



ESCUELA
POLITÉCNICA
NACIONAL

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

MÉTODOS NUMÉRICOS

PROYECTO IIB

Nombres:

Correa Anrango Francisco Adrian

Yanez Sandoval Jorge Luis

Yar Saritama Kenneth Jair

PROFESOR: Jonathan Alejandro Zea G.

FECHA DE ENTREGA:

12/02/2025

Índice

1. Objetivos 3

- 1.1. Objetivo General 3
- 1.2. Objetivos Específicos 3

2. Introducción 3

3. Desarrollo: 4

- 3.1. Planteamiento del Problema 4
- 3.2. Demostración Matemática de la Solución 5
 - 3.2.1. Cálculo Básico de los ángulos de control 6
- 3.3. Implementación en Python 6

4. Casos de Prueba 9

- 4.1. Prueba 1 9
 - Duración: 12 10
 - Figura 1: Prueba 1 10
 - Figura 1.2 Grafica de la simulacion 10
- 4.2 Prueba 2 10
- 4.3 Prueba 3 11

Conclusiones 12

5. Anexo 1 13

1. Objetivos

1.1. Objetivo General

Desarrollar un programa en Python que calcule los ángulos de control pitch y roll para un seguidor solar de dos grados de libertad, asegurando que el panel solar se mantenga perpendicular a la luz solar incidente en función de la posición del sol (elevación y azimut). Además, el programa debe permitir la simulación de la trayectoria del sol y del panel solar a lo largo de un día, proporcionando una visualización interactiva que facilite el análisis del movimiento del sistema.

1.2. Objetivos Específicos

- Diseñar un programa de tal forma que sea escalable, que pueda adaptarse a mejoras independientes como la entrada de múltiples paneles solares o la entrada de diferentes condiciones climáticas.
- Crear un modelo matemático que permita calcular con un alto grado de precisión la posición del sol (azimut y elevación) mediante librerías tales como pysolar y astral, teniendo en cuenta consideraciones como la latitud y longitud geográficas, la fecha y la hora de modo local.
- Diseñar una interfaz gráfica interactiva utilizando Tkinter que permita al usuario ingresar parámetros personalizables, como latitud, longitud, fecha y duración de la simulación.
- Implementar una funcionalidad que permita exportar los resultados de la simulación (trayectoria del sol, ángulos de pitch y roll) a un archivo CSV para su posterior análisis.

2. Introducción

Actualmente aprovechar eficientemente la energía solar es crucial para el desarrollo de tecnologías sostenibles. Los sistemas de seguimiento solar ayudan a maximizar la captación de energía al situar los paneles de manera óptima posible respecto a la posición del sol. Los seguidores solares de dos grados de libertad destacan por la capacidad que tienen de ajustar su inclinación en dos ejes, así asegurando una ubicación perpendicular a la luz solar incidente, a la vez incrementando de manera significativa la eficiencia en la generación de

MÉTODOS NUMÉRICOS

El presente proyecto tiene por finalidad desarrollar un programa en Python que obtenga los ángulos de control pitch y roll de un seguidor solar de dos grados de libertad a partir de la posición del sol y que complete la simulación del seguidor solar a lo largo del día de forma interactiva mostrando el movimiento del sol junto con la posición del panel de un seguidor solar de dos grados de libertad asegurando que lo mantiene bien alineado. Para ello, se usarán modelos matemáticos que relacionen la posición del sol con los ángulos de control del sistema, y se usarán herramientas de programación y visualización gráfica para representar y conducir la simulación del seguidor solar de dos grados de libertad de forma dinámica.

3. Desarrollo:

3.1. Planteamiento del Problema

El aprovechamiento eficiente de la energía solar es un desafío clave en el desarrollo de tecnologías sostenibles. Los paneles solares convencionales tienen una eficiencia limitada cuando su orientación no es óptima, debido a que la cantidad de energía captada depende directamente del ángulo de incidencia de la luz solar. Para maximizar la generación de energía, se emplean seguidores solares, los cuales ajustan dinámicamente su inclinación para mantener una alineación óptima con el sol.

En la Facultad de Ingeniería Mecánica, se dispone de un seguidor solar de dos grados de libertad, cuya estructura permite ajustar su inclinación en los ejes pitch (orientación este-oeste) y roll (orientación norte-sur). Sin embargo, para operar de manera eficiente, es necesario calcular con precisión los ángulos de control en función de la posición del sol, la cual varía a lo largo del día y el año.

