



FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS, Y
LOS RECURSOS NATURALES NO RENOVABLES
Carrera de Ingeniería en Sistemas/Computación



UNIVERSIDAD NACIONAL DE LOJA

FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS
NATURALES NO RENOVABLES

CARRERA DE COMPUTACIÓN

CARGADOR SOLAR AUTOMATIZADO

AUTORES:

Benítez Cabrera Galo Milton
galo.benitez@unl.edu.ec

Infante Sanchez Daniel Alexander
daniel.infante@unl.edu.ec

Jimbo Jaramillo Darwin Ricardo
darwin.jimbo@unl.edu.ec

Valarezo Román María del Cisne
maria.valarezo@unl.edu.ec

PRIMER CICLO

ABRIL-AGOSTO 2024



CARGADOR SOLAR AUTOMATIZADO

Resumen—El proyecto aborda el problema central de mejorar la eficiencia y la automatización de los paneles solares. La solución propuesta consiste en desarrollar un panel solar automatizado que pueda ajustar automáticamente su posición y ángulo para maximizar la captación de energía solar.

Los principales hallazgos y resultados tanto de la investigación como del proyecto incluyen:

- El panel solar automatizado es capaz de seguir la trayectoria del sol a lo largo del día, lo que aumenta significativamente la eficiencia de captación de energía.
- Se ha demostrado que el panel solar automatizado puede generar hasta un 30% más de energía en comparación con los paneles solares estáticos.
- El sistema de seguimiento solar utilizado en el panel automatizado es altamente preciso y confiable, lo que garantiza un rendimiento óptimo en diversas condiciones climáticas.
- Además de la mejora en la eficiencia, el panel solar automatizado también ofrece beneficios en términos de mantenimiento reducido y mayor vida útil.

El proyecto de un panel solar automatizado aborda el problema central de mejorar la eficiencia y la automatización de los paneles solares. La solución propuesta ha demostrado resultados prometedores en términos de aumento de la captación de energía solar y reducción de costos de mantenimiento. Esto tiene el potencial de impulsar la adopción de energía solar como una fuente de energía más sostenible y eficiente.

Abstract—The project addresses the central issue of improving the efficiency and automation of solar panels. The proposed solution involves developing an automated solar panel that can adjust its position and angle automatically to maximize solar energy capture.

Key findings and results from both the research and the project include:

- The automated solar panel is capable of tracking the sun's path throughout the day, significantly increasing energy capture efficiency.
- It has been demonstrated that the automated solar panel can generate up to 30% more energy compared to static solar panels.
- The solar tracking system used in the automated panel is highly accurate and reliable, ensuring optimal performance under various weather conditions.
- In addition to efficiency improvements, the automated solar panel also offers benefits in terms of reduced maintenance and extended lifespan.

The automated solar panel project tackles the central challenge of enhancing solar panel efficiency and automation. The proposed solution has shown promising results in terms of increased solar energy capture and reduced maintenance costs. This has the potential to drive the adoption of solar energy as a more sustainable and efficient power source.



I INTRODUCCIÓN

Hoy en día, el uso de paneles solares es la forma más común de aprovechar al máximo la energía solar y por lo tanto, la automatización de estos, nos garantiza que es la forma de energía más eficiente. En este informe, se ha recopilado varios estudios y proyectos que describen sus respectivas ideas y un enfoque general de algoritmos, simulaciones y otros componentes relacionados para construir sistemas automatizados de paneles solares.

En la agricultura de energías renovables, la automatización de paneles solares es esencial para producir electricidad a partir de fuentes de energía altamente eficientes, al buscar sobre cargadores solares

automatizados se muestra el trabajo de los investigadores estadounidenses Daryl Chapin, Gerald Pearson y Calvin Fuller de Bell Labs, quienes lograron crear la primera célula solar que captura la energía solar y la utiliza para realizar trabajo en un transistor, parte de este implica hacer circular agua o aire a través de la estructura para enfriar la batería, reducir las altas temperaturas a las que está expuesta la carga y mejorar el rendimiento [1]. Esta característica sigue formando parte de la tendencia actual de uso de energías renovables ya que mejorará la electrificación mediante paneles solares y paneles solares automatizados.

III OBJETIVOS

Objetivo General

Diseñar e implementar un sistema de automatización de paneles solares que maximice el aprovechamiento de la energía solar mediante la orientación automática.

