**UNIVERSIDAD NACIONAL DE INGENIERÍA**

**FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA**

**PROGRAMACIÓN ORIENTADA A OBJETOS (POO)**



**Informe sobre Smart Home y su Implementación en Programación Orientada a Objetos**

|  |  |  |
| --- | --- | --- |
| **N°** | **CÓDIGO** | **APELLIDOS Y NOMBRES** |
| 1 | :20212639J | :BELLIDO DOMINGUEZ, Ricardo James |
| 2 | :20221447B | :CHÁVEZ TUÑOQUE, Edgar Raúl |
| 3 | :20221218C | :CHIRA ARAKAKI, Yoshiro Eduardo |
| 4 | :20221218C | :FERNANDEZ HERVACIO, Lennin Cristofher |

**CÓDIGO DE CURSO-SECCIÓN:** BMA15-M

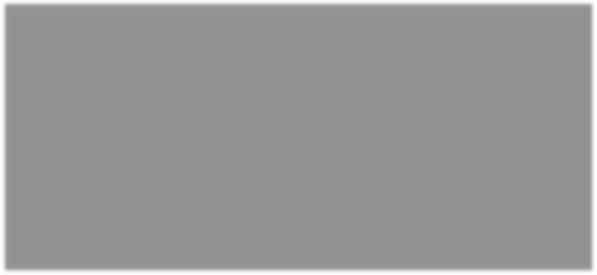
**DOCENTE:** LOAYZA MEJÍA, Washington Ernesto

FECHA DE ENTREGA:

04-03-2025

RÍMAC-LIMA-PERÚ

2024-3



*Los autores autorizan a la UNI a reproducir el siguiente informe realizado, con fines estrictamente educativos.*

*Atte.: Los autores*

**©2024, Universidad Nacional de Ingeniería. Todos los derechos reservados**

**DEDICATORIA**

*Dedicado a quienes creen en la ciencia y en la exploración sin límites. A nuestro mentor y compañeros de trabajo, cuya colaboración y dedicación han sido fundamentales en cada paso de este proyecto. Que este informe refleje y proyecte nuestro compromiso y pasión por el conocimiento*

**AGRADECIMIENTOS**

*Nos gustaría expresar nuestros más sinceros agradecimientos a Loaysa Mejía, Washington Ernesto por su guía experta, apoyo constante y valiosos comentarios a lo largo de esta etapa, a nuestras familias por su constante aliento durante esta etapa, a todos los integrantes por sus valiosas contribuciones que han enriquecido enormemente este trabajo, así como también a nuestra institución la Universidad Nacional de Ingeniería por brindarnos un espacio para poder aprender y mejorar en nuestro futuro profesional*

ÍNDICE

[I. INTRODUCCIÓN 1](#_Toc191944364)

[II. OBJETIVOS 1](#_Toc191944365)

[III. MATERIALES: 2](#_Toc191944366)

[IV. OTROS 4](#_Toc191944367)

[4.1. Herramientas empleadas para planificar nuestro proyecto: 4](#_Toc191944368)

[4.1.1. Work Breakdown Structure (WBS): 4](#_Toc191944369)

[4.1.2. Diagrama de Gantt: 6](#_Toc191944370)

[4.2. Simulaciones y modelado para el hardware 8](#_Toc191944371)

[4.2.1. Simulador de encendido y apagado automático de una estufa y aire acondicionado: 8](#_Toc191944372)

[4.2.2. Simulado del encendido y apagado automático de un foco: 9](#_Toc191944373)

[4.2.3. Simulador de una puerta o el accionar de una persiana automática: 10](#_Toc191944374)

[4.2.4. Simulador de timbre automático: 12](#_Toc191944375)

[4.3. Componentes de un Smart Home 14](#_Toc191944376)

[4.4. Aplicación de POO en un Sistema Smart Home 14](#_Toc191944377)

[4.4.1. Clases y Objetos: 14](#_Toc191944378)

[4.4.2. Herencia: 14](#_Toc191944379)

[4.4.3. Encapsulamiento: 15](#_Toc191944380)

[4.4.4. Polimorfismo: 15](#_Toc191944381)

[4.5. Análisis del Diagrama 17](#_Toc191944382)

[4.5.1. Clase Cuenta 17](#_Toc191944383)

[4.5.2. Clase Smarthome 17](#_Toc191944384)

[4.5.3. Clase Sensor 18](#_Toc191944385)

[4.5.4. Clase Iluminacion 18](#_Toc191944386)

[4.5.5. Clase Ventilación 19](#_Toc191944387)

[4.5.6. Clase Cerradura 20](#_Toc191944388)

[4.6. Relaciones entre Clases 20](#_Toc191944389)

[4.6.1. Clase Cuenta: 20](#_Toc191944390)

[4.6.2. Clase Smarthome: 20](#_Toc191944391)

[4.6.3. Clases Sensor, Iluminacion, Ventilación, Cerradura: 20](#_Toc191944392)

[4.7. Riesgos y Desventajas 21](#_Toc191944393)

# INTRODUCCIÓN

En la actualidad, la tecnología ha transformado la manera en que interactuamos con nuestro entorno, dando lugar a los Smart Homes o casas inteligentes. Un Smart Home es un sistema automatizado que permite controlar dispositivos como luces, ventiladores, sensores y electrodomésticos de manera remota o programada, mejorando la comodidad, eficiencia y seguridad del hogar. Este informe tiene como objetivo explorar el concepto de Smart Home, sus componentes, ventajas, desventajas, y cómo se puede modelar un sistema de casa inteligente utilizando los principios de la Programación Orientada a Objetos (POO) en Python.

