

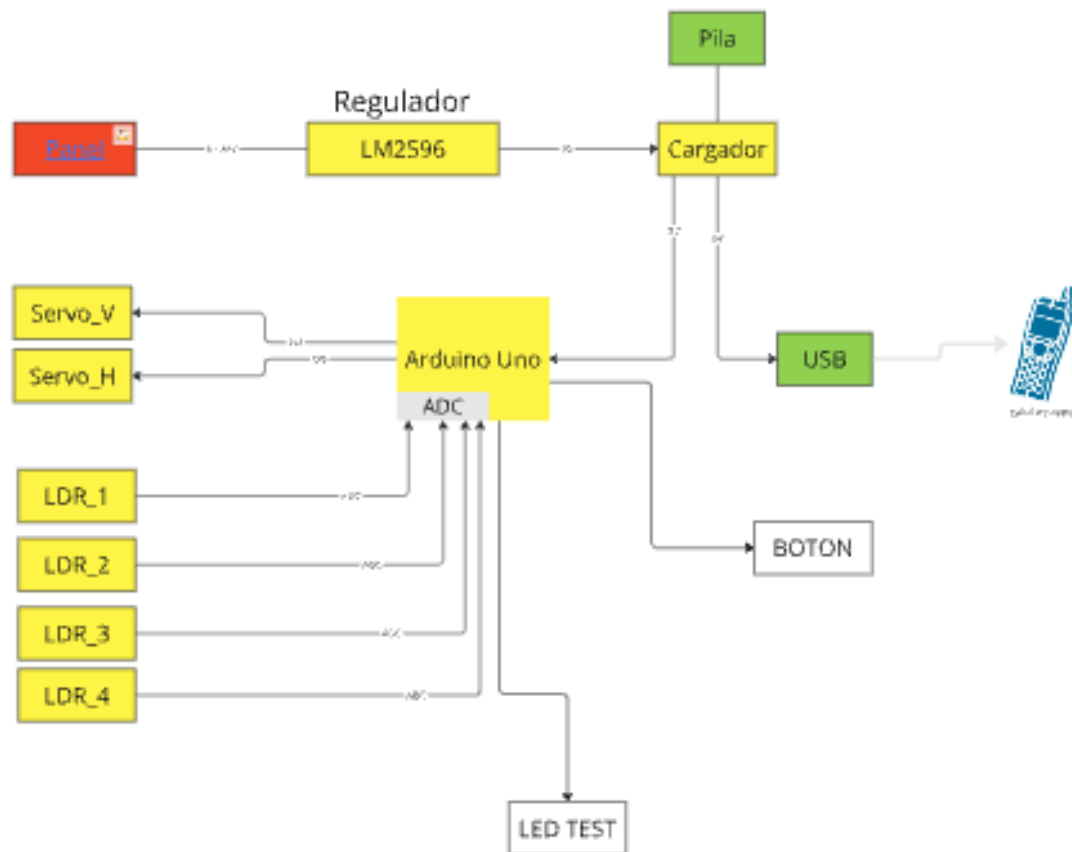
# Documentación del Proyecto de Software del Cargador de Baterías con Panel Solar y Control de LDRs

## Idea General:

Este proyecto consiste en el desarrollo de un sistema de carga de batería solar que incluye un mecanismo de seguimiento de la luz solar.

El sistema está diseñado para optimizar la captación de energía solar mediante la orientación automática de un panel solar en función de la intensidad de la luz solar detectada.

Se utilizan sensores de luz (LDRs Captadores de Luz) para identificar la mejor posición del panel y servomotores para ajustar su orientación. Este enfoque maximiza la eficiencia de la carga de la batería, especialmente en áreas donde la luz solar varía en dirección e intensidad durante el día.



## Objetivos:

El objetivo del proyecto es diseñar e implementar un sistema eficiente y autónomo que aproveche al máximo la luz solar para cargar baterías.

El sistema pretende:

**Optimizar la captación de luz solar:** mediante el seguimiento continuo de la posición del sol.

**Cargar Baterías/Pilas solares de forma eficiente:** asegurando que la energía solar captada sea utilizada de la mejor manera para cargar la batería.

**Promover el uso de energías renovables:** facilitando una solución accesible para usuarios en zonas rurales o remotas, sin acceso confiable a la red eléctrica.

## Usuarios:

El sistema está dirigido a usuarios que requieren una solución autónoma para cargar baterías utilizando energía solar, tales como:

- Agricultores en zonas rurales.
- Personas que habitan en áreas remotas sin acceso a la red eléctrica.
- Entusiastas de la energía renovable que buscan alternativas de carga fuera de la red convencional.

Este sistema está diseñado para usuarios con conocimientos básicos en la instalación de equipos solares, pero no requiere un alto nivel técnico para su operación.

## Especificación de Requerimientos

### Requisitos Generales:

Este proyecto está pensado para ser eficiente en el uso de energía y funcionar con distintos tipos de baterías recargables. Usa tecnología sencilla y componentes de bajo consumo, como:

- **Microcontrolador ARDUINO-UNO:** Es como el "cerebro" del sistema. Se encarga de recibir las señales de los sensores que detectan la luz del sol y, con esa información, controla unos motores que mueven el panel solar.
- **Panel solar:** Este es el encargado de captar la energía del sol y usarla para cargar una batería.
- **Servomotores:** Son los que se encargan de mover el panel solar. Ajustan su posición según donde esté el sol, para que siempre apunte hacia la mejor dirección para captar la mayor cantidad de luz solar.
- **LDRs (Resistencias Dependientes de la Luz):** Son unos sensores que detectan cuánta luz solar hay. Le dicen al sistema hacia dónde está el sol para que el panel pueda moverse y aprovecharlo mejor.

