



*Vigilada Mineducación*

## Proyecto final "AIR": Dispositivo de protección basado en inteligencia artificial

---

Desarrollado por: Luis Fernando Molina Antequera

## Manual de usuario

---

### Arquitectura del dispositivo

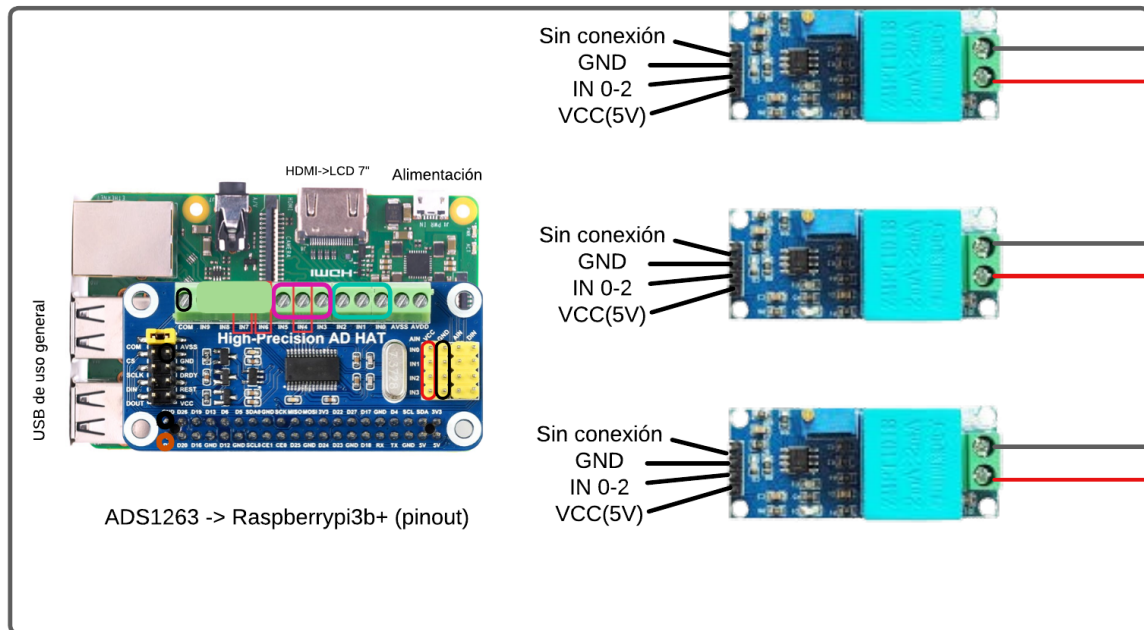
Este dispositivo está constituido por los siguientes elementos:

- 3x Sensor ZMPT101B
- 3x Sensor SCT030
- ADC ADS1263 (*waveshare*)
- Raspberry Pi 3B+
- Pantalla LED 7" HDMI

### Imágenes del dispositivo

#### Conexión del dispositivo

Antes de empezar a utilizar el dispositivo, debe realizar y/o revisar la conexión de todos los elementos siguiendo el diagrama mostrado a continuación:



## Primeros pasos

Si es la primera vez que enciende el dispositivo, deberá tener en cuenta las indicaciones del [primer encendido](#) de una Raspberry Pi 3B+. Una vez haya concluido con esto, deberá habilitar la interfaz de comunicación SPI del dispositivo. Para realizar esta tarea, debe abrir una terminal de comandos ejecutando la combinación de teclas **Ctrl+Alt+T**.

Esto desplegará la interfaz de comandos del dispositivo, en donde, ejecutará el siguiente comando: **sudo raspi-config**. Este comando desplegará el menú de opciones del dispositivo, en este menú deberá seleccionar la opción **Interface options** y luego, **Enable SPI interface**.

Una vez realizado esto, deberá reiniciar el dispositivo para que las opciones establecidas se guarden.

Por último, deberá descargar o clonar este repositorio.