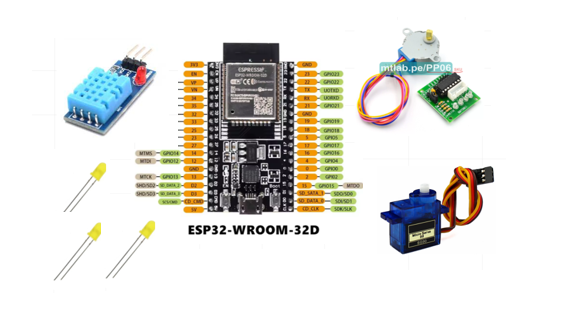
**¿Qué es una Smart Home?:**

Un Smart Home es un hogar que utiliza dispositivos conectados a internet para permitir el control remoto y la automatización de diversos sistemas y electrodomésticos. Estos sistemas pueden incluir iluminación, climatización, seguridad, entretenimiento y más. La aplicación de nuevas tecnologías en el entorno doméstico permite la comunicación entre todos los equipos eléctricos e informáticos con el usuario, optimizando el tiempo y mejorando la calidad de vida.

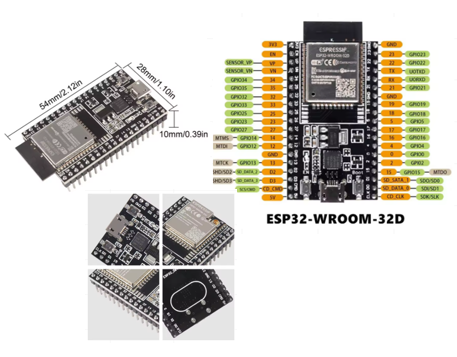
# OBJETIVOS

* Facilitar las actividades domésticas
* Comprender el concepto de Smart Home
* Aplicar POO en la modelación de un Smart Home
* Identificar conceptos de POO

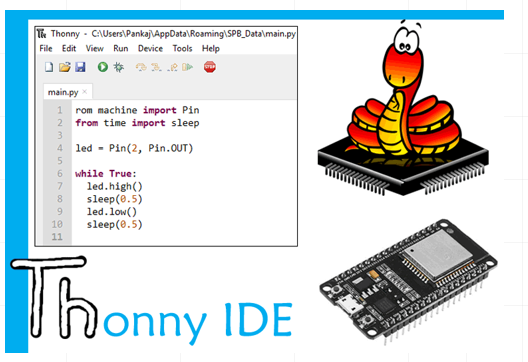
# MATERIALES:



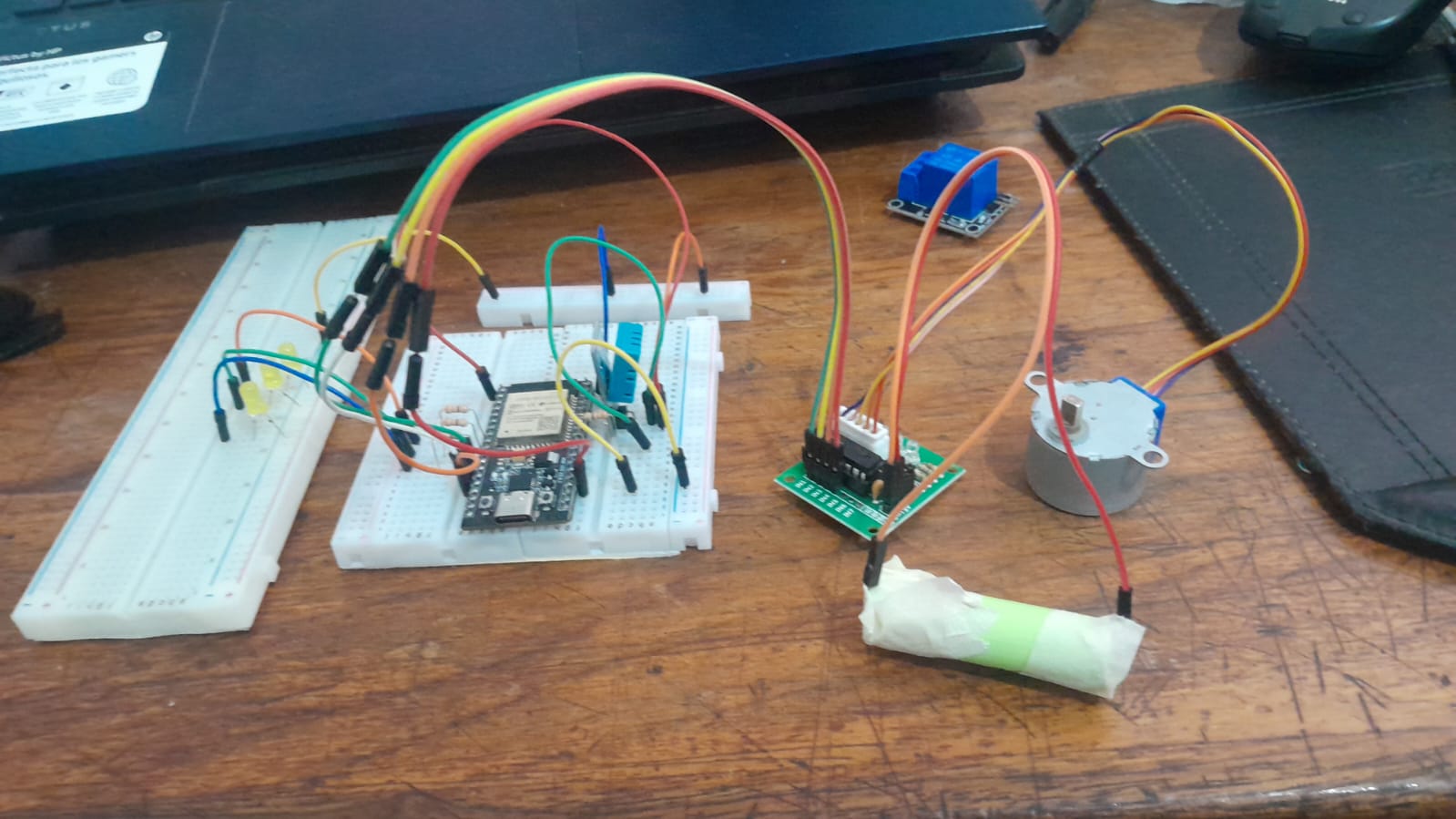
**IMAGEN 1**



**IMAGEN 2**



**IMAGEN 3**



**FOTOGRAFÍA 1**

# OTROS

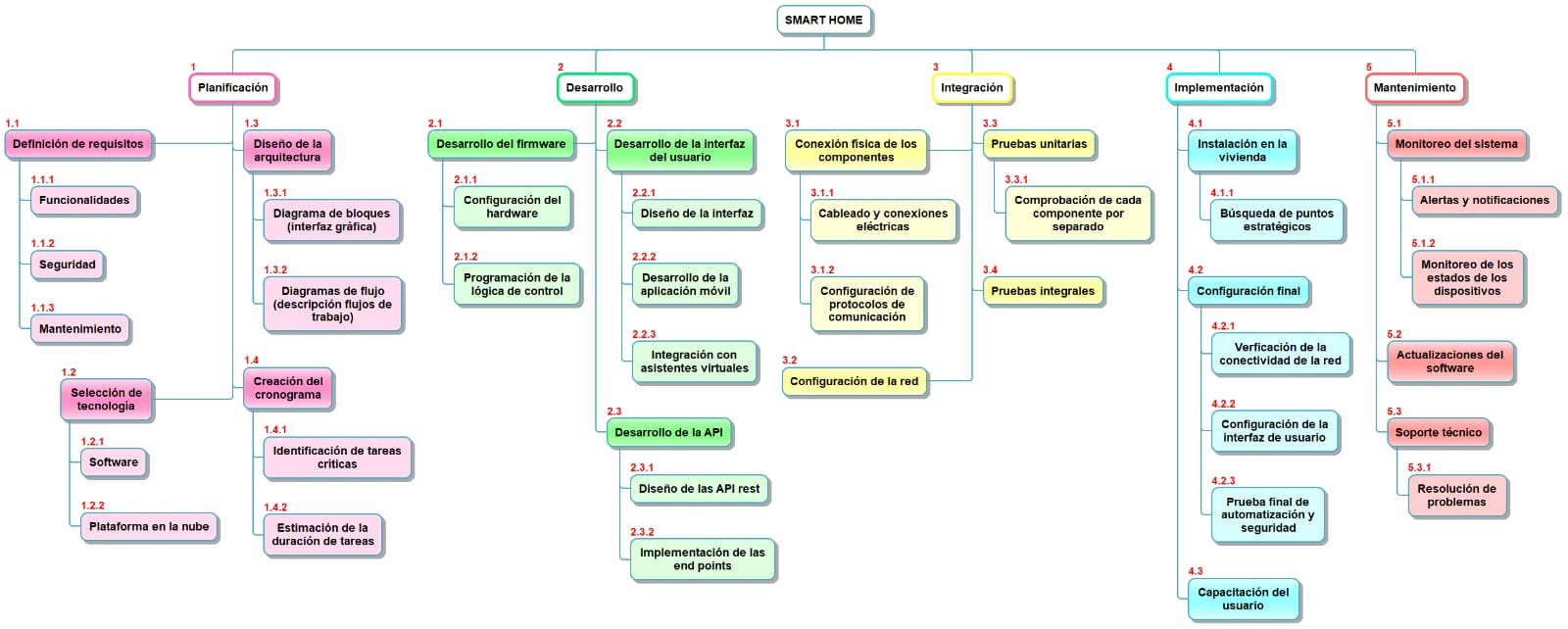
## Herramientas empleadas para planificar nuestro proyecto:

### Work Breakdown Structure (WBS):

**También llamada Estructura de Desglose del Trabajo**, es una herramienta de gestión de proyectos que descompone un proyecto grande en tareas más pequeñas y manejables. Su objetivo principal es organizar y definir el alcance del proyecto para facilitar la planificación, el control y la asignación de responsabilidades.

**DETALLES:**

* **Claridad en el alcance del proyecto:** El WBS permite desglosar el proyecto en elementos más pequeños, lo que ayuda a definir claramente lo que se debe hacer. Esto asegura que todas las tareas necesarias para completar el proyecto sean identificadas.
* **Mejora la planificación y el control:** La estructura facilita la creación de un cronograma detallado, lo que a su vez facilita el seguimiento del progreso del proyecto. Además, permite identificar posibles cuellos de botella o problemas antes de que ocurran.
* **Requiere experiencia:** Para crear un WBS efectivo, se necesita un conocimiento profundo del proyecto. Un WBS mal diseñado o incompleto puede resultar en una planificación deficiente y en dificultades en la ejecución