### **Requisitos Funcionales (Lo que debe hacer el sistema):**

- Detectar la posición del sol: El sistema siempre está leyendo lo que le dicen los LDRs para saber hacia dónde está brillando más el sol.
- Ajustar el panel solar automáticamente: Según lo que detectan los LDRs, el sistema mueve los motores para que el panel solar apunte hacia donde está la mayor cantidad de luz.
- Cargar la batería de manera eficiente: La energía que capta el panel solar tiene que usarse de la mejor forma posible para cargar la batería que está conectada.
- Monitorear la batería: El sistema tiene que estar pendiente de cuánta energía tiene la batería, para asegurarse de que se está usando correctamente y que no se descargue por completo.

### **Alcances (Hasta dónde llega el proyecto):**

- Carga limitada: Este sistema está pensado para cargar baterías pequeñas o medianas, lo que lo hace ideal para dispositivos que no necesitan mucha energía.
- No almacena gran cantidad de energía: No está diseñado para almacenar grandes cantidades de energía, como lo haría un sistema con muchas baterías.
- Sin monitoreo remoto: No tiene la capacidad de revisar o controlar el sistema a distancia o mediante alguna aplicación o pantalla avanzada.

## **Procedimientos de Instalación y Prueba**

### **Obtención e Instalación:**

1. **Requisitos previos:**
  - **Microcontrolador ARDUINO-UNO:** El sistema se implementará en este dispositivo, que debe estar configurado para ser programado desde el **Arduino IDE**.
  - **Entorno Arduino IDE:** Se necesita instalar Arduino IDE en el equipo del usuario. Además, debe agregarse la placa Arduino desde el gestor de placas de Arduino.  
Una vez añadida, seleccione la placa Arduino en el menú del IDE.
2. **Librerías necesarias:**
  - **Servo.h:** Librería para el control de los servomotores.
  - **Librería para lectura de LDRs:** Aunque la lectura de los LDRs se hace mediante entradas analógicas estándar, asegurarse de que el código utilice correctamente las funciones de Arduino para manejar estas entradas.
3. **Instalación del código:**
  - Una vez el Arduino IDE esté configurado y las librerías necesarias instaladas, se debe conectar el ARDUINO-UNO al equipo mediante un cable USB.
  - Cargar el código en el IDE y seleccionar el puerto COM correspondiente al ARDUINO-UNO.
  - Compilar y subir el código al microcontrolador.

## Prueba del Sistema:

### 1. Condiciones controladas:

- Las pruebas se realizan en un entorno donde se puedan simular diferentes intensidades de luz sobre los LDRs. Esto se puede lograr utilizando linternas o variando la exposición del panel en exteriores durante diferentes horas del día.

### 2. Seguimiento de la luz:

- Colocar el panel solar controlado por los servomotores y asegurar que los LDRs estén correctamente conectados. Durante la prueba, los LDRs deben detectar los cambios en la luz y los servos deben ajustarse para orientar el panel en la dirección de mayor luz.

### 3. Optimización de la carga:

- Conectar una Pila o Batería al sistema y verificar el flujo de carga mientras el panel se ajusta para captar la luz de forma óptima. Monitorizar los niveles de carga de la batería para verificar que el sistema esté cargando eficientemente.

### 4. Validación:

- Asegurarse de que, en diferentes condiciones de luz, el sistema reacciona correctamente ajustando la posición del panel y cargando la batería.

## Arquitectura del Sistema

**Descripción Jerárquica:** El sistema está organizado en diferentes módulos que trabajan de manera conjunta para optimizar la captación de energía solar y cargar eficientemente la batería.

### Estos módulos incluyen:

- **Módulo de Detección de Luz (LDRs):** Este módulo consta de 4 LDRs, que detectan la intensidad de la luz solar en diferentes direcciones. El microcontrolador ARDUINO-UNO recibe los valores de los LDRs para determinar la orientación óptima del panel solar.
- **Módulo de Control de Servos:** Basado en las lecturas de los LDRs, este módulo ajusta la posición del panel solar mediante 2 servomotores (servoVertical para el ajuste vertical y servoHorizontal para el ajuste horizontal), garantizando que el panel esté siempre orientado hacia la dirección de mayor luz.
- **Módulo de Gestión de Carga:** Este módulo incluye el regulador de voltaje LM2596, que gestiona la conversión de la energía captada por el panel solar, asegurando que la batería se cargue de manera eficiente y controlada. Este módulo también supervisa el estado de carga de la batería y asegura su protección durante el proceso.
- **Interfaz de Usuario:** Incluye un botón y un LEDs que indican el estado del sistema y permiten al usuario reiniciar o interactuar con el sistema de manera básica.
- **Diagrama de Módulos:** El diagrama adjunto muestra la estructura del sistema, representando las conexiones entre los diferentes componentes:
- **Panel Solar:** Fuente de energía.
- **LDRs (1 a 4):** Módulo de detección de luz que suministra información al Arduino .

