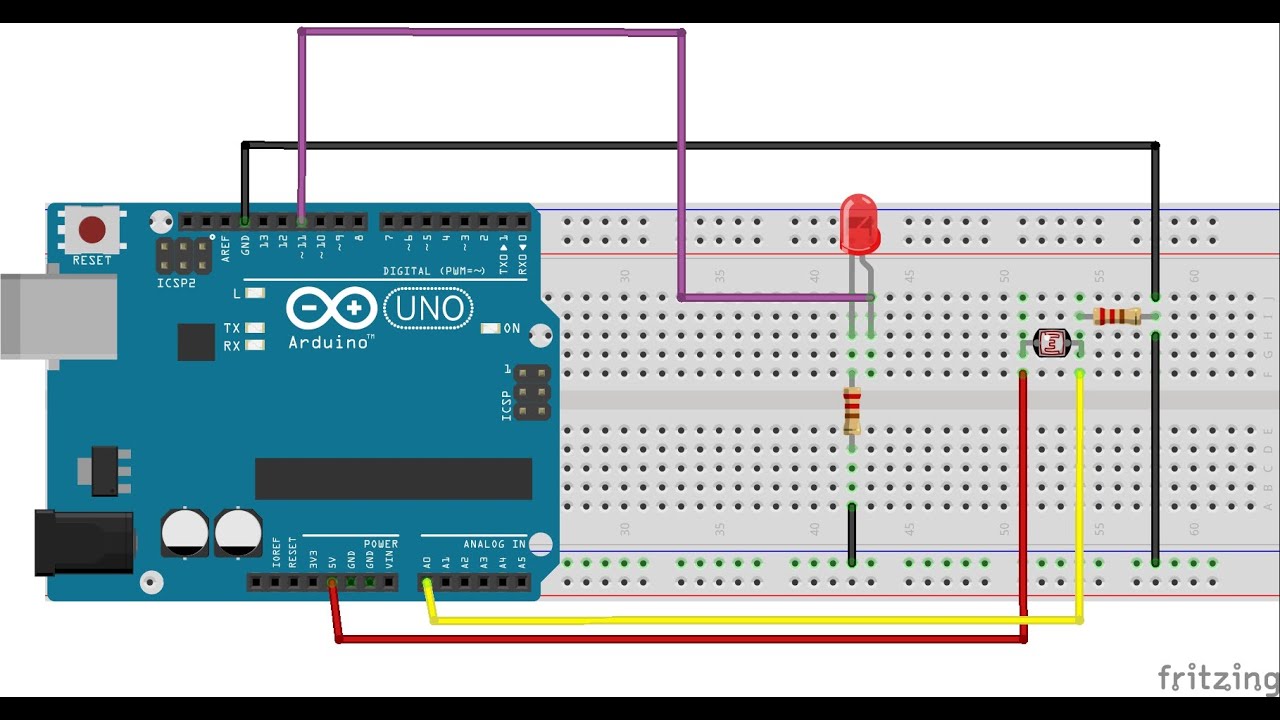
Código:

| const long A = 1000; //Resistencia en oscuridad en KΩ  const int B = 15; //Resistencia a la luz (10 Lux) en KΩ  const int Rc = 10; //Resistencia calibracion en KΩ  const int LDRPin = A0; //Pin del LDR  int V;  int ilum;  void setup()  {  Serial.begin(115200);  }  void loop()  {  V = analogRead(LDRPin);  //ilum = ((long)(1024-V)\*A\*10)/((long)B\*Rc\*V); //usar si LDR entre GND y A0  ilum = ((long)V\*A\*10)/((long)B\*Rc\*(1024-V)); //usar si LDR entre A0 y Vcc (como en el esquema anterior)    Serial.println(ilum);  delay(1000);  } |
| --- |

**EJERCICIO 2**.

Realizar un pequeño programa con arduino que sea capaz de encender un led cuando la lectura del sensor indique baja luminosidad en la sala y se mantenga apagado cuando hay alta luminosidad en la sala.