El desafío principal radica en la necesidad de desarrollar un modelo matemático y una herramienta computacional que permitan determinar los ángulos de control adecuados en tiempo real, asegurando que el panel solar se mantenga perpendicular a la luz solar incidente. Además, es fundamental contar con una simulación interactiva que permita visualizar la trayectoria del sol y la respuesta del seguidor solar, lo que facilitará el análisis y validación de los resultados.

Por lo tanto, este proyecto busca diseñar e implementar un programa en Python que realice estos cálculos, permitiendo ingresar parámetros como la fecha y duración de la simulación, y grafique de manera dinámica la orientación del panel a lo largo del tiempo y generando un video demostrativo de su funcionamiento.

ESCUELA POLITÉCNICA NACIONAL FACULTAD DE
Retos técnicos: INGENIERÍA DE SISTEMAS
Determinación precisa de la posición solar: MÉTODOS NUMÉRICOS

La posición del sol debe ser calculado con alta precisión, tomando en cuenta modelos astronómicos que incluyan latitud, longitud, día y hora local. Para lo que supone trabajar con coordenadas esféricas (azimut y elevación) y convertirlas a coordenadas cartesianas para su representación gráfica.

Determinación de los ángulos de control:

Los ángulos de pitch y roll deben ser calculados dinámicamente a partir de la posición del sol, lo que requiere resolver ecuaciones trigonométricas que relacionan azimut y elevación con los ángulos de inclinación del panel solar.

Simulación interactiva:

La simulación debe ser capaz de mostrar en tiempo real la trayectoria del sol y la orientación del panel solar, actualizando los gráficos 3D y los ángulos de control de forma dinámica. Además, el programa debe permitir al usuario ingresar parámetros personalizados, como la localización geográfica, la fecha y la duración de la simulación.

Generación de resultados visuales:

Debe generarse gráficos 3D interactivos que muestren tanto la trayectoria del sol como la del panel solar, con una conexión clara entre ambas. También se debe permitir exportar los resultados en formatos estándar (imágenes, videos) para facilitar su análisis y presentación.

Validación y optimización:

El sistema debe ser validado mediante la comparación de los resultados obtenidos con datos reales o previamente simulados. Además, debe optimizarse para poder realizar simulaciones de larga duración sin perder el detalle en los cálculos.

3.2. Demostración Matemática de la Solución

Se requiere calcular en tiempo real la posición del sol y determinar los ángulos óptimos de inclinación del panel solar.

Los parámetros clave para este proceso son:

Azimut (α): Dirección horizontal del sol medida desde el norte en el sentido horario.

Elevación (θ): Ángulo del sol sobre el horizonte.

Pitch (β): Ángulo de inclinación del panel sobre el eje horizontal.

Roll (γ): Ángulo de inclinación lateral del panel.

3.2.1. Cálculo Básico de los ángulos de control

Para facilitar la visualización de la trayectoria solar, los valores de azimuth y elevación se convierten a un sistema de coordenadas tridimensional (x, y, z):

$$x = \cos(\theta) \cdot \sin(\alpha)$$

$$y = \cos(\theta) \cdot \cos(\alpha)$$

$$z = \sin(\theta)$$

Donde:

- x, y, z son las coordenadas del sol en un sistema de referencia tridimensional.
- θ y α se expresan en radianes para garantizar la correcta conversión.

ORIENTACIÓN DEL PANEL SOLAR

Cálculo del Ángulo de Roll (γ)

El ángulo de roll (γ) se obtiene a partir de la proyección horizontal del sol:

$$\gamma = \tan^{-1} \left(\frac{\cos(\theta) \cdot \sin(\alpha)}{\sin(\theta)} \right)$$

Este ángulo define la inclinación lateral del panel respecto al suelo.

Cálculo del Ángulo de Pitch (β)

El ángulo de pitch (β) se obtiene considerando la inclinación del sol en la dirección frontal del panel:

$$\beta = \sin^{-1}(-\cos(\theta) \cdot \cos(\alpha))$$

Este ángulo ajusta la orientación del panel para maximizar la perpendicularidad respecto a la luz solar incidente.