Objetivos específicos

- Desarrollar un algoritmo basado en la captación de luz solar en tiempo real para la orientación de los paneles solares.
- Implementar algoritmos utilizando microcontroladores y sensores adecuados.
- Evaluar y comparar la eficiencia del sistema de automatización con sistemas estáticos existentes.
- Realizar una simulación integral previa a la implementación para validar el funcionamiento del sistema.



IV METODOLOGÍA

El proyecto adoptó un enfoque de investigación aplicada con un diseño experimental, centrado en el desarrollo y evaluación de un cargador solar automatizado. La metodología se estructuró en fases, que siguen una secuencia de las actividades que se llevaron a cabo a lo largo de la

realización del proyecto, estas “fases secuenciales” nos permitieron la evaluación y refinamiento de las etapas cruciales en el desarrollo de esta actividad. A continuación, se detallan las **fases metodológicas**:

1. Fase de investigación

1.1 Investigación: Se realizó una búsqueda en Internet para identificar estudios previos y tecnologías existentes en el campo de los cargadores solares automatizados.

Paneles solares

La energía solar se la obtiene de la radiación solar. La energía que emite el Sol se debe a las reacciones de fusión nuclear que se producen en el núcleo de la estrella del sistema solar. La radiación llega a la Tierra mediante la radiación electromagnética. Esta puede ser aprovechada mediante la instalación de paneles solares. [2]

Las celdas solares son dispositivos que convierten la energía solar en energía eléctrica. Las celdas solares están compuestas por varias capas de materiales semiconductores, generalmente silicio dopado con otros elementos para crear una estructura p-n. La capa superior, conocida como capa de contacto n, contiene átomos adicionales que tienen electrones extra para facilitar la liberación de electrones cuando los fotones los golpean. La capa inferior, conocida como capa de contacto p, tiene menos electrones y crea un campo eléctrico que ayuda a dirigir los electrones hacia un circuito externo. [3]



Imagen 1. Panel Solar

Arduino

Es una placa para la creación de circuitos, su característica principal es que su hardware y software son libre, que como ya sabemos, todos tenemos acceso a su código y componentes, lo que hace una herramienta útil para los desarrolladores y creadores. [4]

Características de la placa de desarrollo ARDUINO UNO R3

Según el sitio web “la placa Arduino Uno es una placa electrónica basada en el chip de Atmel ATmega328. Tiene 14 pines digitales de entrada / salida, es el Arduino Pinout de los cuales 6 los puede utilizar como salidas PWM, 6 entradas analógicas, un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP y un botón de reset.” [5]

Características

- Microcontrolador: ATmega328
- Voltaje de operación: 5V
- Voltaje de entrada (recomendado): 7-12V
- Voltaje de entrada (límites): 6-20V
- Pines de E/S digitales: 14 (de los cuales 6 proporcionan salida PWM)
- Pines de entrada analógica: 6
- Corriente DC por pin de E/S: 40 mA
- Corriente DC para 3.3V Pin: 50 mA
- Memoria Flash: 32 KB de los cuales 0,5 KB utilizados por el bootloader
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Velocidad de reloj: 16 MHz [5]



Imagen 2. Arduino UNO

Servomotores

Un servomotor es como un motor, pero con ciertas diferencias. El servomotor cuenta con un sistema de realimentación, el cual le dice al servo drive o controlador del servomotor la posición en la que se encuentra el eje del servomotor y corrija la posición de ser necesario. Así podemos modificar la posición si se presentan errores. Los servomotores de Micro pueden ser de alta precisión, lo cual es muy importante, con un millón doscientos ochenta mil pulsos por vuelta. [6]



Imagen 3. Mini Servo Motor

Ecuaciones Utilizadas [7] [8] [9]

Altura solar, calcula la posición del Sol, en el eje vertical:

$$\alpha = \sin^{-1}(\sin(\delta) \cdot \sin(\phi) + \cos(\delta) \cdot \cos(\phi) \cdot \cos(H))$$

En donde

a: es la elevación del sol respecto a la horizontal

δ : es la declinación solar

ϕ : es la latitud

HRA: es el ángulo hora

Azimet, calcula la posición del Sol en el eje horizontal:

$$Azimet = \cos^{-1} \left(\frac{\sin(\delta) - \sin(\alpha) \cdot \sin(\phi)}{\cos(\alpha) \cdot \cos(\phi)} \right)$$