- **Servomotores (servoVertical y servoHorizontal):** Módulo de control que ajusta el panel solar.
- **Regulador LM2596:** Encargado de controlar el voltaje suministrado a la batería.
- **ARDUINO-UNO:** El núcleo del sistema, que gestiona los datos de los LDRs, controla los servos y supervisa la carga de la batería.
- **Batería o Pila 18650:** Almacenamiento de la energía captada.

## **Descripción de los Módulos:**

### **Módulo de Detección de Luz:**

Detectar la intensidad de la luz solar desde cuatro direcciones diferentes utilizando cuatro LDRs (LDR\_1, LDR\_2, LDR\_3 y LDR\_4). Los valores de estos sensores se leen a través del ADC (convertidor analógico a digital) del ARDUINO-UNO.

Proceso: Los LDRs detectan la luz solar y envían señales analógicas al ARDUINO-UNO, que las convierte en valores digitales. Estos valores se comparan para determinar hacia qué dirección orientar el panel solar.

### **Módulo de Control de Servo:**

Controlar la orientación del panel solar en función de las lecturas de los LDRs.

Componentes: Incluye dos servomotores, uno para el ajuste vertical (servoVertical) y otro para el ajuste horizontal (servoHorizontal).

Proceso: El ARDUINO-UNO, utilizando las lecturas de los LDRs, envía señales de control a los servos, que ajustan el panel solar en la dirección de mayor luz.

### **Módulo de Gestión de Carga:**

Función: Supervisar y regular el proceso de carga de la batería.

Componentes: Utiliza el regulador LM2596 para estabilizar la salida de voltaje del panel solar y un cargador que administra la energía hacia la batería. También incluye un USB para la monitorización o carga externa.

Proceso: El regulador ajusta la energía captada del panel solar antes de enviarla a la batería para su carga. El ARDUINO-UNO también puede monitorear el estado de carga mediante sensores y LEDs.

### **Dependencias Externas:**

Librerías de control de servos: Para manejar los servos desde el ARDUINO-UNO.

# Diseño del Modelo de Datos

## Datos de Entrada:

### Lecturas de los LDRs:

**Descripción:** Los valores de los cuatro LDRs (LDR\_1, LDR\_2, LDR\_3, LDR\_4) representan la intensidad de luz en diferentes direcciones.

**Tipo de Dato:** Valores analógicos, convertidos a enteros (rango de 0 a 1023) mediante el ADC del ARDUINO-UNO.

**Uso:** Estos valores se utilizan para calcular la orientación óptima del panel solar, comparando las intensidades de luz en distintas posiciones.

## Estado de Carga de la Batería:

**Descripción:** Información sobre el nivel de carga de la batería captada por sensores de voltaje o directamente por el ARDUINO-UNO.

**Tipo de Dato:** Enteros o flotantes que indican el voltaje o porcentaje de carga de la batería.

**Uso:** Se utiliza para supervisar y gestionar el proceso de carga, evitando sobrecargas y protegiendo la batería.

## Datos Internos:

### Orientación del Panel Solar:

**Descripción:** Datos que representan la posición actual del panel solar, tanto en el eje vertical como en el horizontal, controlados por los servomotores.

**Tipo de Dato:** Enteros que indican el ángulo de los servos (por ejemplo, 0° a 180°).

**Uso:** Estos datos son fundamentales para determinar si el panel está correctamente orientado hacia la dirección de mayor luz.

### Niveles de Voltaje de la Batería:

**Descripción:** Información sobre el voltaje actual de la batería, necesaria para determinar la cantidad de energía disponible y el estado de carga.

**Tipo de Dato:** Flotante o entero que indica el voltaje en tiempo real.

**Uso:** Se utiliza para gestionar la carga de la batería, asegurando que se mantenga dentro de un rango seguro.

## **Datos de Salida:**

### **Movimientos del Panel Solar:**

**Descripción:** Comandos de control enviados a los servos para ajustar la posición del panel solar.

**Tipo de Dato:** Enteros que indican los ángulos a los que los servomotores deben moverse (por ejemplo, servoVertical y servoHorizontal reciben datos de posición).

**Uso:** Ajustar el panel solar a la orientación óptima basada en las lecturas de los LDRs.

### **Estado de la Batería:**

**Descripción:** Información visual o digital sobre el estado de carga de la batería, ya sea mediante LEDs, en una pantalla o transmitido a una aplicación móvil.

**Tipo de Dato:** Enteros o flotantes que representan el porcentaje de carga o el voltaje de la batería.

**Uso:** Mostrar al usuario el estado actual de la batería, alertar si está completamente cargada o si necesita atención.

## **Descripción de Procesos y Servicios**

### **Servicios Ofrecidos:**

**Ajuste Automático de la Orientación del Panel Solar:** El sistema mueve el panel solar en tiempo real utilizando servomotores, orientándolo hacia la fuente de luz más intensa, optimizando así la captura de energía solar.

### **Monitoreo y Gestión del Estado de la Batería:**

El sistema supervisa el estado de la batería en todo momento, mostrando al usuario el nivel de carga y gestionando la carga para asegurar eficiencia y seguridad.