3.3. Implementación en Python

**ESCUELA POLITÉCNICA NACIONAL FACULTAD DE
INGENIERÍA DE SISTEMAS
MÉTODOS NUMÉRICOS**

La implementación del sistema se dividió en dos componentes principales: el Back-End y Front-End.

El Back-End es responsable de realizar los cálculos matemáticos y la lógica del programa. Se encarga de determinar la posición del sol, calcular los ángulos de pitch y roll del panel solar, y generar la trayectoria solar en coordenadas cartesianas. Además, incluye funciones para simular dinámicamente tanto el movimiento del sol como la orientación del panel solar en tiempo real.

El Front-End proporciona una interfaz gráfica interactiva que permite al usuario ingresar parámetros, visualizar los resultados de manera dinámica y manipular las simulaciones de forma sencilla y visual.

Cálculo de la Posición Solar

Se desarrollaron funciones para calcular la posición del sol, específicamente los ángulos de azimuth y elevación, utilizando las bibliotecas `pysolar` y `astral`. Estas funciones toman como entrada la latitud, longitud y fecha/hora local, y devuelven los valores correspondientes de azimuth y elevación en grados, los cuales son fundamentales para determinar la orientación del panel solar.

```
az = get_azimuth(latitude, longitude, date)
el = get_altitude(latitude, longitude, date)
return az, el
```

Determinación de Ángulos de Pitch y Roll

Los ángulos de pitch (inclinación vertical) y roll (rotación horizontal) se calculan utilizando principios de trigonometría esférica. La función `calculate_pitch_roll` convierte los ángulos de azimuth y elevación a radianes, y luego aplica fórmulas matemáticas para determinar los ángulos de control del panel solar. Estos cálculos permiten ajustar la orientación del panel para mantenerlo perpendicular a la luz solar incidente, maximizando así la captación de energía solar:

```
def calculate_pitch_roll(azimuth, elevation):
    theta = math.radians(elevation)
    alpha = math.radians(azimuth)
    roll_rad = math.atan2(math.cos(theta) * math.sin(alpha), math.sin(theta))
    pitch_rad = math.asin(-math.cos(theta) * math.cos(alpha))
    return math.degrees(pitch_rad), math.degrees(roll_rad)
```

Conversión a Coordenadas Cartesianas

ESCUELA POLITÉCNICA NACIONAL FACULTAD DE INGENIERÍA DE SISTEMAS

Para graficar la trayectoria del sol en 3D, se implementó una función que convierte las coordenadas esféricas (azimuth y elevation) a coordenadas cartesianas (x, y, z). Este proceso permite representar visualmente la trayectoria del sol en un sistema tridimensional, facilitando la visualización y análisis del movimiento del sol a lo largo del día:

```
def spherical_to_cartesian(azimuths, elevations):
    azimuths_rad = np.radians(azimuths)
    elevations_rad = np.radians(elevations)
    sun_x = np.cos(elevations_rad) * np.sin(azimuths_rad)
    sun_y = np.cos(elevations_rad) * np.cos(azimuths_rad)
    sun_z = np.sin(elevations_rad)
    return sun_x, sun_y, sun_z
```

Rotación del Panel Solar

Para ajustar la orientación del panel solar en el espacio 3D, se implementó una matriz de rotación que aplica los ángulos de pitch y roll previamente calculados. Esta matriz de rotación se define como:

$$R = R_x(\beta) \cdot R_y(\gamma)$$

Donde R_x y R_y son matrices de rotación alrededor de los ejes X e Y, respectivamente. Estas matrices permiten al panel solar ajustar su orientación correctamente en función de los ángulos calculados, asegurando que se mantenga perpendicular a la luz solar incidente:

```
def rotate_panel(pitch, roll):
    pitch_rad = math.radians(pitch)
    roll_rad = math.radians(roll)
    rotation_x = np.array([
        [1, 0, 0],
        [0, math.cos(pitch_rad), -math.sin(pitch_rad)],
        [0, math.sin(pitch_rad), math.cos(pitch_rad)]
    ])
    rotation_y = np.array([
        [math.cos(roll_rad), 0, math.sin(roll_rad)],
        [0, 1, 0],
        [-math.sin(roll_rad), 0, math.cos(roll_rad)]
    ])
    return np.dot(rotation_x, rotation_y)
```

Simulación

Dinámica

Se desarrolló una función que actualiza dinámicamente la posición del sol y la orientación del panel solar en intervalos regulares. Este proceso se logró mediante el uso de bucles y actualizaciones en tiempo real dentro del gráfico 3D, permitiendo observar cómo cambia la

Se implementó una funcionalidad para exportar los resultados de la simulación, incluyendo la trayectoria del sol y los ángulos de pitch y roll, a un archivo CSV. Además, se incluyó la opción de guardar los gráficos generados en formatos estándar como PNG y PDF, facilitando así el análisis y presentación de los resultados.