Posibles montajes(o bien con un LDR solo o con un módulo que incorpora el LDR y facilita la tarea):



Posible solución al ejercicio:

| const int ldrPin = A0; // Pin analógico donde está conectado el LDR  const int ledPin = 13; // Pin donde está conectado el LED  void setup() {  pinMode(ldrPin, INPUT); // Configura el pin del LDR como entrada  pinMode(ledPin, OUTPUT); // Configura el pin del LED como salida  }  void loop() {  int ldrValue = analogRead(ldrPin); // Lee el valor analógico del LDR  delay(100); // Pequeña pausa para estabilizar la lectura  // Puedes ajustar estos valores según tus condiciones de luz  if (ldrValue < 300) {  digitalWrite(ledPin, HIGH); // Enciende el LED cuando la luminosidad es baja  } else {  digitalWrite(ledPin, LOW); // Apaga el LED cuando la luminosidad es alta  }  } |
| --- |

### Servomotor.

Un servomotor es un dispositivo que permite controlar y posicionar un eje en ángulos específicos. A diferencia de los motores convencionales, los servomotores están diseñados para un control preciso de la posición angular. Estos dispositivos son comúnmente utilizados en proyectos y dispositivos que requieren movimiento controlado, como juguetes, robots, modelismo, y otras aplicaciones.

La característica principal de un servomotor es su capacidad para girar hasta un ángulo determinado en respuesta a señales eléctricas. La mayoría de los servomotores comerciales para proyectos electrónicos tienen un rango de movimiento de 0 a 180 grados, aunque también existen modelos que pueden girar 360 grados.

En nuestro proyecto utilizaremos dos servomotores, uno que se encargue del giro de la placa solar en vertical y otro en horizontal. Este movimiento que realizarán será en intervalos de 10º para ahorrar energía y simplificar la posible optimización de la incidencia de luz solar. No importa si el servomotor es de 180º o de 360º, realizaremos un diseño pensando en un servomotor de 180º (es el más restrictivo).

Estos motores suelen poseer los siguientes pines:

* Un pin de alimentación (5V).
* Un pin de GND/tierra/(-)
* Un pin PWM: Pin que le proporciona la señal digital que le indicará al servomotor la posición a la que debe girar.

Cabe destacar que para su conexión con arduino no vale cualquier pin digital, deben de ser pines PWM. "PWM" significa "Modulación de Ancho de Pulso" en inglés (Pulse Width Modulation). Es una técnica utilizada en electrónica para controlar la cantidad de energía entregada a un dispositivo, mediante la variación del ancho de los pulsos de una señal eléctrica. Posee un funcionamiento distinto al del resto de pines digitales ya que no se encargan simplemente de transmitir un 1(HIGH) o 0 (LOW). Hay que tener en cuenta que estos pines pueden usarse como pines digitales comunes.

Para saber cuáles son los pines PWM, consulta la tabla de pines del arduino que se vaya a utilizar.

NOTA: Si estás utilizando un arduino nano, los pines GPIO que admiten PWM son aquellos pines digitales que poseen el simbolo (~)

Para simplificar la programación de los servomotores se utilizará la librería de arduino *servo.h*

A continuación se muestra un pequeño programa que realiza una demostración del funcionamiento de los servomotores.

Montaje:

### 

Código:

| #include <Servo.h>  Servo miServo; // Crear un objeto de tipo Servo  void setup() {  miServo.attach(9); // El pin al que está conectado el servo (ajusta según tu conexión)  }  void loop() {  for (int angulo = 0; angulo <= 180; angulo += 1) {  miServo.write(angulo); // Mover el servo al ángulo actual  delay(15); // Pequeña pausa para dar tiempo al servo  }  delay(1000); // Pausa de 1 segundo al final del recorrido  for (int angulo = 180; angulo >= 0; angulo -= 1) {  miServo.write(angulo); // Mover el servo al ángulo actual  delay(15); // Pequeña pausa para dar tiempo al servo  }  delay(1000); // Pausa de 1 segundo al final del recorrido  } |
| --- |

# ENUNCIADO DEL PROYECTO.

El propósito final de este proyecto es realizar un control guiado, usando arduino, fotorresistores y servomotores, de una placa solar para optimizar su obtención de energía.(Maximizar el voltaje que recibe la placa).

Este voltaje no vamos a medirlo directamente, en cambio si vamos a medir la intensidad de luz recibida por los fotorresistores.