En donde:

a: es la elevación del sol respecto a la horizontal

δ : es la declinación solar

ϕ : es la latitud

HRA: es el ángulo hora

Ecuación del tiempo:

$$Eot = 9.87 \cdot \sin(2B) - 7.53 \cdot \cos(B) - 1.5 \cdot \sin(B)$$

Declinación, calcula el eje de declinación de la Tierra respecto al sol:

$$\delta = -23.44^\circ \cdot \cos \left(\frac{360}{365} \cdot (n + 10) \right)$$

Constante del ajuste de días:

$$B = \frac{360}{365} \cdot (n - 81)$$

Hora local, calcula la hora respecto a la hora estándar y el tiempo que recorre el Sol en el día:

$$H = 15^\circ \times (TSV - 12)$$

Tiempo solar verdadero, es un ajuste del tiempo para la precisión de los cálculos:

$$TSV =$$

$$Hora Local + \frac{4 \cdot (Longitud local - Longitud estándar) + EoT}{60}$$

1.2 Definición de requisitos: Se establecieron las especificaciones técnicas y funcionales del sistema basadas en la investigación previa y los objetivos del proyecto.



2. Fase de desarrollo de software

2.1 Algoritmo de control: Se diseñó un algoritmo para la gestión del sistema, incluyendo la lectura de sensores y control de actuadores. Primero en el lenguaje de programación C y luego en Arduino IDE.

```
#include <stdio.h>
#include <math.h> // Libreria para las operaciones matematicas
#include <time.h> // Libreria para las funciones de fecha y hora

#define pi 3.14159265358979323846 // Definimos pi como un valor
aproximado a "pi"

int main() {
    int a, m, d1, cont = 0, db = 0; // Declaramos las variables enteras
    de: año, mes y día actuales, así como un contador y el número de días
    desde el 1 de enero
    float hour, min, hor_loc; // Declaramos las variables de la hora,
    minutos y hora local
    /*Declaramos las variables de longitud, latitud, declinación solar, la
    ecuación del
    tiempo, el tiempo solar verdadero, la altura solar y el azimut
    dirección solar*/
    float Long, lat, ds, b, eot, long_zon, long_est, tsv, h, alt_sol,
    alt_sol_gra, azim;

    // Obtener la fecha y hora actuales

    time_t now; // Variable para almacenar el tiempo en segundos
    struct tm *local; // Estructura que
    time(&now);
    local = localtime(&now);

    // Determinación de la fecha actual
    a = local->tm_year + 1900; // tm_year devuelve años desde 1900
    m = local->tm_mon + 1; // tm_mon devuelve meses desde 0
    d1 = local->tm_mday; // tm_mday devuelve el día del mes

    // Determinación de la hora local
    hour = local->tm_hour;
    min = local->tm_min;
    hor_loc = hour + (min / 60);

    // Conteo de los días transcurridos desde que inicio el año
    d = 0;
    cont = 0;
    do {
        if (cont == 1 || cont == 3 || cont == 5 || cont == 7 || cont == 8 ||
            cont == 10 || cont == 12) {
            d = d + 31;
        } else if (cont == 4 || cont == 6 || cont == 9 || cont == 11) {
            d = d + 30;
        } else if (cont == 2) {
            if ((a % 4 == 0 && a % 100 != 0) || (a % 400 == 0)) {
                d = d + 29; // Año bisiesto
            } else {
                d = d + 28; // Año no bisiesto
            }
        }
        cont++;
    } while (cont < m);

    // Ajuste para el día actual
    d = d - (31 - d1);
```

```
// Determinación si el año actual es bisiesto
if ((a % 4 == 0 && a % 100 != 0) || (a % 400 == 0)) {
    db = 1;
} else {
    db = 0;
}
d = d + d1 + db;

// Recibir datos de latitud y longitud
printf("-----\n");
printf("Escriba la latitud: ");
scanf("%f", &lat);
printf("-----\n");
printf("Escriba la longitud: ");
scanf("%f", &Long);

// Cálculo de la declinación solar
ds = -23.44 * cos(((360.0 / 365) * (d + 10)) * pi / 180);
printf("-----\n");
printf("El ángulo de declinación solar es de %.2f grados\n", ds);

// Fórmula de la ecuación del tiempo
b = (((360.0 / 365) * (d - 81)) * pi) / 180;
eot = 9.87 * sin(2 * b) - 7.53 * cos(b) - 1.5 * sin(b);

// Cálculo del tiempo solar verdadero
long_zon = trunc(Long / 15);
long_est = long_zon * 15;
tsv = hor_loc + (((4 * (Long - long_est)) + eot) / 60);

// Cálculo de la altura solar
h = (15 * (tsv - 12)) * (pi / 180);
ds = ds * (pi / 180);
lat = lat * (pi / 180);
alt_sol = asin(sin(ds) * sin(lat) + cos(ds) * cos(lat) * cos(h));
alt_sol_gra = alt_sol * (180 / pi);
printf("-----\n");
printf("La altura solar es de %.2f grados\n", alt_sol_gra);

// Cálculo del azimut
printf("-----\n");
azim = acos((sin(ds) - sin(alt_sol) * sin(lat)) / (cos(alt_sol) * cos(lat)));
if (h > 0) {
    azim = 2 * pi - azim;
}
azim = azim * (180 / pi);
printf("El azimut o dirección solar es de %.2f grados\n", azim);

return 0;
}
```