3.3.2 Front-End

El Front-End fue diseñado para proporcionar una interfaz gráfica intuitiva que permita al usuario interactuar con el sistema y visualizar los resultados de manera clara y sencilla. Para lograr esto, se emplearon las siguientes tecnologías:

Tkinter

Tkinter fue utilizado para crear la interfaz gráfica principal. A través de esta interfaz, el usuario puede ingresar parámetros personalizables, como latitud, longitud, fecha y duración de la simulación. Además, la interfaz muestra las horas de salida y puesta del sol para la ubicación ingresada, proporcionando un acceso rápido a la información clave.

Matplotlib

Se integró Matplotlib con Tkinter para generar gráficos 3D interactivos que muestran tanto la trayectoria del sol como la orientación del panel solar. Estos gráficos se actualizan dinámicamente durante la simulación, permitiendo una visualización en tiempo real del movimiento del sol y la respuesta del panel solar.

Interacción Dinámica

Se implementaron funciones en Python para gestionar la interacción entre el usuario y la aplicación. Los datos ingresados por el usuario son procesados en tiempo real, lo que permite actualizar los resultados y gráficos de manera instantánea a medida que la simulación avanza.

Estilización

Para garantizar una experiencia visualmente atractiva y fácil de usar, se aplicaron técnicas de diseño gráfico simples. Se utilizaron colores claros, etiquetas descriptivas y botones bien organizados, creando una interfaz limpia y accesible.

4. Casos de Prueba

4.1. Prueba 1

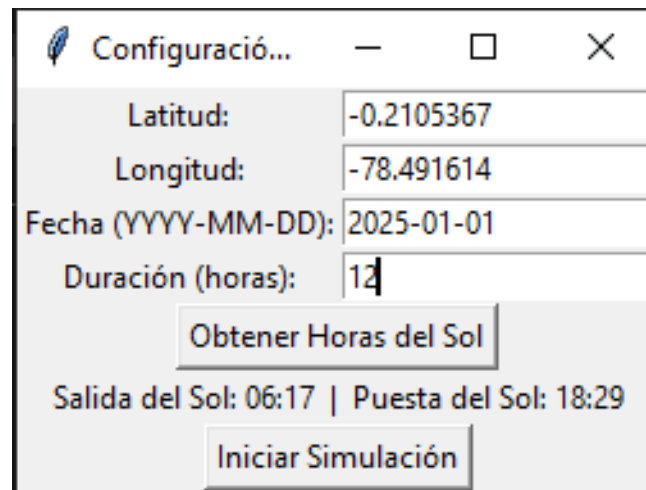
Latitud: -0.2105367

Longitud: -78.491614

**ESCUELA POLITÉCNICA NACIONAL FACULTAD DE
INGENIERÍA DE SISTEMAS
MÉTODOS NUMÉRICOS**

Fecha: 2025-01-01

Duración: 12



Configuració...