## Requisitos generales

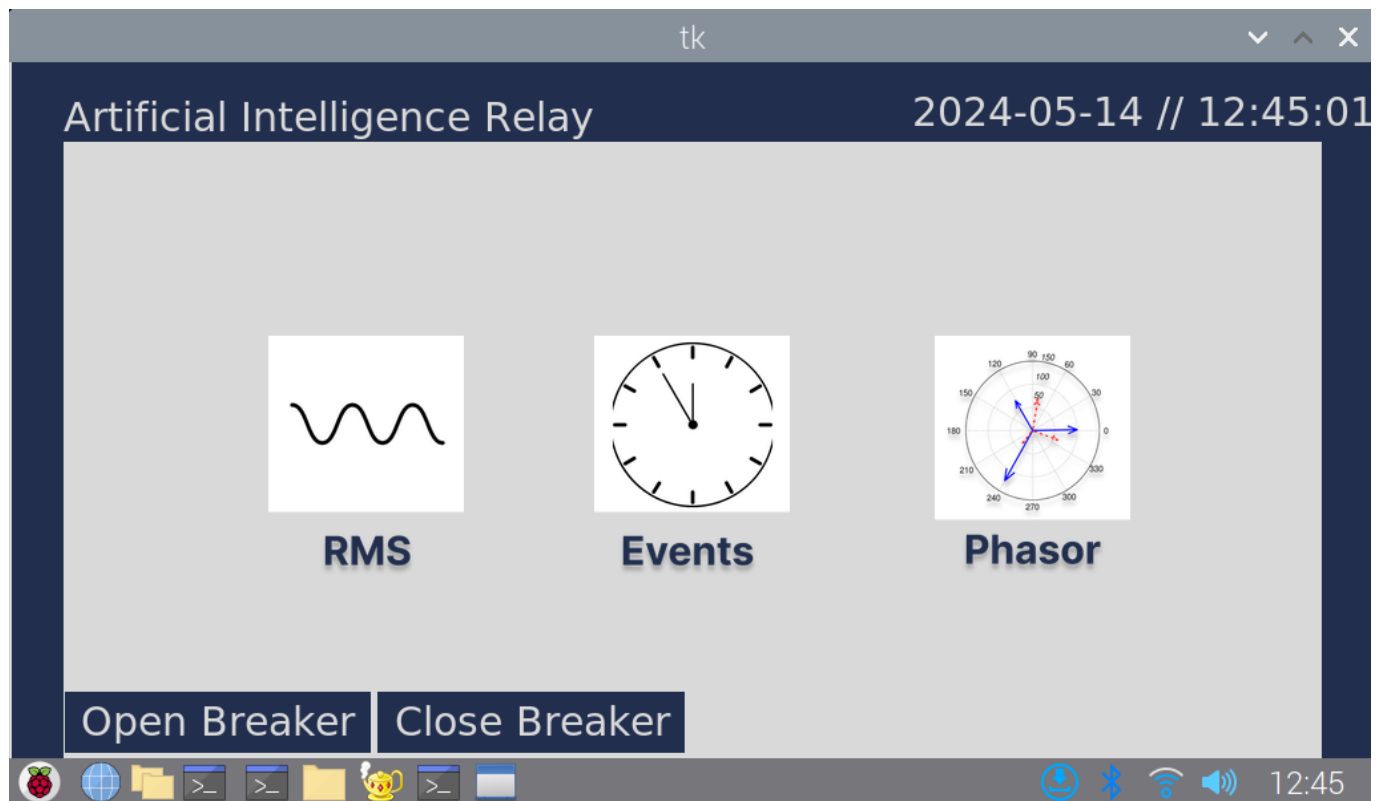
- Python (3.1 o superior)