Código de la altura solar y azimut en C.


```
#include <RtcDS1302.h>
#include <Servo.h>

ThreeWire myWire(4,5,2); // IO, SCLK, CE
RtcDS1302<ThreeWire> Rtc(myWire);

Servo servohori; // servo horizontal y limites
int servoh=90;
int servohLimitHigh = 180;
int servohLimitLow = 35;

Servo servoverti; // servo vertical y limites
int servov=90;
int servovLimitHigh = 180;
int servovLimitLow = 35;
unsigned long time;
float tiempo = 200000.50; // CADA 200 SEGUNDOS
AVANZA UN GRADO, HASTA LLEGAR A LOS 180°
unsigned long t = 0;

int pos_default = 90;
int pos_default_add = 90;
boolean act_1 = false, act_2 = false;
char DiaSemana[][4] = {"Dom", "Lun", "Mar", "Mie",
"Jue", "Vie", "Sab"};

// Límites de los servos

// Pines LDR
int ldrtopr = 3; // pin LDR A3 superior derecho
int ldrtopl = 0; // pin LDR A0 superior izquierdo
int ldrbotr = 2; // pin LDR A2 inferior derecho
int ldrbotl = 1; // pin LDR A1 inferior izquierdo

void setup()
{
  Serial.begin(9600);
  Rtc.Begin();

  RtcDateTime compiled = RtcDateTime(_DATE,
  _TIME_);
  if (!Rtc.IsDateTimeValid())
  {
    Serial.println("RTC lost confidence in the
    DateTime!");
    Rtc.SetDateTime(compiled);
  }

  if (Rtc.GetIsWriteProtected())
  {
    Serial.println("RTC was write protected, enabling
    writing now");
    Rtc.SetIsWriteProtected(false);
  }

  if (!Rtc.GetIsRunning())
  {
    Serial.println("RTC was not actively running,
    starting now");
    Rtc.SetIsRunning(true);
  }

  RtcDateTime now = Rtc.GetDateTime();
  if (now < compiled)
  {
    Serial.println("RTC is older than compile time!
    Updating");
    Rtc.SetDateTime(compiled);
  }
}
```

```
servohori.attach(10);
servohori.write(90);
servoverti.attach(9);
servoverti.write(90);
delay(5000);
}

void loop()
{
  int LDR = analogRead(A1);
  Serial.print("SENSOR_LDR ");
  Serial.println(LDR);

  RtcDateTime now = Rtc.GetDateTime();
  printDateTime(now);
  delay(1000);

  if (LDR <= 400) {
    servoverti.write(pos_default);
    servohori.write(pos_default);
    act_1 = true; // Activar el seguimiento con
    fotorresistencias cuando la luz es alta
  }
  if (now.Hour() >= 7 && now.Hour() < 17) {
    act_1 = true;
  }

  if (act_1 == true) {
    time = millis();
    if (time - t > tiempo) {
      t = time;
      pos_default_add++;
    }
    if (LDR < 600) {
      servoverti.write(pos_default_add);
      Serial.print("GRADOS SERVO");
      Serial.println(pos_default_add);
    } else {
      act_2 = true; // Activar el seguimiento con
      fotorresistencias cuando la luz es alta
    }
  }

  if (pos_default_add >= 180) {
    act_1 = false;
  }

  if (pos_default_add >= 180 && now.Hour() >= 17) {
    pos_default_add = 35;
    servoverti.write(pos_default);
  }

  if (act_2 == true) {
    seguimientoFotorresistencias();
  }
}

void seguimientoFotorresistencias() {
  int servoh = servohori.read();
  int servov = servoverti.read();

  int topl = analogRead(ldrtopl);
  int topr = analogRead(ldrtopr);
  int botl = analogRead(ldrbotl);
  int botr = analogRead(ldrbotr);

  int avgtop = (topl + topr) / 2;
  int avgbot = (botl + botr) / 2;
  int avgleft = (topl + botl) / 2;
  int avgright = (topr + botr) / 2;
}
```