### **Descripción del Proceso:**

Lectura de los LDRs y Cálculo de la Orientación Óptima:

**Paso 1:** Los cuatro LDRs ubicados en diferentes direcciones capturan la intensidad de la luz solar.

**Paso 2:** El microcontrolador ARDUINO-UNO procesa las lecturas y compara los valores para determinar cuál de los LDRs recibe más luz.

**Paso 3:** Basado en los valores, el sistema calcula el ángulo ideal para ajustar el panel solar.

**Paso 4:** Se envían señales a los servomotores vertical (servoVertical) y horizontal (servoHorizontal), que mueven el panel solar hacia la dirección óptima.

## **Monitoreo del Estado de la Batería y Regulación de la Carga:**

**Paso 1:** El sistema constantemente lee el nivel de voltaje de la batería mediante el regulador conectado al ARDUINO-UNO.

**Paso 2:** Si el voltaje cae por debajo de un umbral predefinido, el cargador solar comienza a suministrar energía para cargar la batería.

**Paso 3:** Cuando la batería alcanza su capacidad máxima, el sistema regula la entrada de energía para evitar la sobrecarga, preservando la vida útil de la batería.

**Paso 4:** El estado de la batería se refleja en un indicador visual, como LEDs, o es transmitido a una aplicación para monitoreo remoto si el usuario decide integrarlo.

Flujo General del Proceso:

Inicio: Se enciende el sistema y los LDRs empiezan a tomar lecturas de la luz solar.

Orientación del Panel: El Arduino procesa las lecturas y ajusta la orientación del panel solar en función de los valores de luz.

Monitoreo de la Batería: El sistema monitorea el voltaje de la batería, activando el cargador si es necesario.

Optimización Continua: El sistema sigue ajustando el panel y gestionando la carga de la batería en función de las condiciones de luz y energía almacenada.

## **Documentación Técnica - Especificación API**

**Función:** leerLDRs()

**Descripción:** Lee los valores de los sensores de luz (LDR) conectados al Arduino. Los valores son usados para determinar la posición óptima del panel solar.

**Prototipo:**

```
int leerLDRs();
```

**Resultado:**

Retorna un puntero a un array de enteros que contiene los valores leídos de cada LDR.

**Explicación:** Esta función toma la lectura de los 4 sensores LDR que están conectados al Arduino. Estos sensores detectan la intensidad de la luz en diferentes direcciones, lo que permite saber cuál es la posición del sol.



**Función:** `calcularPosicionPanel()`

**Descripción:** Calcula la orientación óptima del panel solar basándose en los valores de luz detectados por los LDR.

**Prototipo:**

```
void calcularPosicionPanel(int valoresLDR[], int &anguloHori, int &anguloVerti);
```

**Argumentos:**

- `valoresLDR[ ]`: Array de enteros con los valores de luz detectados por los sensores.
- `anguloHori`: Referencia al ángulo calculado para el movimiento horizontal del panel.
- `anguloVerti`: Referencia al ángulo calculado para el movimiento vertical del panel.

**Resultado:**

- No retorna valor, pero actualiza los ángulos calculados para orientar el panel.

**Explicación:** Esta función promedia las lecturas de los LDR para determinar si el panel debe moverse hacia arriba, abajo, izquierda o derecha, ajustando los ángulos que se usarán para mover los servomotores.

**Función:** `ajustarPanel()`

**Descripción:** Ajusta la posición del panel solar utilizando los servomotores de acuerdo con los ángulos calculados.

**Prototipo:**

```
void ajustarPanel(int anguloHori, int anguloVert);
```

**Argumentos:**

- `anguloHori`: Ángulo que mueve el servomotor horizontal.
- `anguloVert`: Ángulo que mueve el servomotor vertical.

**Resultado:**

- No retorna valor, pero mueve los servomotores para orientar el panel solar.

**Explicación:** Esta función recibe los ángulos calculados previamente y mueve el panel solar usando los servomotores, asegurando que esté siempre apuntando hacia la dirección óptima de luz.

**Función:** `monitorizarBateria()`

**Descripción:** Supervisa el nivel de carga de la batería y proporciona el voltaje actual.

**Prototipo:**

```
float monitorizarBateria();
```

**Resultado:**

- Retorna un valor flotante con el voltaje actual de la batería.

**Explicación:** Se encarga de verificar cuánta energía tiene la batería, devolviendo el voltaje. Esta información puede utilizarse para saber cuándo la batería necesita ser cargada.

**Función:** `cargarBateria()`

**Descripción:** Inicia el proceso de carga de la batería si el voltaje es menor a un nivel predeterminado.

**Prototipo:**

```
void cargarBateria(float voltajeActual);
```

**Argumentos:**

- **`voltajeActual`:** Flotante que representa el voltaje actual de la batería.

**Resultado:**

- No retorna un valor, pero inicia el proceso de carga cuando la batería lo necesita.

**Explicación:** Esta función revisa si la batería está baja de energía y comienza a cargarla si es necesario, ayudando a mantener el sistema alimentado.

