

Proyecto de sistema automatizado de Backups de Switches.

Descripción del Proyecto.

El proyecto consistió en la implementación de un sistema de respaldo para las configuraciones de los switches instalados en el campamento minero. Estos dispositivos poseen configuraciones críticas, ya que permiten la interconexión de servicios esenciales como Internet, intranet, telefonía IP, sistema de alerta temprana, sistema SCADA, servidores y cámaras IP.

La necesidad de este proyecto surgió debido a la ausencia de un sistema de respaldos automatizado en la empresa, a pesar de contar con una gran cantidad de dispositivos en producción distribuidos en distintas locaciones remotas del campamento. Ante esta situación, se diseñó e implementó una solución que garantizara la disponibilidad y recuperación rápida de las configuraciones en caso de fallas o daños en los equipos.

Procedimientos de pruebas para la implementación.

Como parte inicial, en un ambiente de laboratorio aislado de la red corporativa se utilizó una computadora y un switch Cisco Catalyst 2960-XR Series.

Luego se procedió a la instalación y configuración de un servidor FTP utilizando el software **FileZilla Server**, el cual funciona como repositorio central de las copias de respaldo. Este servidor permite almacenar de manera local las configuraciones más recientes de los switches y facilitar su recuperación ante una contingencia, minimizando el tiempo de inactividad y asegurando la continuidad de los servicios con la última configuración disponible. Además se instaló el software Python para llevar a cabo el backup de las configuraciones de los switches, y como librerías se usaron:

1. Custom Tkinter:

- a. **Descripción:** Esta librería es una extensión de Tkinter, que permite crear interfaces gráficas de usuarios (GUI) más modernas y personalizadas
- b. **Uso en el Proyecto:** Se utilizó para construir la interfaz del sistema de respaldo, permitiendo a los usuarios interactuar fácilmente con las funciones del software, como agregar, editar y eliminar configuraciones de switches.

Instalación: `py -m pip install customtkinter`

2. Paramiko:

- a. **Descripción:** Paramiko es una librería de Python que proporciona una implementación del protocolo SSH2, permitiendo la conexión segura a dispositivos remotos. Facilita la ejecución de comandos en dispositivos a través de SSH y la transferencia de archivos.
- b. **Uso en el Proyecto:** Se utilizó para establecer conexiones SSH con los switches, permitiendo la ejecución de comandos para obtener las configuraciones actuales y realizar copias de seguridad de manera automatizada.

3. CTkMessageBox:

- a. **Descripción:** Esta librería es parte de CustomTkinter y proporciona cuadros de mensaje personalizados que permiten mostrar información, advertencias o errores a los usuarios de manera visualmente atractiva.
- b. **Uso en el Proyecto:** Se utilizó para mostrar mensajes de confirmación y errores durante la interacción del usuario con la interfaz, mejorando la comunicación y la experiencia del usuario.

Instalación: `py -m pip install ctkmessagebox`

4. Threading

- a. **Descripción:** Esta librería estándar de Python permite la creación y gestión de hilos (threads), lo que permite ejecutar múltiples operaciones simultáneamente.
- b. **Uso en el Proyecto:** Se utilizó para ejecutar tareas de respaldo y validación de conexiones en segundo plano, evitando que la interfaz se congele y mejorando la experiencia del usuario.

Luego de que el programa realice el backup de manera correcta y se programe automáticamente para el siguiente mes, se pasó a otro nivel más alto de pruebas. En esta fase, se llevaron a cabo simulaciones para evaluar la resiliencia del sistema ante fallos.

5. Simulación de Fallos:

- a. **Apagado del Switch:** Se apagó el switch manualmente para simular un fallo de hardware. Esto permitió evaluar cómo el sistema respondía ante la pérdida de un dispositivo crítico.
- b. **Desconexión de la Red:** Se retiró el cable de red del switch, simulando una pérdida de conectividad.. Esta prueba fue crucial para verificar que

el sistema pudiera manejar situaciones en las que los dispositivos no estuvieran accesibles.

- c. **Cambio de Fecha/Hora:** Se modificó la fecha y hora de la computadora de prueba para simular que se había llegado al mes en el que debía realizarse el backup programado. Esto ayudó a comprobar que el sistema pudiera ejecutar correctamente las tareas programadas en el momento adecuado.

