

Taller Raspberry Pi 3: Exploración de Raspbian, Dashboards, Automatización y Auditoría de Red

Miguel Ángel Caro Molano y Samuel Esteban Parra Osorio
Universidad Santo Tomás
Ingeniería Electrónica

17 de octubre de 2025

Índice

1. Introducción	2
2. Instalación y configuración de Raspbian	2
2.1. Descarga e instalación	2
2.2. Configuraciones iniciales	2
3. Dashboard con Streamlit	2
3.1. Instalación del entorno	2
3.2. Creación de la aplicación	3
4. Exploración de Grafana	3
4.1. Instalación de Grafana	3
4.2. Ejecución y configuración	3
5. Automatización con Cron y Crontab	4
5.1. Creación de un script de respaldo	4
5.2. Programación automática con Crontab	4
6. Exploración del entorno de red y auditoría	4
6.1. Visualización de la IP local	4
6.2. Exploración de dispositivos vecinos	4
6.3. Exploración de puertos	5
6.4. Auditoría básica	5
7. Conclusiones	5
8. Referencias	5

1. Introducción

El presente documento describe el desarrollo del taller sobre el manejo y exploración de la Raspberry Pi 3, abarcando desde la instalación del sistema operativo Raspbian hasta la creación de dashboards con Streamlit y Grafana, la automatización de tareas mediante cron y crontab, y la exploración del entorno de red del dispositivo.

Durante el proceso se documentaron las etapas principales con capturas de pantalla y evidencias, las cuales estarán anexas en la **carpeta de evidencias** dentro del repositorio de GitHub denominado **taller**.

2. Instalación y configuración de Raspbian

2.1. Descarga e instalación

Se descargó el sistema operativo **Raspberry Pi OS (Raspbian)** desde la página oficial <https://www.raspberrypi.com/software/>. Posteriormente, se utilizó la herramienta **Raspberry Pi Imager** para grabar la imagen en una memoria microSD.

1. Seleccionar el sistema operativo: *Raspberry Pi OS (32-bit)*.
2. Elegir la tarjeta microSD correspondiente.
3. Pulsar en *Write* para iniciar el proceso de escritura.

Una vez completado, la microSD se insertó en la Raspberry Pi 3, conectando monitor, teclado, mouse y red WiFi. Las capturas del proceso de instalación se encuentran anexas en la carpeta **carpeta de evidencias**.

2.2. Configuraciones iniciales

Durante el primer arranque se configuraron los parámetros regionales (idioma, zona horaria y teclado), se conectó la red WiFi y se ejecutaron las actualizaciones del sistema mediante:

```
sudo apt update && sudo apt upgrade -y
```

3. Dashboard con Streamlit

3.1. Instalación del entorno

Para el desarrollo del dashboard se instaló la librería **Streamlit** en Python:

```
sudo apt install python3-pip -y  
pip3 install streamlit pandas numpy
```

3.2. Creación de la aplicación

Se generó un archivo `app.py` con el siguiente contenido:

```
import streamlit as st
import pandas as pd
import numpy as np

st.title("Dashboard de Prueba con Streamlit")
st.write("Visualización de datos simulados en tiempo real")

data = pd.DataFrame(
    np.random.randn(10, 3),
    columns=['Temperatura ( C )', 'Humedad (%)', 'Presión (hPa)']
)

st.line_chart(data)
```

La aplicación se ejecuta con:

```
streamlit run app.py
```

Al abrir la dirección local (<http://localhost:8501>) se visualiza el dashboard generado. Las imágenes de la interfaz de Streamlit y su ejecución se encuentran en la carpeta **carpeta de evidencias** del repositorio.

4. Exploración de Grafana

4.1. Instalación de Grafana

Para la visualización avanzada de datos, se instaló el sistema Grafana utilizando los siguientes comandos:

```
sudo apt-get install -y apt-transport-https software-properties-common wget
sudo mkdir -p /etc/apt/keyrings/
wget -q -O - https://apt.grafana.com/gpg.key | sudo gpg --dearmor -o /etc/apt/keyrings/grafana.gpg
echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
sudo apt-get update
sudo apt-get install grafana -y
```

4.2. Ejecución y configuración

Una vez instalado, el servicio se habilitó y se inició con:

```
sudo systemctl enable grafana-server
sudo systemctl start grafana-server
```

Grafana se ejecuta en <http://localhost:3000>, con usuario y contraseña predeterminados `admin / admin`. Se cambió la contraseña y se añadió la fuente de datos *TestData*

DB para crear un dashboard de ejemplo con gráficas de tipo línea y gauge. Las capturas de este proceso también están disponibles en la **carpeta de evidencias**.

5. Automatización con Cron y Crontab

5.1. Creación de un script de respaldo

Se elaboró un script de respaldo automático llamado `backup.sh`:

```
#!/bin/bash
tar -czf /home/pi/respaldo_$(date +%F).tar.gz /home/pi/Documents
```

El script genera un archivo comprimido con fecha actual y lo guarda en el directorio personal del usuario.

5.2. Programación automática con Crontab

Se programó la ejecución automática diaria del script a las 2:00 a.m. mediante:

```
crontab -e
```

Y se agregó la línea:

```
0 2 * * * /home/pi/backup.sh
```

Esto permite ejecutar la tarea sin intervención humana, garantizando la creación diaria de copias de respaldo. Una captura del crontab configurado se encuentra en la **carpeta de evidencias**.

6. Exploración del entorno de red y auditoría

6.1. Visualización de la IP local

Se identificó la dirección IP de la Raspberry Pi con:

```
hostname -I
```

6.2. Exploración de dispositivos vecinos

Para listar los equipos conectados en la misma red se empleó:

```
arp -a
```

y con la herramienta **nmap**:

```
sudo apt install nmap -y
nmap -sn 192.168.1.0/24
```

6.3. Exploración de puertos

Para verificar los puertos abiertos localmente:

```
nmap -p 1-1024 localhost
```

El resultado mostró los servicios activos, como SSH y HTTP, lo cual permitió identificar configuraciones seguras y posibles vulnerabilidades.

6.4. Auditoría básica

Se realizó una auditoría sencilla de la red, verificando los puertos más sensibles y los dispositivos conectados. Las evidencias gráficas de los escaneos de red se anexan igualmente en la **carpeta de evidencias**.

7. Conclusiones

- Se logró la instalación completa del sistema Raspbian y la configuración básica de la Raspberry Pi 3.
- Se implementó un dashboard funcional con Streamlit que permite la visualización de datos simulados.
- Grafana se configuró exitosamente como herramienta de monitoreo y visualización.
- Se desarrolló una tarea automatizada mediante cron y crontab, mostrando la capacidad del sistema para ejecutar procesos periódicos sin supervisión.
- Finalmente, se exploró el entorno de red y se realizó una auditoría básica, fortaleciendo los conocimientos sobre seguridad y análisis de red.

8. Referencias

- Raspberry Pi Foundation. (2025). *Raspberry Pi OS Documentation*. Recuperado de: <https://www.raspberrypi.com/software/>
- Streamlit Inc. (2025). *Streamlit Documentation*. Recuperado de: <https://docs.streamlit.io/>
- Grafana Labs. (2025). *Grafana Documentation*. Recuperado de: <https://grafana.com/docs/>
- Linux Man Pages. *Cron and Crontab*. Recuperado de: <https://linux.die.net/man/5/crontab>
- Nmap Project. (2025). *Network Scanning Guide*. Recuperado de: <https://nmap.org/book/>