Latitud: -0.2105367

Longitud: -78.491614

Fecha (YYYY-MM-DD): 2025-01-01

Duración (horas): 12

Obtener Horas del Sol

Salida del Sol: 06:17 | Puesta del Sol: 18:29

Iniciar Simulación

Figura 1: Prueba 1

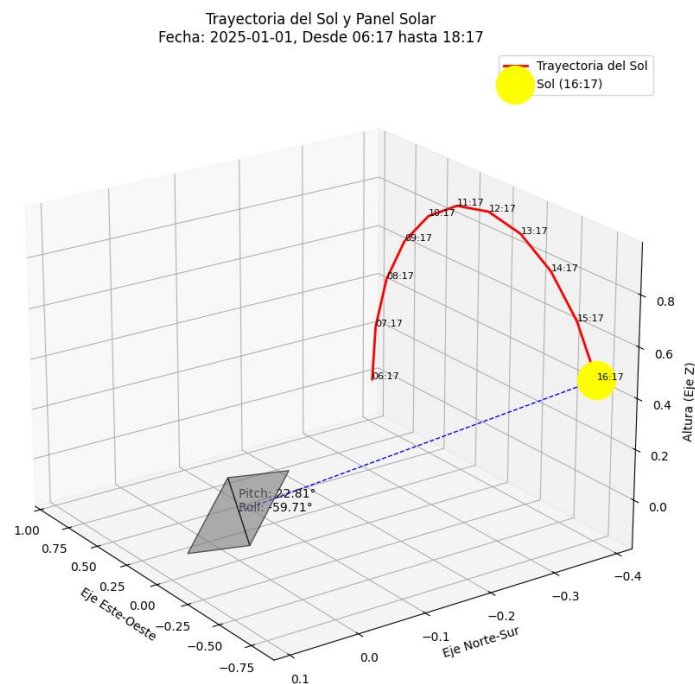


Figura 1.2 Grafica de la simulacion

4.2 Prueba 2

ESCUELA POLITÉCNICA NACIONAL FACULTAD DE
 INGENIERÍA DE SISTEMAS
 MÉTODOS NUMÉRICOS

Latitud: -0.2105367
 Longitud: -78.491614

Fecha: 2025-01-29

Duración: 6

Configuració...

Latitud: -0.2105367
 Longitud: -78.491614
 Fecha (YYYY-MM-DD): 2025-01-29
 Duración (horas): 6