## Requisitos Python

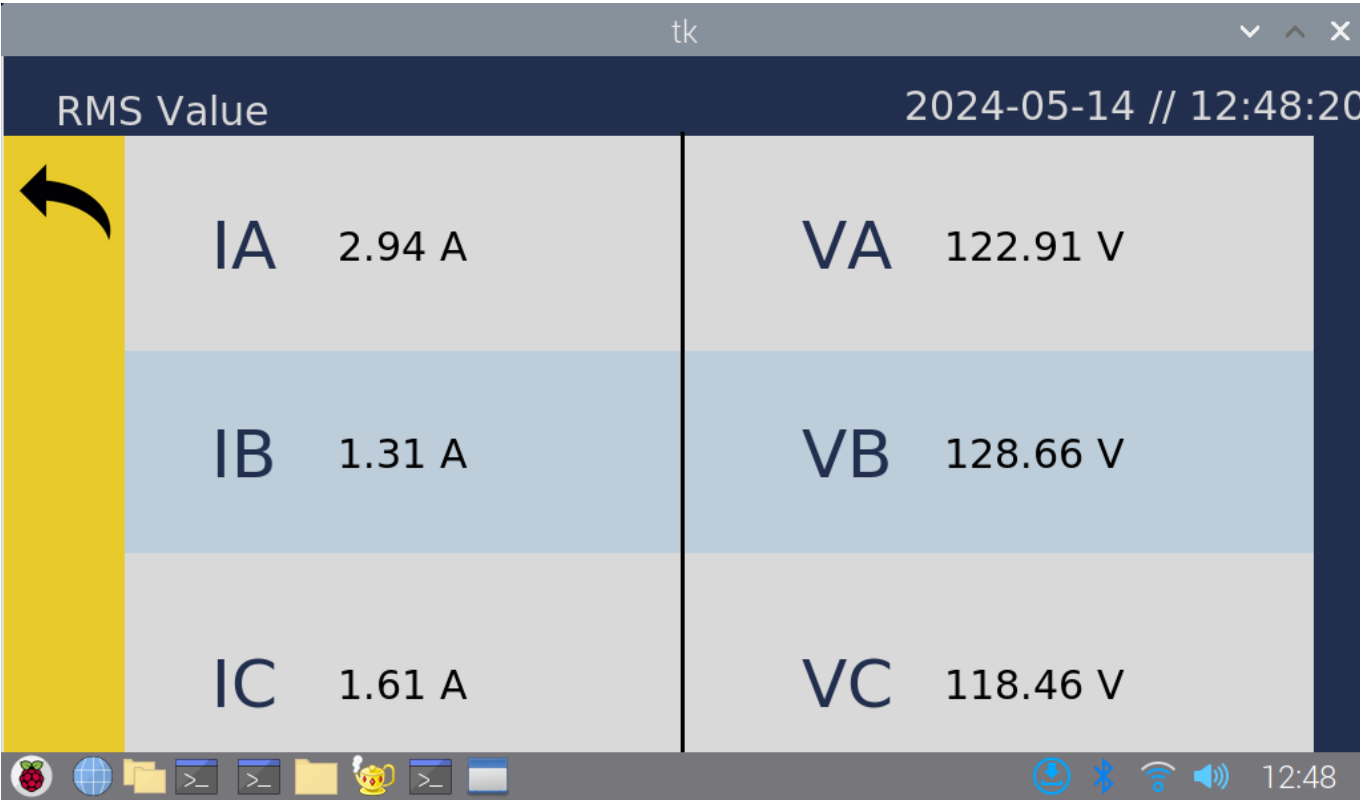
Diríjase a la carpeta principal de este repositorio. En esta, encontrará un archivo llamado `requirements.txt`, inicie una terminal de comandos en esta carpeta y ejecute el comando `pip install -r requirements.txt` o `pip3 install -r requirements.txt` dependiendo de la versión instalada de Python.

### ¿Cómo utilizar el dispositivo?

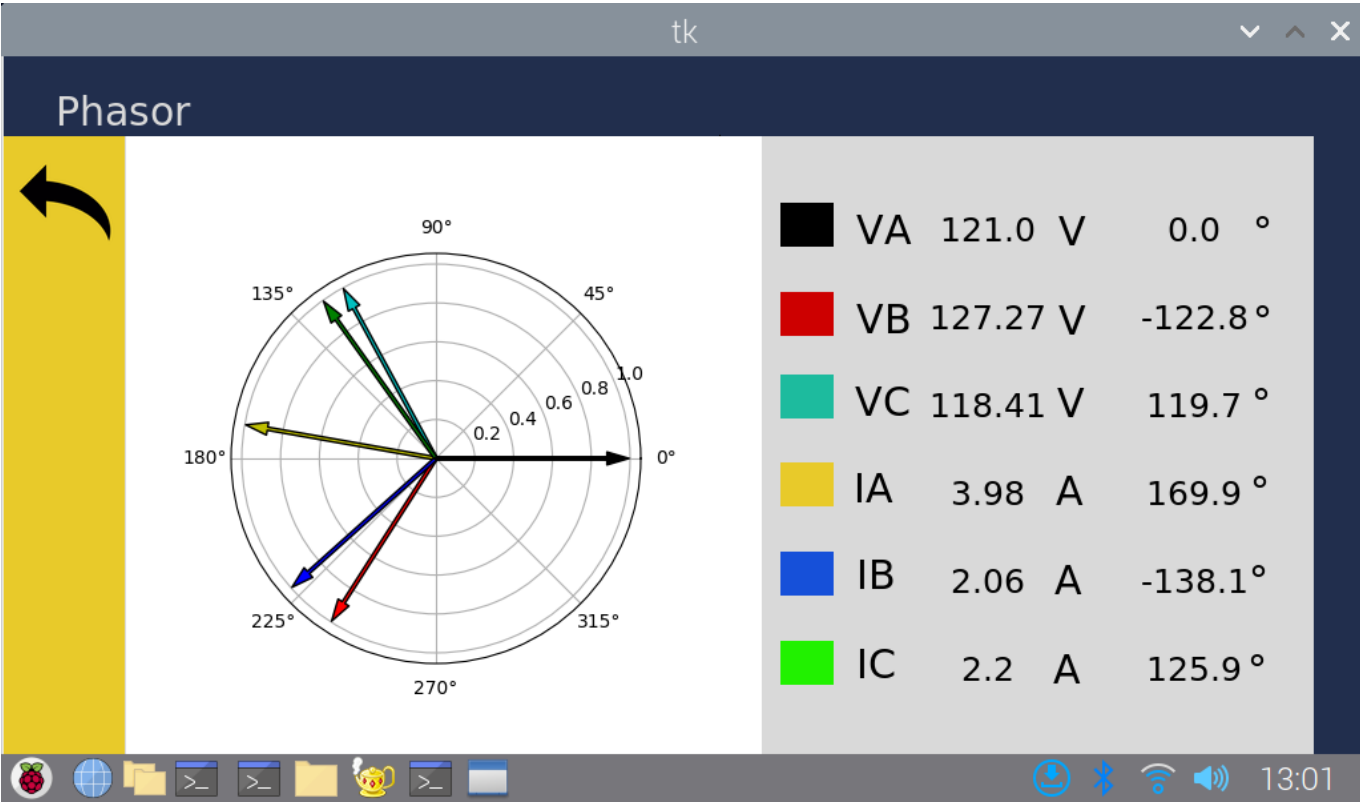
Ingresa a la carpeta del repositorio, luego a la carpeta main y ejecute el archivo de Python "gui.py". Esto desplegará la ventana principal de la interfaz con la que funciona el dispositivo. En esta se encuentran las 3 opciones de visualización de parámetros. Adicionalmente, usted tendrá la opción de enviar una señal de disparo o de apertura al dispositivo interruptor que esté utilizando.