## Diagrama de Flujo para las Funciones

**Función:** `leerLDRs()`

**Inicio -> Leer valores de LDRs -> Almacenar valores -> Retornar valores -> Fin**

**Función: calcularPosicionPanel()**

Inicio -> Recibir valores de LDRs -> Calcular promedio superior e inferior -> Calcular promedio izquierdo y derecho -> Definir ángulos -> Fin

**Función: ajustarPanel()**

Inicio -> Recibir ángulos -> Mover servos -> Ajustar posición del panel -> Fin

**Función: monitorizarBateria()**

Inicio -> Leer voltaje de la batería -> Retornar voltaje -> Fin

**Función: cargarBateria()**

Inicio -> Leer voltaje actual -> Comparar con nivel mínimo -> Si voltaje es bajo, iniciar carga -> Fin

## Código:

```
#include <Servo.h>
```

```
#define PIN_LED 13
```

```
#define LDR1 A1
```

```
#define LDR2 A0
```

```
#define LDR3 A2
```

```
#define LDR4 A3
```

```
// Definimos Servos Horizontal y Vertical

Servo servoHorizontal;

int servoH = 0;

int servohLimiteHigh = 160;

int servohLimiteLow = 20;


Servo servoVertical;

int servoV = 0;

int servovLimiteHigh = 160;

int servovLimiteLow = 20;


unsigned long previousMillisServoH = 0;

unsigned long previousMillisServoV = 0;

const long intervalServo = 100; // Intervalo de 100 ms para los servos


// Variables para el parpadeo del LED

unsigned long previousMillisLED = 0;

const long intervalLED = 500;

bool ledState = LOW;


void setup()

{

    // Configura el LED

    pinMode(PIN_LED, OUTPUT);
```

```
digitalWrite(PIN_LED, LOW);

// Configura los servos

servoHorizontal.attach(10);

servoHorizontal.write(0);

servoVertical.attach(9);

servoVertical.write(0);

}

void loop()

{

    // Lee los valores de los LDRs

    int lecturaLDR1 = analogRead(LDR1);

    int lecturaLDR2 = analogRead(LDR2);

    int lecturaLDR3 = analogRead(LDR3);

    int lecturaLDR4 = analogRead(LDR4);

    // Promediamos los valores de los LDR para control de movimiento

    int promSup = (lecturaLDR1 + lecturaLDR2) / 2;

    int promInf = (lecturaLDR3 + lecturaLDR4) / 2;

    int promIzq = (lecturaLDR1 + lecturaLDR3) / 2;

    int promDer = (lecturaLDR2 + lecturaLDR4) / 2;

    // Temporización para el control vertical del servo

    unsigned long currentMillis = millis();
```

```

if (currentMillis - previousMillisServoV >= intervalServo) {

    previousMillisServoV = currentMillis;

    if (promSup < promInf) {

        servoV = constrain(servoV + 1, servovLimiteLow,
servovLimiteHigh);

        servoVertical.write(servoV);

    } else if (promInf < promSup) {

        servoV = constrain(servoV - 1, servovLimiteLow,
servovLimiteHigh);

        servoVertical.write(servoV);

    }

}

```

```

if (currentMillis - previousMillisServoH >= intervalServo) {

    previousMillisServoH = currentMillis;

    if (promIzq > promDer) {

        servoH = constrain(servoH + 1, servohLimiteLow,
servohLimiteHigh);

        servoHorizontal.write(servoH);

    } else if (promDer > promIzq) {

        servoH = constrain(servoH - 1, servohLimiteLow,
servohLimiteHigh);

        servoHorizontal.write(servoH);

    }

}

```

```

}

// Parpadeo del LED independiente

if (currentMillis - previousMillisLED >= intervalLED) {

    previousMillisLED = currentMillis;

    ledState = !ledState;

    digitalWrite(PIN_LED, ledState);

}

}

```

## Diagramas

Diagrama de LEL (Léxico Extendido del Lenguaje)

### Sinónimo: Panel Solar

- **Noción:** Dispositivo que convierte la energía solar en electricidad. El panel solar se ajusta automáticamente para optimizar la captación de luz solar a través de un sistema de seguimiento solar.
- **Impacto:**
  - Convierte la energía solar en electricidad para cargar dispositivos a través de un puerto USB y cargar pilas recargables.
  - El panel solar se mueve automáticamente para seguir la posición del sol.

### Sinónimo: Seguimiento Solar Automático

- **Noción:** Sistema que permite ajustar la orientación del panel solar en función de la posición del sol, utilizando servomotores para rotar horizontal y verticalmente.
- **Impacto:**
  - El sistema ajusta el panel solar automáticamente para maximizar la captación de luz.
  - Los servos mueven el panel solar horizontalmente (derecha e izquierda) y verticalmente (arriba y abajo) para seguir la trayectoria del sol.
  - La posición inicial de los servos es de 90 grados cuando el sistema se activa.

### Sinónimo: Servo Horizontal

- **Noción:** Componente que controla el movimiento horizontal del panel solar, permitiendo rotar de derecha a izquierda.

- **Impacto:**
  - El servo mueve el panel en dirección horizontal en función de los datos proporcionados por los sensores de luz (LDR).
  - El panel vuelve a 90 grados cuando no está siguiendo la luz.

### **Sinónimo: Servo Vertical**

- **Noción:** Componente que controla el movimiento vertical del panel solar, ajustando la inclinación del panel hacia arriba o hacia abajo.
- **Impacto:**
  - El servo inclina el panel en dirección vertical para maximizar la captación de luz.
  - El panel regresa a 90 grados cuando no está en funcionamiento activo.