Obtener Horas del Sol

Salida del Sol: 06:26 | Puesta del Sol: 18:38

Iniciar Simulación

Figura 2: Prueba 2

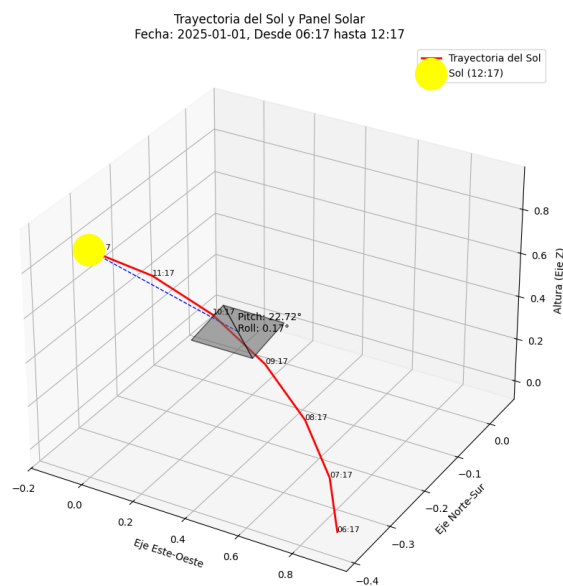


Figura 2.2 Grafica de la simulación

4.3 Prueba 3

Latitud: -0.2105367

Longitud: -78.491614

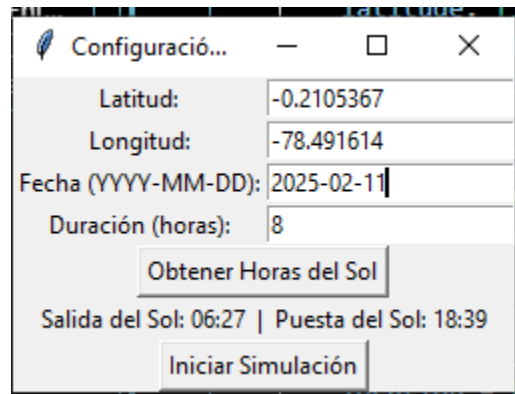


Figura 3: Prueba 3

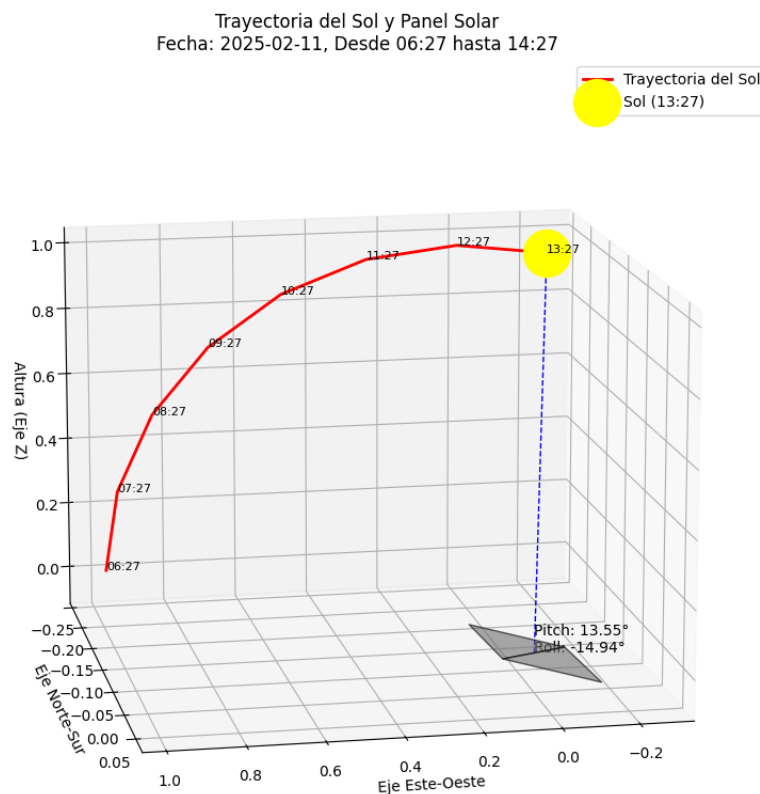


Figura 3.2 Grafica de la simulación

Conclusiones

- El prototipo desarrollado ha permitido crear una herramienta computacional en Python capaz de simular la trayectoria del sol y calcular los ángulos de pitch y roll para orientar un panel solar de dos grados de libertad. La implementación combina modelos matemáticos precisos con una interfaz gráfica interactiva para representar el

- Al calcular los ángulos de **pitch** y **roll** en función de la posición del sol, se asegura que el panel solar se mantenga perpendicular a la luz solar incidente, lo que maximiza la eficiencia en la generación de energía. Este enfoque es especialmente útil en sistemas de seguimiento solar, donde la orientación precisa del panel es crucial para optimizar la captación de energía.
- La integración de Tkinter y Matplotlib permitió desarrollar una interfaz gráfica interactiva que no solo visualiza la trayectoria del sol y la orientación del panel solar en tiempo real, sino que también ofrece la flexibilidad de personalizar parámetros como la localización geográfica, la fecha y la duración de la simulación.
- El desarrollo de este prototipo ha proporcionado valiosa experiencia práctica en programación, visualización de datos y diseño de interfaces gráficas, habilidades esenciales para abordar los desafíos tecnológicos en áreas como la ingeniería, las ciencias computacionales y las energías renovables.

Potencial para Futuras Mejoras

- Para mejorar la simulación, se podría incorporar el impacto de condiciones climáticas variables, como nubosidad o sombras. Estos factores afectan la cantidad de luz solar disponible y, por lo tanto, la eficiencia del sistema. Considerar estos efectos permitiría una evaluación más realista del rendimiento del panel solar en condiciones atmosféricas cambiantes.
- Una posible mejora sería extender la simulación para incluir múltiples paneles solares, lo que permitiría analizar el rendimiento colectivo del sistema en lugar de uno solo. Esto podría ser útil para evaluar cómo varios paneles interactúan entre sí en un sistema de energía solar más grande, optimizando la distribución de energía.
- El prototipo podría mejorarse al incorporar la exportación de datos en formatos estándar como CSV o JSON, facilitando la integración con otras aplicaciones o herramientas de análisis de datos. Además, la capacidad de generar videos demostrativos del funcionamiento del sistema podría ser útil tanto para presentaciones como para la documentación del proyecto.

5. Anexo 1

El flujograma muestra, de manera resumida, la secuencia principal de etapas del programa:

ESCUELA POLITÉCNICA NACIONAL FACULTAD DE INGENIERÍA DE SISTEMAS
desde la captura de datos (latitud, longitud, fecha), pasando por los cálculos de la posición solar y la rotación del panel, hasta la representación gráfica 3D de la trayectoria del sol y la inclinación del panel. Este diagrama sirve para visualizar la lógica de ejecución y la interacción entre las funciones principales del proyecto.