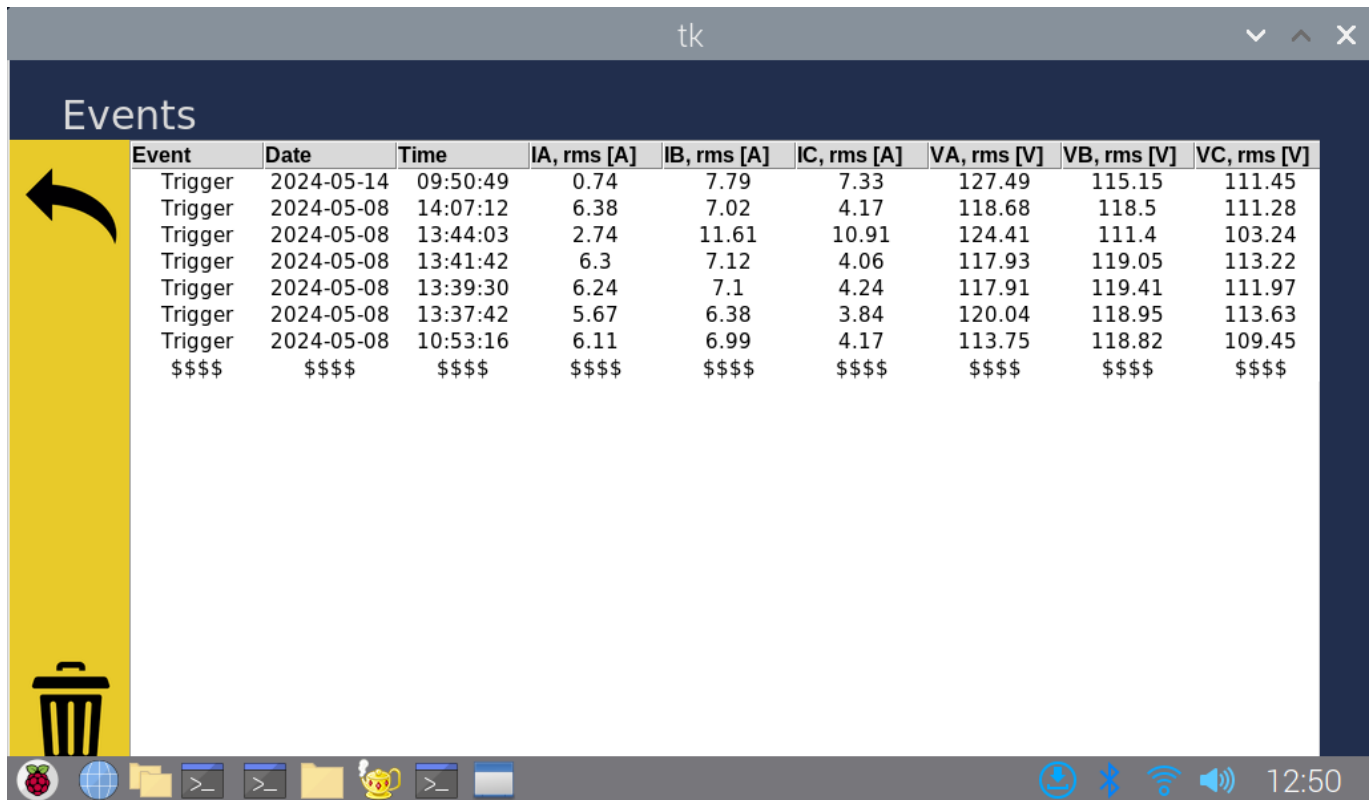
RMS le otorgará la vista de las tensiones y corrientes RMS captadas por los sensores de tensión y corriente.