### **Sinónimo: LDR (Resistencias Dependientes de Luz)**

- **Noción:** Sensores que detectan la intensidad de la luz solar y proporcionan datos al sistema de seguimiento solar para orientar el panel en la dirección más óptima.
- **Impacto:**
  - Los LDR detectan la luz solar y envían señales al controlador para ajustar la posición del panel.
  - Cuando los LDR detectan suficiente luz, activan el sistema de carga USB y de baterías.

### **Sinónimo: Cargador USB**

- **Noción:** Puerto de salida que permite la carga de dispositivos móviles utilizando la energía generada por el panel solar.
- **Impacto:**
  - Carga dispositivos móviles cuando el sistema está en funcionamiento y recibiendo suficiente luz solar.
  - La carga comienza automáticamente cuando los LDR detectan una captación adecuada de luz.

### **Sinónimo: Cargador de Pilas**

- **Noción:** Componente que permite cargar pilas recargables utilizando la energía generada por el panel solar.
- **Impacto:**
  - Carga las pilas cuando el sistema de seguimiento solar está funcionando correctamente.
  - La carga se detiene automáticamente si no hay suficiente energía solar.

### **Sinónimo: LED de Estado (LED Indicador)**

- **Noción:** Indicador visual que muestra el estado del sistema de seguimiento solar. Funciona con dos modos: parpadeo para funcionamiento correcto y luz fija para indicar fallos.



- **Impacto:**
  - El LED parpadea cuando el sistema funciona correctamente, es decir, cuando los servos y los LDR están operativos y el panel sigue la luz.
  - Si el LED está encendido de manera fija, indica que hay un problema en el sistema, como un fallo en los servos o los sensores.

### **Sinónimo: Botón de Reinicio**

- **Noción:** Botón físico que reinicia el sistema de seguimiento solar, restableciendo la posición de los servomotores y reiniciando los sensores de luz.
- **Impacto:**
  - Cuando se pulsa, el sistema vuelve a su estado inicial, con los servos en la posición de 90 grados.
  - Restablece cualquier error que pueda haber detenido el funcionamiento del sistema.

### **Sinónimo: Controlador de Seguimiento Solar**

- **Noción:** Unidad de control que procesa las señales de los LDR y controla los servos para ajustar la orientación del panel solar.
- **Impacto:**
  - Procesa los datos de los sensores y ajusta los servos de manera continua para seguir la luz solar.
  - Controla la activación del LED de estado y los sistemas de carga USB y de pilas.

Diagrama de Secuencia:

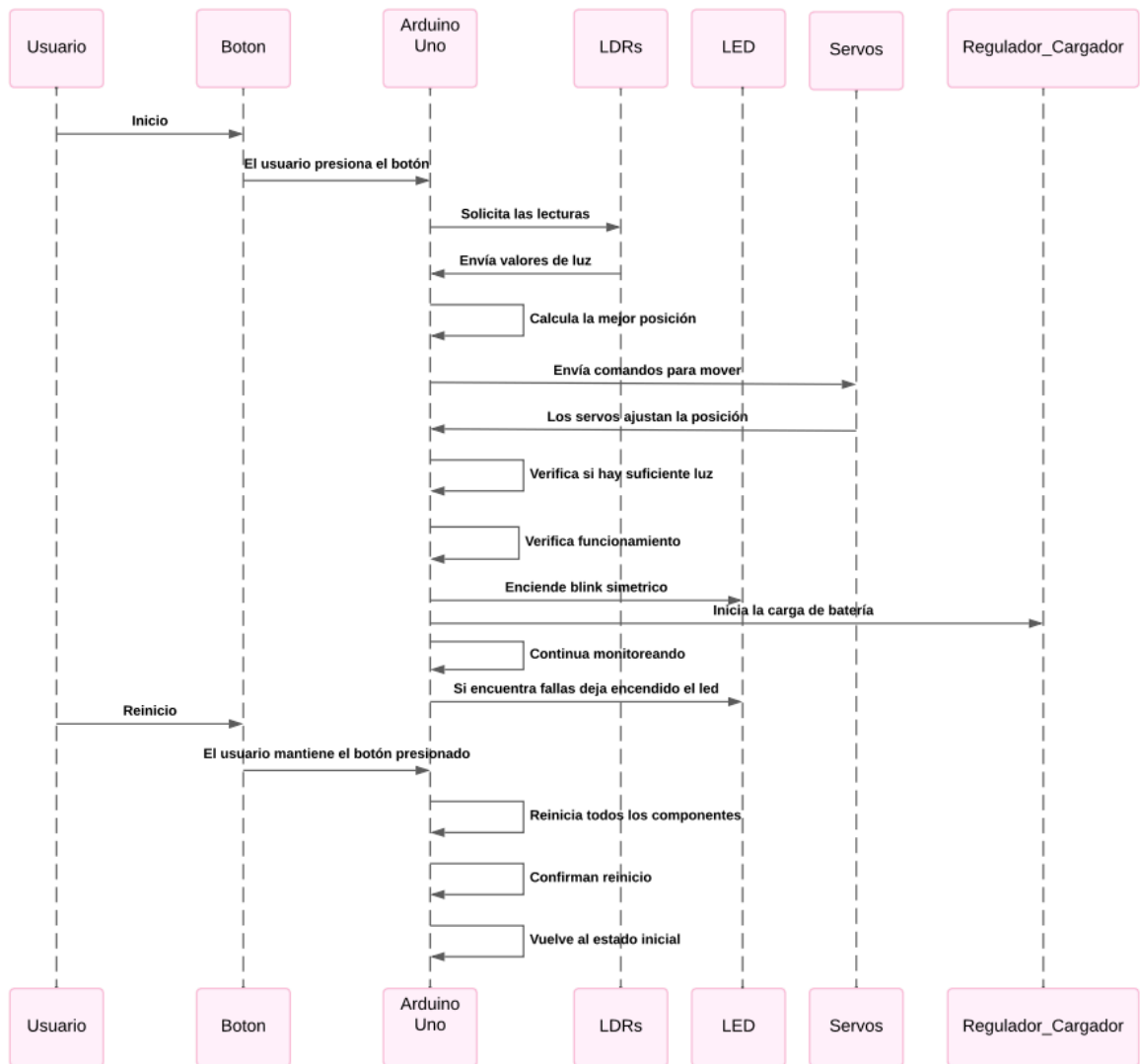


Diagrama Gantt:

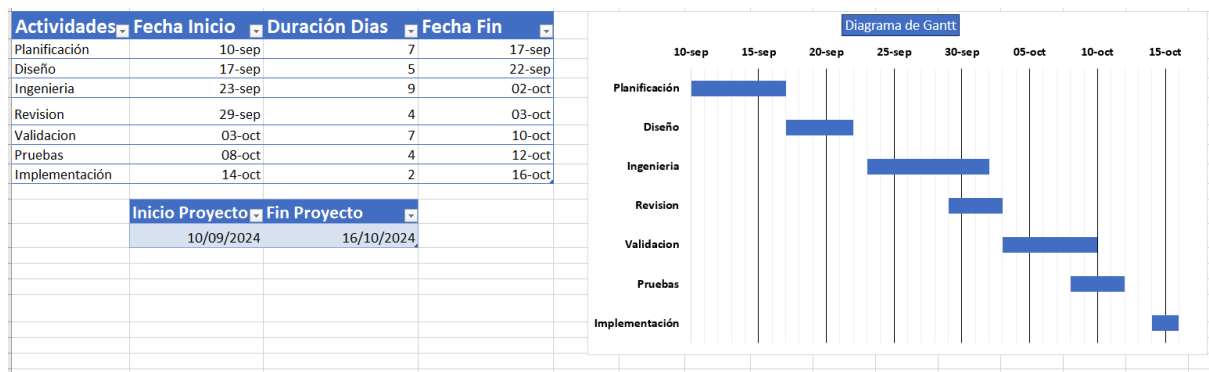


Diagrama de Estados:

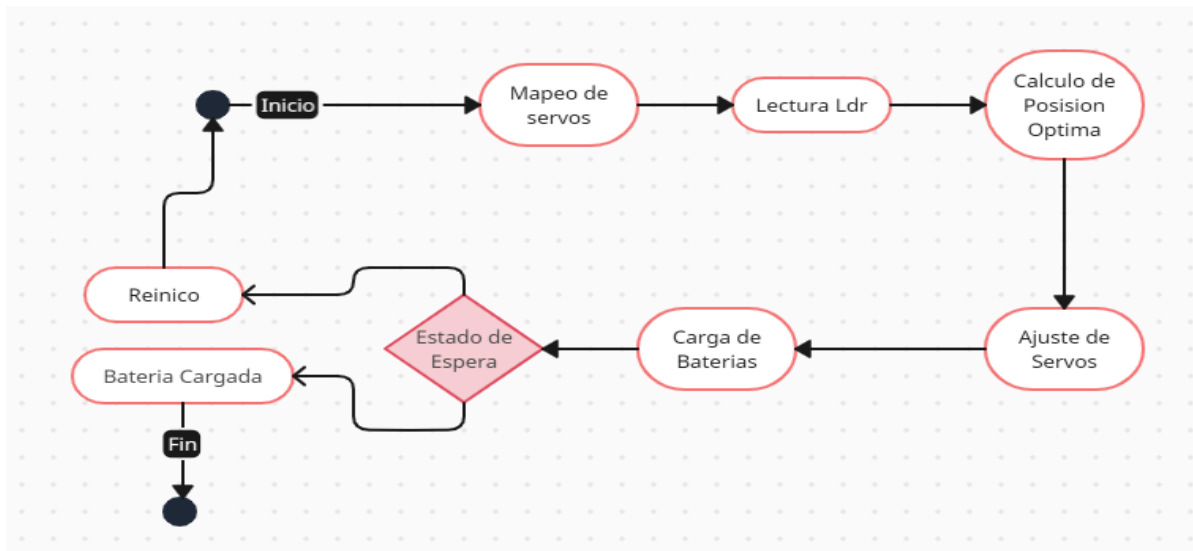
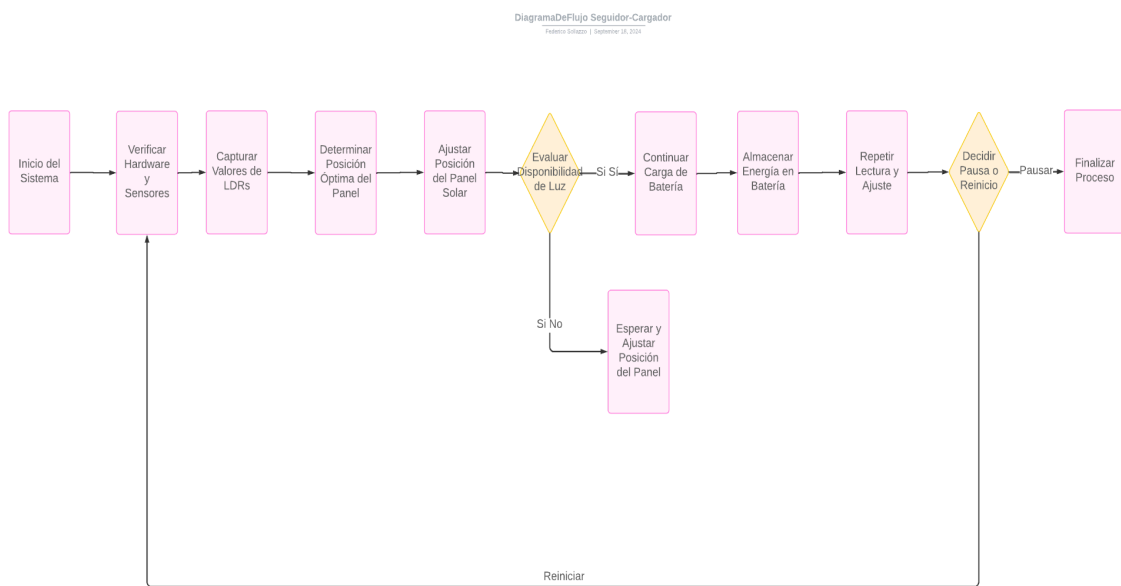