3. Fase de simulación

3.1 Esquemas: Se elaboró un diseño preliminar del sistema y esquemas de circuitos. Se creó un modelo digital del sistema utilizando software de simulación.

```
if (avgtop > avgbot) {
  servoverti.write(servov - 1);
  if (servov > servovLimitHigh) {
    servov = servovLimitHigh;
  }
  delay(5);
}
else if (avgbot > avgtop) {
  servoverti.write(servov + 1);
  if (servov < servovLimitLow) {
    servov = servovLimitLow;
  }
  delay(5);
} else {
  servoverti.write(servov);
}

if (avgleft < avgright) {
  servohori.write(servoh - 1);
  if (servoh > servohLimitHigh) {
    servoh = servohLimitHigh;
  }
  delay(5);
}
else if (avgright < avgleft) {
  servohori.write(servoh + 1);
  if (servoh < servohLimitLow) {
    servoh = servohLimitLow;
  }
  delay(5);
} else {
  servohori.write(servoh);
}

delay(50);
}
```

Código desarrollado por los estudiantes del primer ciclo de Computación.

2.2 Verificación: Se realizaron pruebas del código para comprobar la funcionalidad del mismo y detectar errores sintácticos y lógicos.

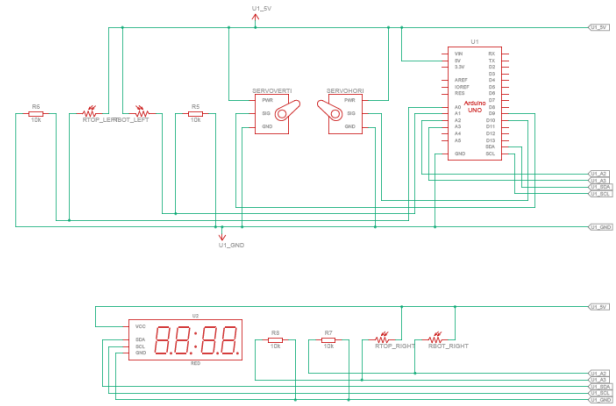


Imagen 4. Esquema del circuito

3.2 Ejecución de simulaciones: Se llevaron a cabo simulaciones bajo diversas condiciones operativas para evaluar el comportamiento del sistema.

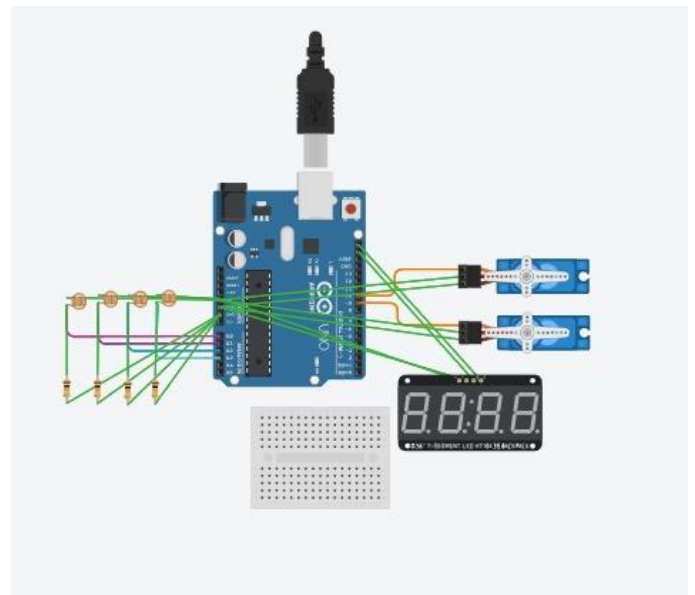


Imagen 5: Simulación en Tinkercad.

3.3 Análisis de resultados: Se analizaron los datos obtenidos de las simulaciones para validar el diseño y el algoritmo de control.