En la ventana Phasors usted podrá observar adicionalmente el desfase angular que existe entre todas estas señales teniendo como referencia aquella señal conectada al canal 0.



Por último, la pestaña events almacena todos los eventos que el dispositivo registre como falla, en esta, usted tendrá la opción de reiniciar estos datos. La información correspondiente también la podrá extraer adicionalmente de un archivo de Excel presente en la carpeta `main/Assets` del repositorio.



Event	Date	Time	IA, rms [A]	IB, rms [A]	IC, rms [A]	VA, rms [V]	VB, rms [V]	VC, rms [V]
Trigger	2024-05-14	09:50:49	0.74	7.79	7.33	127.49	115.15	111.45
Trigger	2024-05-08	14:07:12	6.38	7.02	4.17	118.68	118.5	111.28
Trigger	2024-05-08	13:44:03	2.74	11.61	10.91	124.41	111.4	103.24
Trigger	2024-05-08	13:41:42	6.3	7.12	4.06	117.93	119.05	113.22
Trigger	2024-05-08	13:39:30	6.24	7.1	4.24	117.91	119.41	111.97
Trigger	2024-05-08	13:37:42	5.67	6.38	3.84	120.04	118.95	113.63
Trigger	2024-05-08	10:53:16	6.11	6.99	4.17	113.75	118.82	109.45
\$\$\$\$	\$\$\$\$	\$\$\$\$	\$\$\$\$	\$\$\$\$	\$\$\$\$	\$\$\$\$	\$\$\$\$	\$\$\$\$

### ¿Cómo cargar un modelo de aprendizaje de máquina?

Diríjase al archivo "gui.py" en la carpeta "Main". Edite la línea 619:

`Model=joblib.load(relative_to_assets("NN2.joblib"))` de forma tal que quede de la siguiente manera: `Model=joblib.load(relative_to_assets("Su_Modelo_de_IA.joblib"))`.

Adicionalmente, deberá seleccionar los parámetros con los que usted entrenó su modelo y planea hacer predicciones. Para realizar esto, diríjase a la línea 659 `Estado=np.array([IA,IB,IC])` e ingrese los parámetros en el orden con el que entrenó su modelo. Los parámetros disponibles actualmente se listan a continuación:

- VA: Magnitud de tensión de la fase A
- VB: Magnitud de tensión de la fase B
- VC: Magnitud de tensión de la fase C
- IA: Magnitud de corriente de la fase A
- IB: Magnitud de corriente de la fase B
- IC: Magnitud de Corriente de la fase C
- VAP: Ángulo de la tensión en la fase A
- VBP: Ángulo de la tensión en la fase B
- VCP: Ángulo de la tensión en la fase C
- IAP: Ángulo de la corriente en la fase A
- IBP: Ángulo de la corriente en la fase B
- ICP: Ángulo de la corriente en la fase C

Suponiendo que usted use un modelo que utilice las tensiones y sus magnitudes en este orden: VA, VB, VC, VAP, VBP, VCP, usted deberá cambiar la línea de la siguiente manera:

`Estado=np.array([VA, VB, VC, VAP, VBP, VCP])`

## Notas

- Este dispositivo solo admite modelos de redes neuronales entrenadas con las librerías scikit-learn de Python.
- Debido al alto consumo de recursos de la función Phasors, la habilidad de proteger el sistema usando el modelo solo se encuentra habilitada en la función de RMS. (Esto no implica que no se puedan usar los ángulos en los modelos que usted desee entrenar).
- En caso de observar un comportamiento "lento" del dispositivo luego de utilizar la ventana Phasors, se recomienda cerrar la aplicación y volverla a abrir.