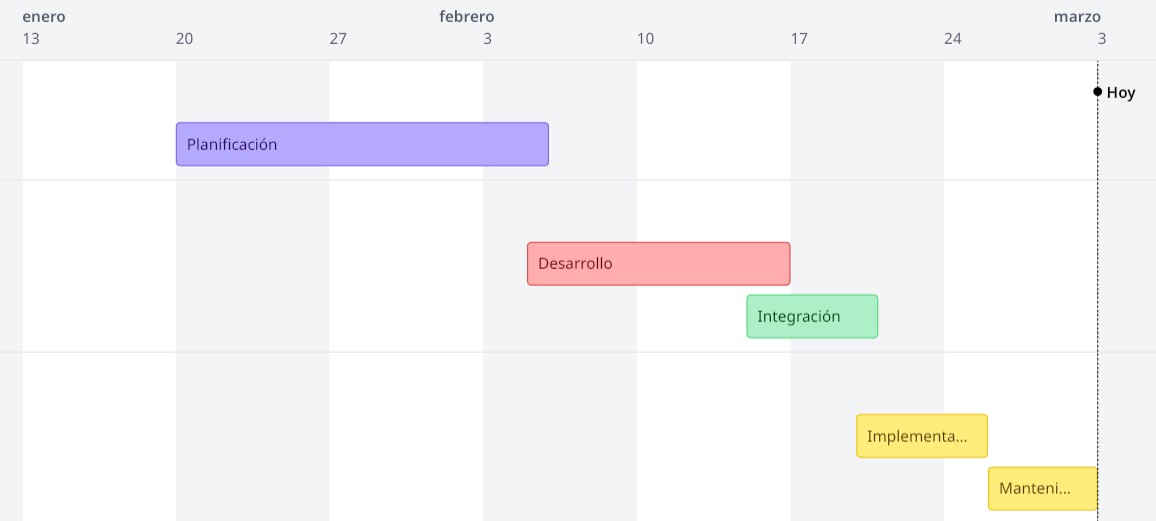
**IMAGEN 5: NUESTRO WBS**

### Diagrama de Gantt:

El **Diagrama de Gantt** es una herramienta visual de gestión de proyectos que se utiliza para representar de manera gráfica las actividades y tareas de un proyecto a lo largo del tiempo. Cada tarea se presenta como una barra horizontal, cuyo inicio y final indican la duración de la tarea dentro de un calendario.

**DETALLES:**

* **Monitoreo del avance:** Permite hacer un seguimiento sencillo del avance del proyecto. Al actualizar las barras conforme se completan las tareas, se puede visualizar de manera inmediata si el proyecto avanza según lo planeado o si hay retrasos.
* **Identificación de dependencias:** Muestra claramente las dependencias entre las tareas, es decir, qué tareas no pueden iniciarse hasta que otras hayan terminado, lo que ayuda a evitar cuellos de botella y a gestionar mejor los recursos.
* **No muestra detalles cualitativos:** El diagrama de Gantt se enfoca únicamente en la temporalidad de las tareas y no ofrece información detallada sobre el progreso de las mismas, el esfuerzo requerido, o los recursos específicos asignados. Esto significa que no proporciona una visión integral del proyecto, solo de su cronograma.