Diagrama de Flujo:



## Conclusiones

Durante el desarrollo del cargador de batería solar con seguidor de luz, se encontraron diversas complicaciones que impulsaron a adoptar soluciones estratégicas. A continuación, se resumen los principales desafíos y las políticas implementadas para resolverlos:

### Complicaciones Encontradas:

**Optimización del Seguimiento Solar:** Uno de los principales retos fue afinar la precisión en la detección de la luz solar mediante los LDRs. Las variaciones en la luz ambiental y las condiciones climáticas complicaron el ajuste preciso del panel solar.

**Gestión Eficiente de la Carga:** Lograr una carga eficiente sin sobrecalentar la batería fue un desafío, especialmente al manejar diferentes tipos de baterías. Esto implicó un ajuste constante de los parámetros de carga basados en las mediciones del voltaje.

**Limitaciones de Componentes:** El uso de servomotores para ajustar el panel solar introdujo limitaciones en la velocidad y la precisión del movimiento, lo que requería ajustes finos en los ángulos de los servos y la forma en que se calculaba la orientación.

**Interferencias en la Lectura de los LDRs:** La sensibilidad de los sensores LDR a la luz ambiental generó interferencias en las lecturas, afectando la precisión del seguimiento del sol.

### **Políticas Adoptadas para Resolverlas:**

**Filtrado de Señal y Promedio de Lecturas:** Para mejorar la precisión en la detección de la luz solar, se implementaron filtros para suavizar las lecturas de los LDRs y se utilizaron algoritmos de promediado que redujeron el ruido en las señales de entrada.

**Control Inteligente de Carga:** Se adoptó un algoritmo de carga eficiente que monitorea constantemente el nivel de voltaje de la batería. Se estableció un umbral seguro para evitar sobrecargas y se agregó un sistema de corte automático al alcanzar el nivel máximo de carga.

**Optimización del Movimiento del Panel:** Se ajustaron los tiempos de respuesta y los ángulos de los servos mediante un algoritmo PID simple, lo que mejoró la precisión en el posicionamiento del panel en función de los valores de los LDRs.

**Pruebas en Distintos Entornos:** Se realizaron pruebas bajo diferentes condiciones de luz y clima para asegurar la funcionalidad del sistema en diversos escenarios. Esto permitió ajustar los parámetros del sistema para mejorar su robustez y rendimiento.

### **Experiencia Adquirida:**

El desarrollo de este proyecto brindó un aprendizaje valioso en varios aspectos:

**Eficiencia Energética:** La experiencia adquirida en la optimización del seguimiento solar y la gestión de la carga permitió una mejor comprensión de los principios de eficiencia energética en sistemas autónomos alimentados por energía solar.

**Integración de Hardware y Software:** Este proyecto involucró la integración eficiente de componentes físicos (LDRs, servos, Arduino , etc.) con la lógica de control en el software, lo que requirió un diseño modular y bien estructurado para facilitar futuras modificaciones y mejoras.

**Manejo de Componentes Sensibles:** El proyecto proporcionó experiencia en la manipulación de sensores y actuadores sensibles, y en cómo mitigar las interferencias externas para obtener datos más precisos.

**Desarrollo de Algoritmos de Control:** Se adquirió experiencia en la implementación de algoritmos de control eficientes para optimizar tanto el movimiento del panel solar como la gestión de la carga de la batería.