Es necesario que el sistema se actualice dado un intervalo de tiempo concreto ya que, en principio, el sol no cambia de posición en cuestión de segundos. Para ello tambien haremos que el sistema busque el sol de forma aproximada sin pretender estar totalmente Perpendicular a su fuente de luz. Es decir, los cambios de dirección se harán en intervalos de 1º-5º de giro y con una tolerancia alta a la medida de los fotorresistores.

| #include <Servo.h> // Incluye la librería para el control de servos  // Definición de pines para los LDR (Fotodiodos)  #define PIN\_LDR\_SUPERIOR\_DERECHA A0  #define PIN\_LDR\_INFERIOR\_DERECHA A1  #define PIN\_LDR\_SUPERIOR\_IZQUIERDA A2  #define PIN\_LDR\_INFERIOR\_IZQUIERDA A3  // Definición de pines para los servomotores  #define PIN\_SERVO\_V 9 // Pin para el servo vertical  #define PIN\_SERVO\_H 10 // Pin para el servo horizontal  // Constantes de configuración  #define ANG 5 // Ángulo de giro para el movimiento de los servos  #define TOL 20 // Tolerancia para el movimiento del seguidor solar  #define POS\_SUP\_MAX 180 // Posición máxima para el servo vertical  #define POS\_INF\_MAX 0 // Posición mínima para el servo vertical  // Declaración de objetos servo  Servo servoHorizontal; // Servo para el movimiento horizontal  Servo servoVertical; // Servo para el movimiento vertical  // Variables para almacenar los valores de los LDR  int VLdrSD, VLdrSI, VLdrID, VLdrII;  // Configuración inicial  void setup() {  Serial.begin(9600); // Inicia la comunicación serial a 9600 baudios  // Configuración de los pines de los LDR como entradas  pinMode(PIN\_LDR\_SUPERIOR\_DERECHA, INPUT);  pinMode(PIN\_LDR\_INFERIOR\_DERECHA, INPUT);  pinMode(PIN\_LDR\_SUPERIOR\_IZQUIERDA, INPUT);  pinMode(PIN\_INFERIOR\_IZQUIERDA, INPUT);  // Inicialización de los servomotores  servoHorizontal.attach(PIN\_SERVO\_H); // Conecta el servo horizontal al pin 10  servoVertical.attach(PIN\_SERVO\_V); // Conecta el servo vertical al pin 9  servoHorizontal.write(90); // Establece la posición inicial del servo horizontal en 90 grados  servoVertical.write(90); // Establece la posición inicial del servo vertical en 90 grados  }  // Bucle principal  void loop() {  // Medición del valor analógico de cada LDR  VLdrSD = analogRead(PIN\_LDR\_SUPERIOR\_DERECHA);  VLdrID = analogRead(PIN\_LDR\_INFERIOR\_DERECHA);  VLdrSI = analogRead(PIN\_LDR\_SUPERIOR\_IZQUIERDA);  VLdrII = analogRead(PIN\_INFERIOR\_IZQUIERDA);    // Cálculo de la media de los valores de los LDR  int mSuperior = (VLdrSI + VLdrSD) / 2;  int mInferior = (VLdrII + VLdrID) / 2;  int mDerecha = (VLdrSD + VLdrID) / 2;  int mIzquierda = (VLdrII + VLdrSI) / 2;  // Llamada a la función para mover los servomotores  realizarMovimientoMotores(mSuperior, mInferior, mDerecha, mIzquierda);  delay(1000); // Retardo de 1 segundo  }  // Función para mover los servomotores según la posición de los LDR  void realizarMovimientoMotores(int mSuperior, int mInferior, int mDerecha, int mIzquierda) {  float mTotal = (mSuperior + mInferior + mIzquierda + mDerecha) / 4;  // Movimiento vertical del seguidor solar  if ((mSuperior - mInferior) > TOL && servoVertical.read() < POS\_SUP\_MAX) {  servoVertical.write(servoVertical.read() + ANGULO\_DE\_GIRO); // Movimiento hacia arriba  } else if ((mInferior - mSuperior) > TOL && servoVertical.read() > POS\_INF\_MAX) {  servoVertical.write(servoVertical.read() - ANGULO\_DE\_GIRO); // Movimiento hacia abajo  }  // Movimiento horizontal del seguidor solar  if ((mIzquierda - mDerecha) > TOL && servoHorizontal.read() < POS\_SUP\_MAX) {  servoHorizontal.write(servoHorizontal.read() + ANGULO\_DE\_GIRO); // Movimiento hacia la izquierda  } else if ((mDerecha - mIzquierda) > TOL && servoHorizontal.read() > POS\_INF\_MAX) {  servoHorizontal.write(servoHorizontal.read() - ANGULO\_DE\_GIRO); // Movimiento hacia la derecha  }  } |
| --- |

PASO 2.