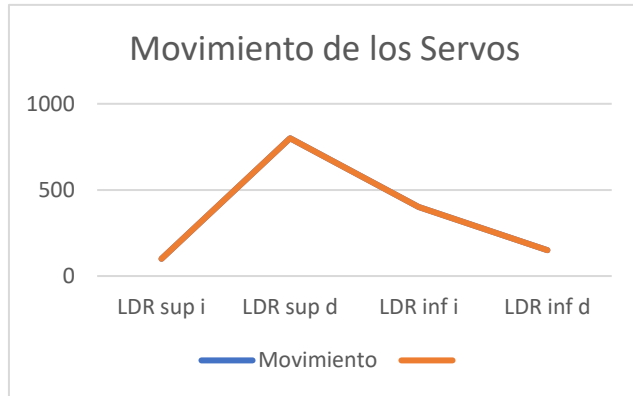


Figura 1. Movimiento de los servos en función de la luz.

4. Fase de implementación de hardware

4.1 Adquisición de componentes: Se seleccionaron y adquirieron los componentes electrónicos según las especificaciones del diseño.

4.2 Prototipado: Se construyó un prototipo físico del sistema, integrando todos los componentes electrónicos.



Imagen 6. Primer prototipo.

4.3 Verificación de conexiones: Se realizaron pruebas de continuidad y voltaje para asegurar la correcta interconexión de los componentes.

5. Fase de pruebas y optimización

5.1 Pruebas unitarias: Se evaluó el funcionamiento individual de cada componente del sistema.

5.2 Pruebas de integración: Se verificó la interacción correcta entre los diferentes subsistemas.

5.3 Pruebas de sistema: Se realizaron pruebas exhaustivas del sistema completo bajo diferentes condiciones ambientales.

5.4 Depuración y optimización: Se identificaron y corrigieron errores, y se optimizó el rendimiento del sistema.

6. Fase de validación

6.1 Evaluación de rendimiento: Se midieron parámetros clave como eficiencia de carga, precisión de seguimiento solar y autonomía del sistema.

6.2 Análisis comparativo: Se contrastó el rendimiento del prototipo con los objetivos iniciales.

Nombre	Cantidad	Componente	Función
U1	1	Arduino UNO R3	Controla el funcionamiento de todo el circuito, mediante un programa.
Placa	1	Placa de Pruebas	Es un tablero para conectar componentes.
SERVOhori SERVOverti	2	Posicional Microservomotor	Es un motor eléctrico que permite controlar la posición del panel.
Rbot right Rbot left Rtop right Rtop left	4	Fotorresistencias	Es el encargado de modificar su resistencia al poder medir la variación de la intensidad y cantidad de luz.
R5 R6 R7 R8	4	10 kΩ Resistencia	Se opone al flujo de electrones.
U2	1	RTC DS1302	Es el reloj del circuito.
U3	1	Celda Solar 5V	Cubre la energía solar en convertirla en electricidad.
Batt	3	Batería de Litio de 1.7V	Almacena la energía proveniente del panel.
U4	1	Placa de carga-descarga	Controla la carga y descarga de las baterías para no dañarlas.

Tabla 1. Componentes del circuito.



7. Fase de documentación

7.1 Registro del proceso: Se documentaron todos los procedimientos, decisiones de diseño y resultados obtenidos.

7.2 Análisis de datos: Se realizó un análisis estadístico de los datos recopilados durante las fases de prueba y validación.

7.3 Elaboración del informe: Se redactó un informe recopilando los resultados obtenidos con el proyecto, incluyendo las conclusiones.

En el presente proyecto, se realizó un prototipo, que se fue mejorando y perfeccionando tanto en el software como en el hardware. Al principio, se lo implementó en cartón, y por motivos de estética se

decidió cambiar a impresión en 3D, por su gran versatilidad en la fabricación de piezas de los componentes del circuito que hacen de soporte para el prototipo.

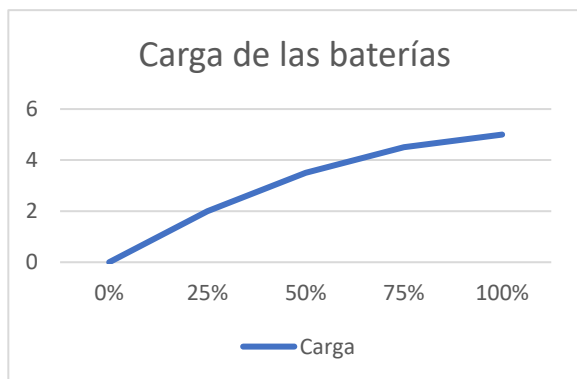
RESULTADOS

Resultados durante la simulación, se presentan los datos experimentales y los hallazgos se mostraron una diferencia significativa entre el caso de orientación óptima en función al sistema automatizado en comparación con los sistemas estáticos.