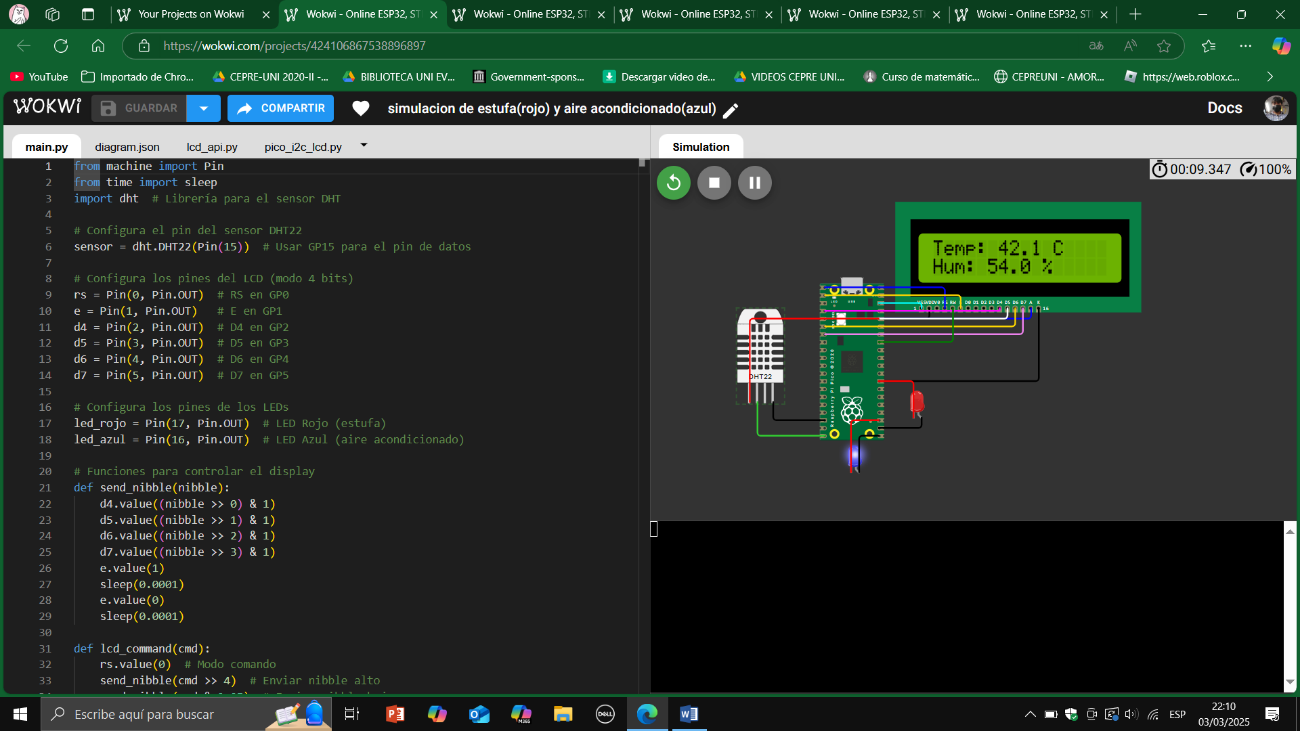
**IMAGEN 5: NUESTRO DIAGRAMA GANTT**

## Simulaciones y modelado para el hardware

Estos modelos eran los que en un principio se querían hacer sin hacer uso del hardware, sin embargo; al final decidimos hacer uso del microcontrolador ESP-32, es por ello que en esta parte solo serán simulaciones comparado al producto final que se vio en la parte de materiales

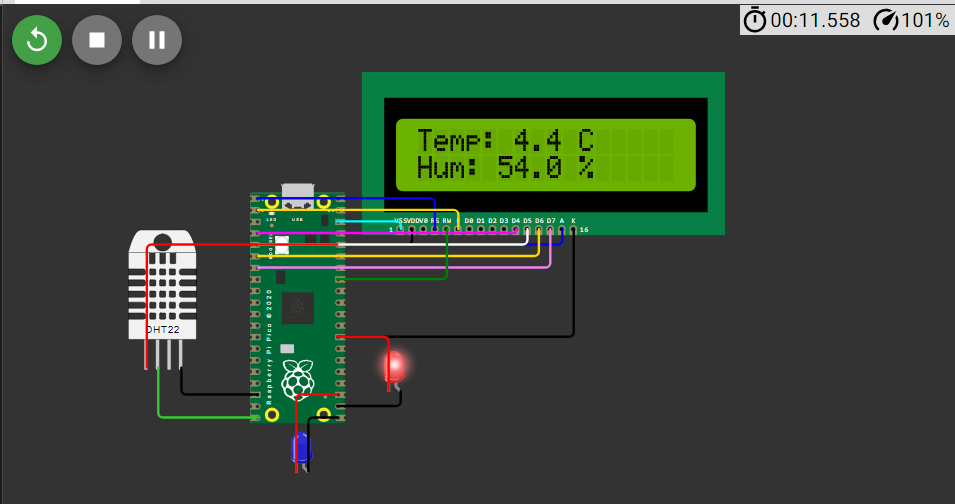
### Simulador de encendido y apagado automático de una estufa y aire acondicionado:

Ambas simulaciones se basan en un sensor de temperatura y humedad los cuales se van a mostrar en un display para el encendido automático de la estufa y el aire acondicionado se optó por poner una temperatura menor a 10°C para el caso de la estufa y un valor mayor a 20°C para el aire acondicionado además se representaron con leds de color rojo y azul respectivamente

En el ejemplo siguiente se muestra cuando se enciende el aire acondicionado (led azul):

**IMAGEN 6**

A continuación, se mostrará el encendido de la estufa (led rojo):

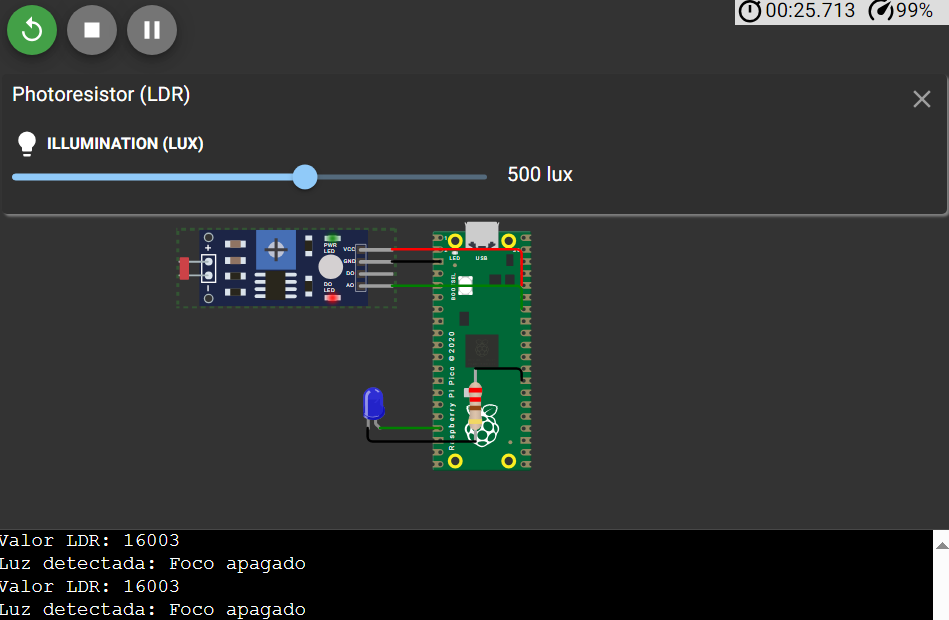


**IMAGEN 7**

### Simulado del encendido y apagado automático de un foco:

El foco está representado por un led el cual se apagará o encenderá de acuerdo a los niveles de luz que reciba la fotorresistencia

A continuación, se mostrará cómo funciona:



**IMAGEN 8**

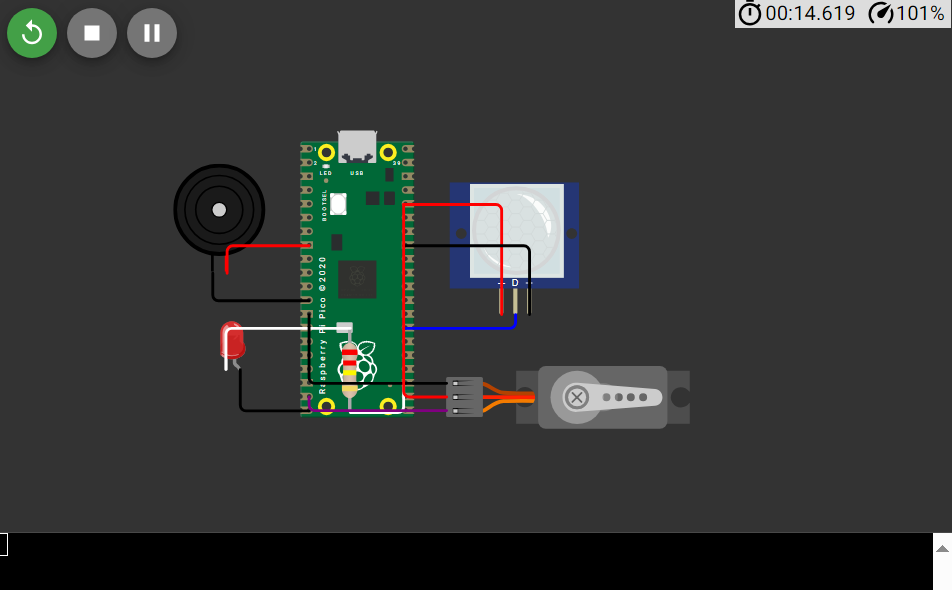
Como se puede apreciar en el ejemplo anterior cuando hay un nivel de luz aceptable el led(foco) no enciende, pero ahora que pasaría si bajamos la luminosidad

**IMAGEN 9**

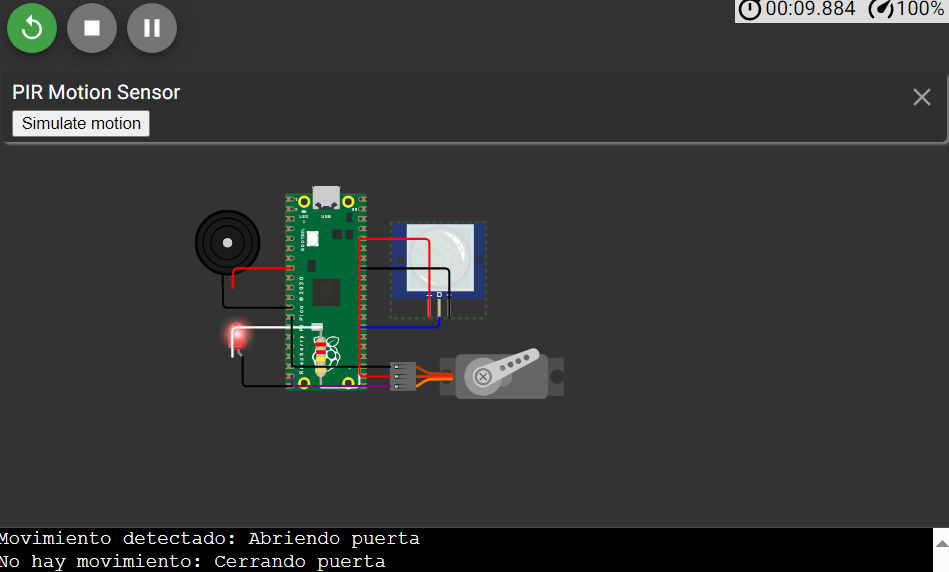
### Simulador de una puerta o el accionar de una persiana automática:

Para hacer esta simulación se optó por imaginar el servomotor que se mostrará después funciona como una puerta o el accionador de una persiana automática que se despliega dependiendo del comando de movimiento que se le dé adicionalmente agregue un buzzer para que imaginar que se le está dando sonido a la puerta.

Se mostrará cómo funciona el mecanismo:

Cuando no se da el comando de movimiento necesario para que se active el sensor esto no responderá

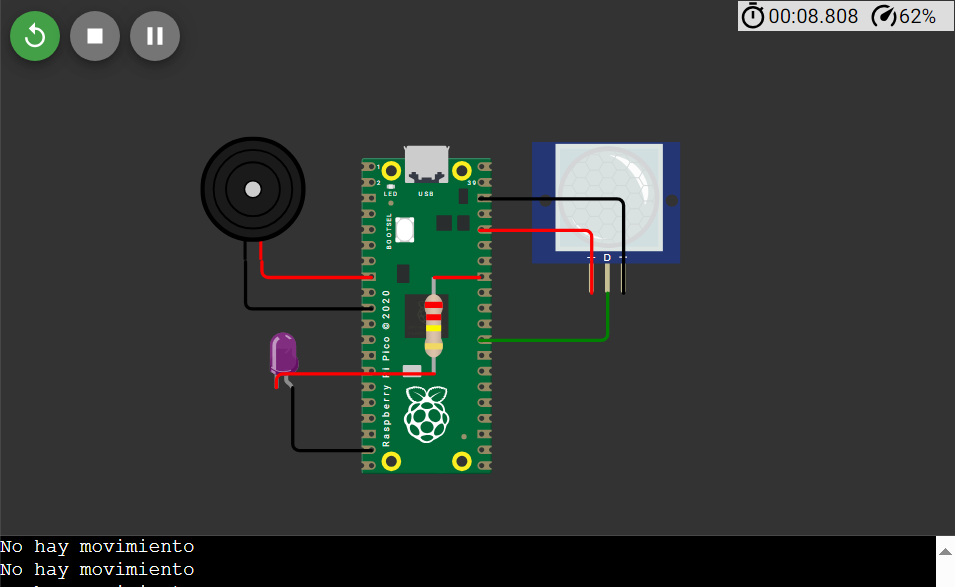
**IMAGEN 10**

Sin embargo, si se le da la orden correcta se accionará el mecanismo y esto se verá reflejado en el led rojo