## 

## 

## Código del proyecto

Bien comentado y estructurado

## Montaje paso a paso de las piezas

Montaje del proyecto:

## Explicación del modelado

Proceso de modelado de los componentes:

## Propuestas de mejora

#### Para este proyecto se propone una mejora en la cual mediante el uso de dos potenciómetros pueda controlarse manualmente la orientación de la placa solar.

# Guía de formador

## Introducción a los sensores

En este proyecto emplearemos el sensor X que emplea este tipo de entradas o salidas y funciona del siguiente modo. Necesitaremos para controlarlo X biblioteca y los comandos que emplearemos serán estos:

# 

## Mapa de flujo de la programación

Este es el diagrama de secuencia del proyecto:

Inicio

|

|---- Configuración Inicial

| |

| |---- Inicialización de comunicación serial

| |

| |---- Configuración de pines de LDR como entradas

| |

| |---- Inicialización de servomotores

| |

| |---- Conexión de servo horizontal al pin 10

| |

| |---- Conexión de servo vertical al pin 9

| |

| |---- Establecimiento de posición inicial de servos en 90 grados

|

|---- Bucle Principal

| |

| |---- Medición de valores analógicos de cada LDR

| |

| |---- Cálculo de media de valores de los LDR

| |

| |---- Llamada a función para mover los servomotores

| |

| |---- Movimiento vertical del seguidor solar

| | |

| | |---- Si diferencia entre medias superior e inferior > TOL y posición vertical < POS\_SUP\_MAX:

| | | |

| | | |---- Mover servo vertical hacia arriba (incrementar ángulo)

| | |

| | |---- Si diferencia entre medias inferior y superior > TOL y posición vertical > POS\_INF\_MAX:

| | |

| | |---- Mover servo vertical hacia abajo (decrementar ángulo)

| |

| |---- Movimiento horizontal del seguidor solar

| |

| |---- Si diferencia entre medias izquierda y derecha > TOL y posición horizontal < POS\_SUP\_MAX:

| | |

| | |---- Mover servo horizontal hacia la izquierda (incrementar ángulo)

| |

| |---- Si diferencia entre medias derecha e izquierda > TOL y posición horizontal > POS\_INF\_MAX:

| |

| |---- Mover servo horizontal hacia la derecha (decrementar ángulo)

|

Fin

## Código del proyecto

Bien comentado y estructurado

## Montaje paso a paso de las piezas

Montaje del proyecto:

## Explicación del modelado

Proceso de modelado de los componentes:

## Explicación del modelado

# 

# 

# 

# 

# Guía de usuario

## Introducción a los sensores

En este proyecto emplearemos el sensor X que emplea este tipo de entradas o salidas y funciona del siguiente modo. Necesitaremos para controlarlo X biblioteca y los comandos que emplearemos serán estos:

## Mapa de flujo de la programación

Este es el diagrama de secuencia del proyecto:

## Código del proyecto

Bien comentado y estructurado

## Montaje paso a paso de las piezas

Montaje del proyecto:

## Explicación del modelado

Proceso de modelado de los componentes:

# 

# Guía de formador

## Metodología de enseñanza

Este es un proyecto que combina muchas tecnologías. Como formador, deberás asegurarte que el usuario entienda las diversas máquinas y herramientas que va a utilizar a lo largo del proceso.

Se pueden realizar formaciones a lo largo de la realización del proyecto para completar esos conocimientos o realizarlas todas con anterioridad para que este sea como un proyecto donde los usuarios puedan comprobar su pericia a la hora de utilizar las diversas tecnologías.

Como formador, deberás tener en cuenta las dudas que puedan llegar a surgir pero no debes realizar tú la tarea, sino que sean ellos los que vayan resolviendo los problemas a través del autoaprendizaje.

## Procedimiento del proyecto

En esta guía se dará un orden donde se empezará con la tecnología con la que ellos estén familiarizados. Aunque en esta guía se realice un camino a recorrer para la resolución del proyecto, no se tiene por qué seguir el orden al ser partes independientes unas de otras.

### Fase 1: Corte láser de los archivos DWG

Para

### Fase 2: Introducción a los sensores

En este proyecto emplearemos el sensor X que emplea este tipo de entradas o salidas y funciona del siguiente modo. Necesitaremos para controlarlo X biblioteca y los comandos que emplearemos serán estos:

### Fase 3: Código del proyecto

Bien comentado y estructurado

### Fase 4: Montaje paso a paso de las piezas

Montaje del proyecto:

## 

# Propuestas de mejora

## 