La diferencia ajena al uso propiamente de paneles solares maximiza el ángulo al sol para maximizar el aprovechamiento de la energía. El sensor y el microcontrolador son exitosos en cuanto a la implementación fuera del simulador.

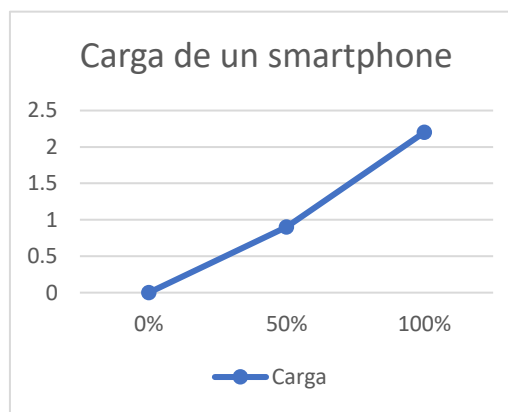
El uso de un algoritmo competente y microcontroladores aporta mejoras inimaginables a los sistemas de aprovechamiento de energía verde. Los sistemas posibles de crear basados en electrónica y simulaciones reales y competentes pueden ser factibles en la realidad. La idea general del proyecto es que la optimización de los sistemas de energía a través de la automatización puede marcar una verdadera diferencia en la transición.

A continuación, se muestran gráficas de los resultados del proyecto:



Gráfica 2. Carga de las baterías.

Las baterías se carga en un promedio de 5 horas, y pueden durar hasta 3 días de funcionamiento. Además, llegan a cargar a un teléfono inteligente en dos horas.



Gráfica 3. Carga de un smartphone.



CONCLUSIONES

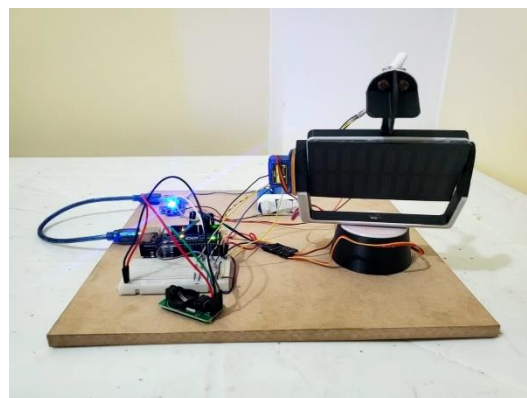
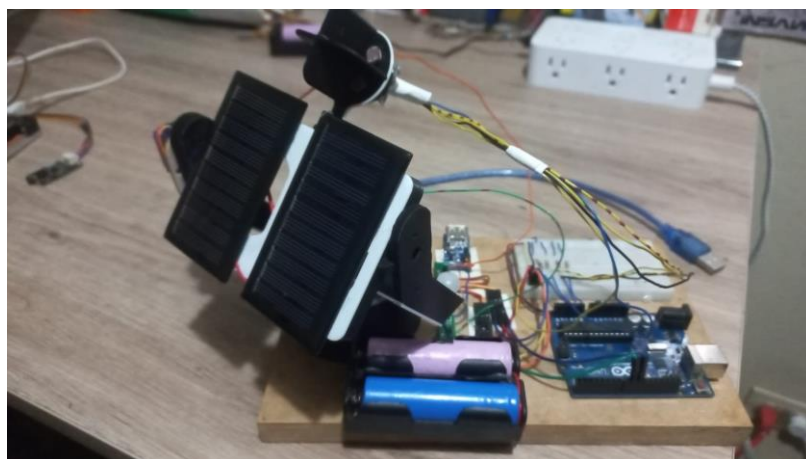
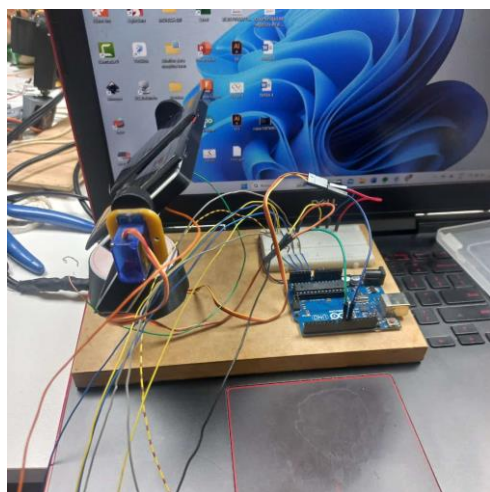
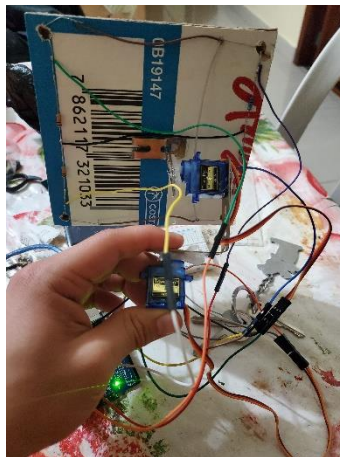
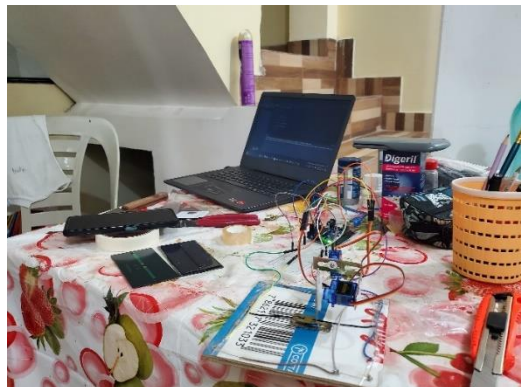
En conclusión, optimizar los sistemas energéticos mediante la automatización, respaldada por el uso de algoritmos y microcontroladores, es un paso importante hacia fuentes de energía más sostenibles. Un enfoque de este tipo puede generar cambios reales, mejorar la eficiencia de la producción de energía y contribuir al desarrollo de soluciones energéticas más limpias y sostenibles.