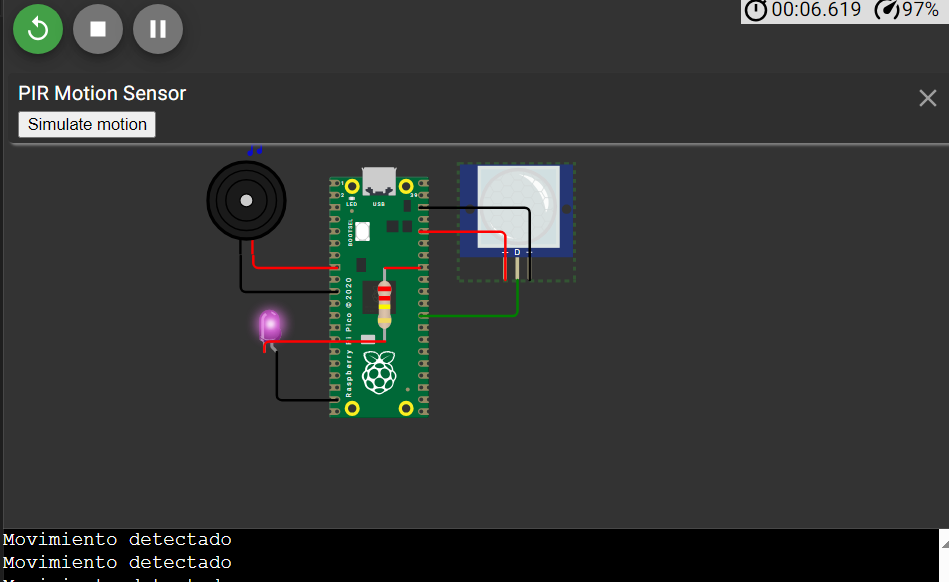
**IMAGEN 11**

### Simulador de timbre automático:

Para simular este timbre al igual que el mecanismo anterior se usó un sensor de movimiento, un led y una buzzer o bocina que simbolizará el timbre.

El mecanismo de este modelado es fácil de entender simplemente si detecta movimiento se activa el led y la bocina después de un tiempo de su encendido se vuelve a pagar automáticamente.

**IMAGEN 12**



**IMAGEN 13**

Finalmente se mostrará los links que se usó para las simulaciones y modelados:

* Primera simulación: <https://wokwi.com/projects/424106867538896897>
* Segunda simulación: <https://wokwi.com/projects/424104416157387777>
* Tercera simulación: <https://wokwi.com/projects/424085388330168321>
* Cuarta simulación: <https://wokwi.com/projects/424081384835336193>

## Componentes de un Smart Home

Un sistema de casa inteligente consta de los siguientes componentes principales:

* **Interfaz de usuario:** Permite al usuario interactuar con el sistema, ya sea a través de una aplicación móvil, una pantalla táctil o comandos de voz.
* **Modo de transmisión:**Es el medio por el cual se comunican los dispositivos, como Wi-Fi, Bluetooth, Zigbee, etc.
* **Controlador central:** Es el cerebro del sistema, que recibe las órdenes del usuario y las transmite a los dispositivos correspondientes.

## Aplicación de POO en un Sistema Smart Home

Para modelar un sistema Smart Home en Python utilizando POO, se pueden aplicar los siguientes conceptos:

### Clases y Objetos:

* Cada dispositivo (luces, ventiladores, sensores, etc.) puede ser representado como una clase.
* Los objetos serían instancias de estas clases, representando dispositivos específicos en el hogar.

### Herencia:

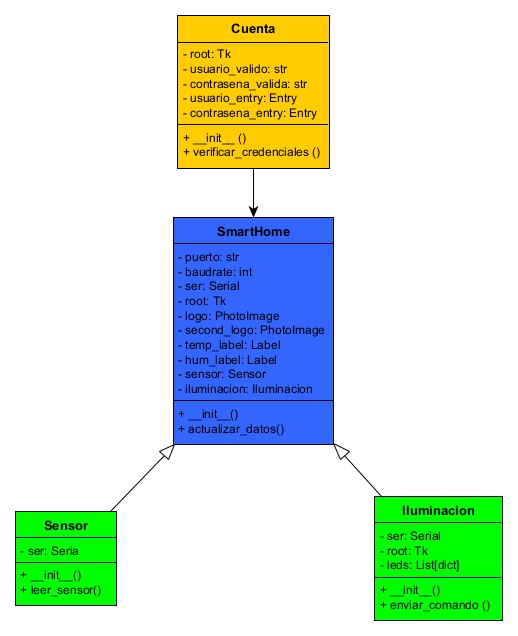
* Se puede crear una clase base llamada Dispositivo que contenga atributos y métodos comunes a todos los dispositivos (como encender, apagar, etc.).
* Luego, se pueden crear clases derivadas como Luz, Ventilador, Sensor, que hereden de la clase Dispositivo y añadan funcionalidades específicas.

### Encapsulamiento:

* Los atributos de cada dispositivo (como el estado de encendido/apagado) pueden ser encapsulados para protegerlos de modificaciones no autorizadas.
* Se pueden usar métodos públicos para interactuar con estos atributos.