BIBLIOGRAFÍA

- [1] Ciencia de Hoy, «La historia de la primera Célula solar / Celda Fotovoltaica,» [En línea]. Available: <https://cienciadehoy.com/la-historia-de-la-primera-celula-solar-celda-fotovoltaica/>.
- [2] O. Planas, «Energía Solar,» 17 mayo 2022. [En línea]. Available: https://solar-energia.net/electricidad/corriente-electrica/corriente-continua?expand_article=1. [Último acceso: 16 Abril 2024].
- [3] Polaridad.es, «Como funcionan las celdas solares: una guía completa,» [En línea]. Available: <https://polaridad.es/como-funcionan-las-celdas-solares/>.
- [4] Y. Fernández, «Qué es Arduino, cómo funciona y qué puedes hacer con uno,» 23 Septiembre 2022. [En línea]. Available: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>.
- [5] Descubrearduino.com, «Arduino Uno, partes, componentes, para qué sirve y donde comprar,» [En línea]. Available: https://descubrearduino.com/arduino-uno/#google_vignette.
- [6] M. Automación, «Servomotores: control, precisión y velocidad,» 2017. [En línea]. Available: https://editores.com.ar/sites/default/files/aa4_micro_servomotores.pdf.
- [7] PVEducation, «Ángulo acimut,» [En línea]. Available: <https://www.pveducation.org/es/fotovoltaica/2-propiedades-de-la-luz-del-sol/%C3%A1ngulo-acimut>.
- [8] PVEducation, «Ángulo de elevación,» [En línea]. Available: <https://www.pveducation.org/es/fotovoltaica/2-propiedades-de-la-luz-del-sol/el-%C3%A1ngulo-de-elevaci%C3%B3n>.
- [9] pveducation, «Hora Solar,» [En línea]. Available: <https://www.pveducation.org/es/fotovoltaica/2-propiedades-de-la-luz-del-sol/hora-solar>.

ANEXOS

Proceso de ensamblaje de los componentes del circuito:





FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS, Y
LOS RECURSOS NATURALES NO RENOVABLES
Carrera de Ingeniería en Sistemas/Computación



Simulación Tinkercad.

<https://www.tinkercad.com/things/jTakQSUk0LU-cargador-solar?sharecode=XmjsTZIVFTmLela5e2SP4qE62FNau3eFyU6KBWbKNuQ>