### Polimorfismo:

* Diferentes dispositivos pueden tener métodos con el mismo nombre, pero comportamientos diferentes. Por ejemplo, el método encender () puede tener una implementación distinta para una luz y un ventilador.



**IMAGEN 5: NUESTRO DIAGRAMA DE CLASES**

## Análisis del Diagrama

### Clase Cuenta

**-ATRIBUTOS:**

* **root:** Instancia de Tk para la ventana principal de la interfaz gráfica.
* **usuario\_valido:** Almacena el nombre de usuario válido.
* **contrasena\_valida:** Almacena la contraseña válida.
* **usuario\_entry:** Campo de entrada (Entry) para que el usuario ingrese su nombre de usuario.
* **contrasena\_entry:** Campo de entrada (Entry) para que el usuario ingrese su contraseña.

**-MÉTODOS:**

* **\_\_init\_\_():** Constructor de la clase, inicializa los atributos y configura la interfaz gráfica.
* **verificar\_credenciales():** Verifica si las credenciales ingresadas por el usuario son válidas.

### Clase Smarthome

**-ATRIBUTOS:**

* **puerto:** Puerto serial utilizado para la comunicación con los dispositivos.
* **baudrate:** Velocidad de transmisión de datos en baudios.
* **ser:** Instancia de Serial para la comunicación serial.
* **root:** Instancia de Tk para la ventana principal de la interfaz gráfica.
* **logo y second\_logo:** Imágenes (PhotoImage) utilizadas en la interfaz gráfica.
* **temp\_label y hum\_label:** Etiquetas (Label) para mostrar la temperatura y la humedad.
* **sensor:** Instancia de la clase Sensor para leer datos del sensor.
* **iluminacion:** Instancia de la clase Iluminacion para controlar la iluminación.

**-MÉTODOS:**

* **\_\_init\_\_():** Constructor de la clase, inicializa los atributos y configura la interfaz gráfica.
* **actualizar\_datos():** Actualiza los datos mostrados en la interfaz gráfica, como la temperatura y la humedad.

### Clase Sensor

**-ATRIBUTOS:**

* **ser:** Instancia de Serial para la comunicación serial con el sensor.

**-MÉTODOS:**

* \_\_init\_\_(): Constructor de la clase, inicializa la comunicación serial.
* leer\_sensor(): Lee los datos del sensor (por ejemplo, temperatura y humedad).

### Clase Iluminacion

**-ATRIBUTOS:**

* **ser:** Instancia de Serial para la comunicación serial con los dispositivos de iluminación.
* **root:** Instancia de Tk para la interfaz gráfica.
* **leds:** Lista de diccionarios que almacena información sobre los LEDs controlados.

**-MÉTODOS:**

* **\_\_init\_\_():** Constructor de la clase, inicializa los atributos y configura la interfaz gráfica.
* **enviar\_comando():** Envía comandos a los dispositivos de iluminación para encender, apagar o ajustar los LEDs.

### Clase Ventilación

**-ATRIBUTOS:**

* **ser:** Instancia de Serial para la comunicación serial con los dispositivos de ventilación.
* **root:** Instancia de Tk para la interfaz gráfica.

**-MÉTODOS:**

* **\_\_init\_\_():** Constructor de la clase, inicializa los atributos y configura la interfaz gráfica.
* **enviar\_comando():** Envía comandos a los dispositivos de ventilación para controlar su funcionamiento.

### Clase Cerradura

**-ATRIBUTOS:**

* ser: Instancia de Serial para la comunicación serial con los dispositivos de cerradura.
* root: Instancia de Tk para la interfaz gráfica.

**-MÉTODOS:**

* \_\_init\_\_(): Constructor de la clase, inicializa los atributos y configura la interfaz gráfica.
* enviar\_comando (): Envía comandos a los dispositivos de cerradura para abrir o cerrar la cerradura.

## Relaciones entre Clases

### Clase Cuenta:

* Es independiente y se encarga de la autenticación del usuario. Una vez que las credenciales son verificadas, se puede acceder al sistema Smarthome.

### Clase Smarthome:

* Es la clase principal que gestiona la interfaz gráfica y coordina las demás clases (Sensor, Iluminacion, Ventilación, Cerradura).
* Utiliza instancias de Sensor para leer datos y de Iluminacion para controlar la iluminación.

### Clases Sensor, Iluminacion, Ventilación, Cerradura:

* Estas clases están relacionadas con la clase Smarthome y se encargan de interactuar con los dispositivos físicos a través de la comunicación serial.
* Cada clase tiene un método enviar\_comando () para enviar instrucciones a los dispositivos correspondientes.

## Riesgos y Desventajas

A pesar de sus ventajas, los sistemas Smart Home también presentan ciertos riesgos y desventajas:

* **Ciberseguridad:** Los dispositivos conectados a internet pueden ser vulnerables a ataques cibernéticos, permitiendo que hackers tomen el control de los dispositivos.
* **Interceptación de comunicación:**La comunicación entre dispositivos puede ser interceptada, exponiendo información sensible.
* **Vigilancia no autorizada:** Dispositivos con cámaras pueden ser utilizados para espiar el hogar.
* **Fallos técnicos:**Los dispositivos pueden sufrir fallos, interrumpiendo la vida diaria.
* **Cortes de energía:**Dependen de un suministro eléctrico constante; un corte de energía puede dejar el sistema inoperativo.
* **Costo inicial elevado:**La inversión inicial para implementar un sistema Smart Home puede ser costosa.
* **Mantenimiento y actualizaciones:** Los dispositivos requieren mantenimiento y actualizaciones regulares para funcionar correctamente.