Proceso de Implementación

El proceso de implementación del sistema de backups automatizado de switches se llevó a cabo en varias fases:

1. **Entorno de Pruebas Inicial**
2. **Pruebas Avanzadas y Simulación de Fallos**
3. **Implementación en Producción:**
 - a. Una vez superadas las pruebas en el entorno de laboratorio y las simulaciones de fallos, se procedió a la implementación del sistema en el entorno de producción del campamento minero.
 - b. Se configuró el software de backup para realizar copias de seguridad automáticas de los switches según la programación establecida.

Nota: Los puntos 1 y 2 fueron explicadas anteriormente en la sección “Procedimiento para la prueba de implementación”

En resumen, el proceso de implementación incluyó pruebas exhaustivas en un entorno aislado, simulaciones de fallos y finalmente la implementación en el entorno de producción, asegurando la confiabilidad y resiliencia del sistema de backups.

Flujo de Trabajo

1) Agregar un nuevo Switch

a. Diagrama de flujo

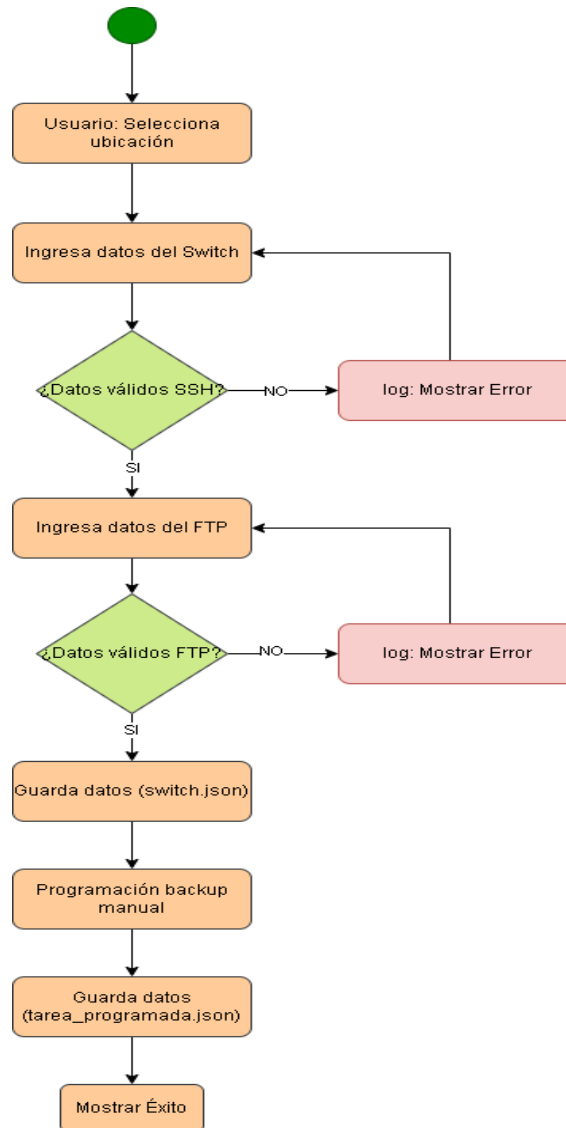


Figura 1: Diagrama de flujo de agregar switch

b. Pasos:

- i. Usuario selecciona ubicación (San Juan, Jachal, Gualcamayo)
- ii. Ingresa IP, usuario, contraseña y enable password (opcional)
- iii. Validación SSH en tiempo real:
- iv. Conexión al switch
- v. Verificación de privilegios

- vi. Ingreso de credenciales FTP (valores predeterminados pre-cargados)
- vii. Validación FTP:
- viii. Creación de estructura de directorios automática:
/[código_ubicación]/[último_octeto]/BKP-Mensual
- ix. Guardado en switches.json
- x. Programación automática de backup mensual (día 5 a las 3:00 AM)
- xi. Guardado de tarea en tareas_programadas.json

2) Edición de Switches

a. Diagrama de Flujo:

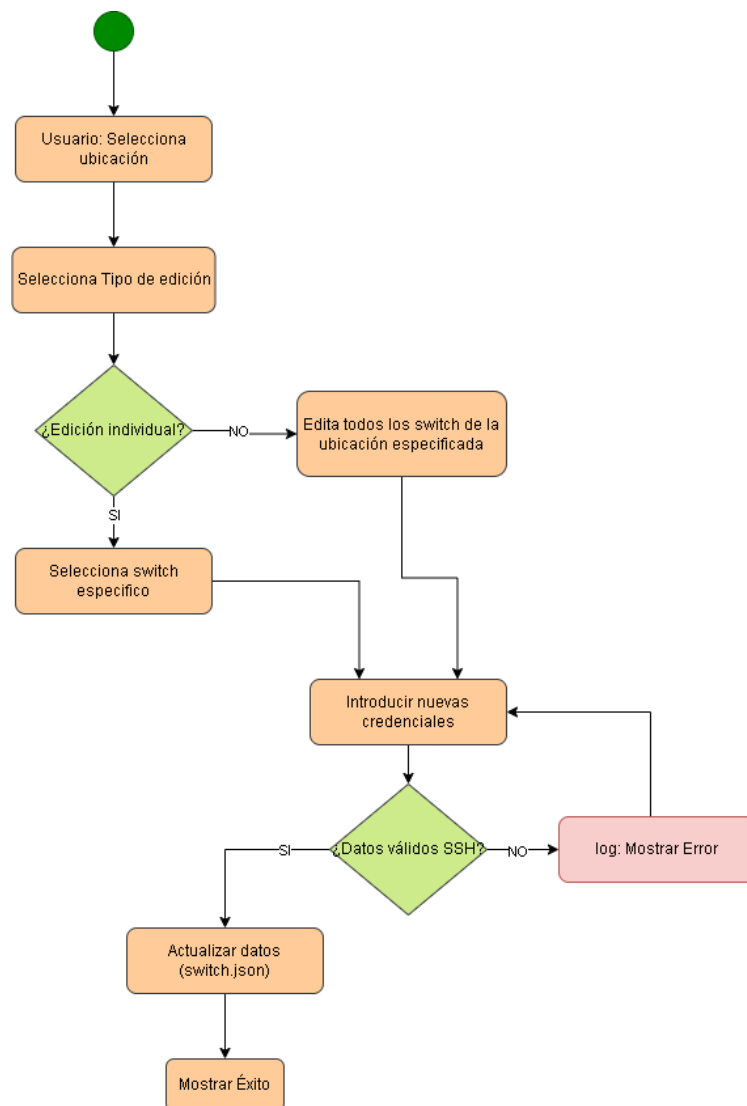


Figura 2: Diagrama de flujo editar switch

b. Características únicas:

- i. **Edición masiva:** Actualiza todos switches de una ubicación simultáneamente
- ii. **Validación inteligente:** Solo valida campos modificados
- iii. **Propagación automática:** Actualiza tareas programadas relacionadas

3) Eliminación de Switch

a. Diagrama de flujo

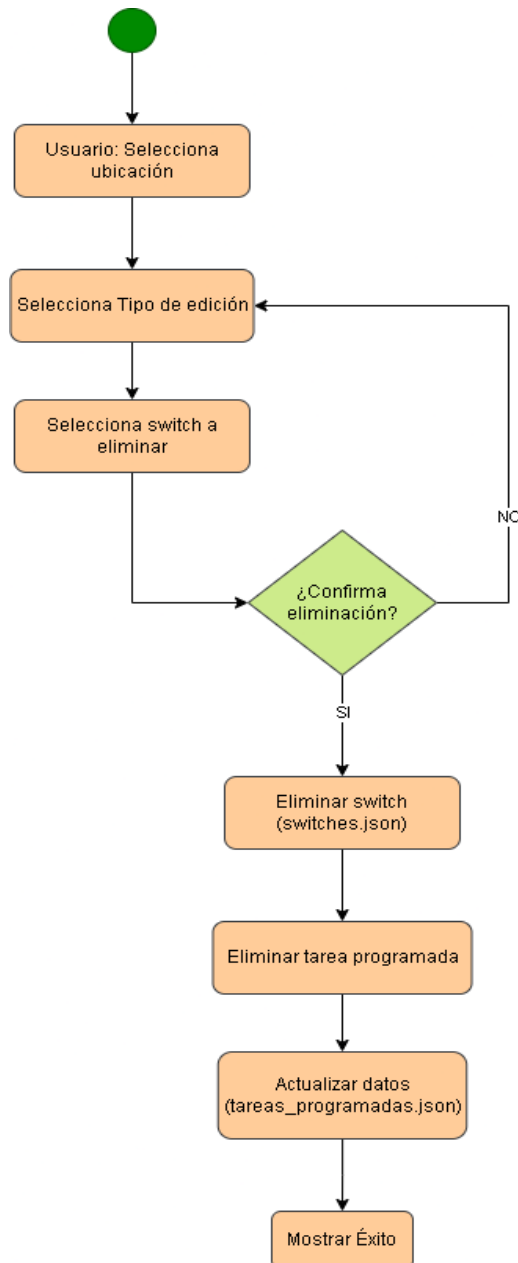


Figura 3: Diagrama de flujo de eliminar un switch

4) Gestión de tareas programadas

a. Diagrama de flujo

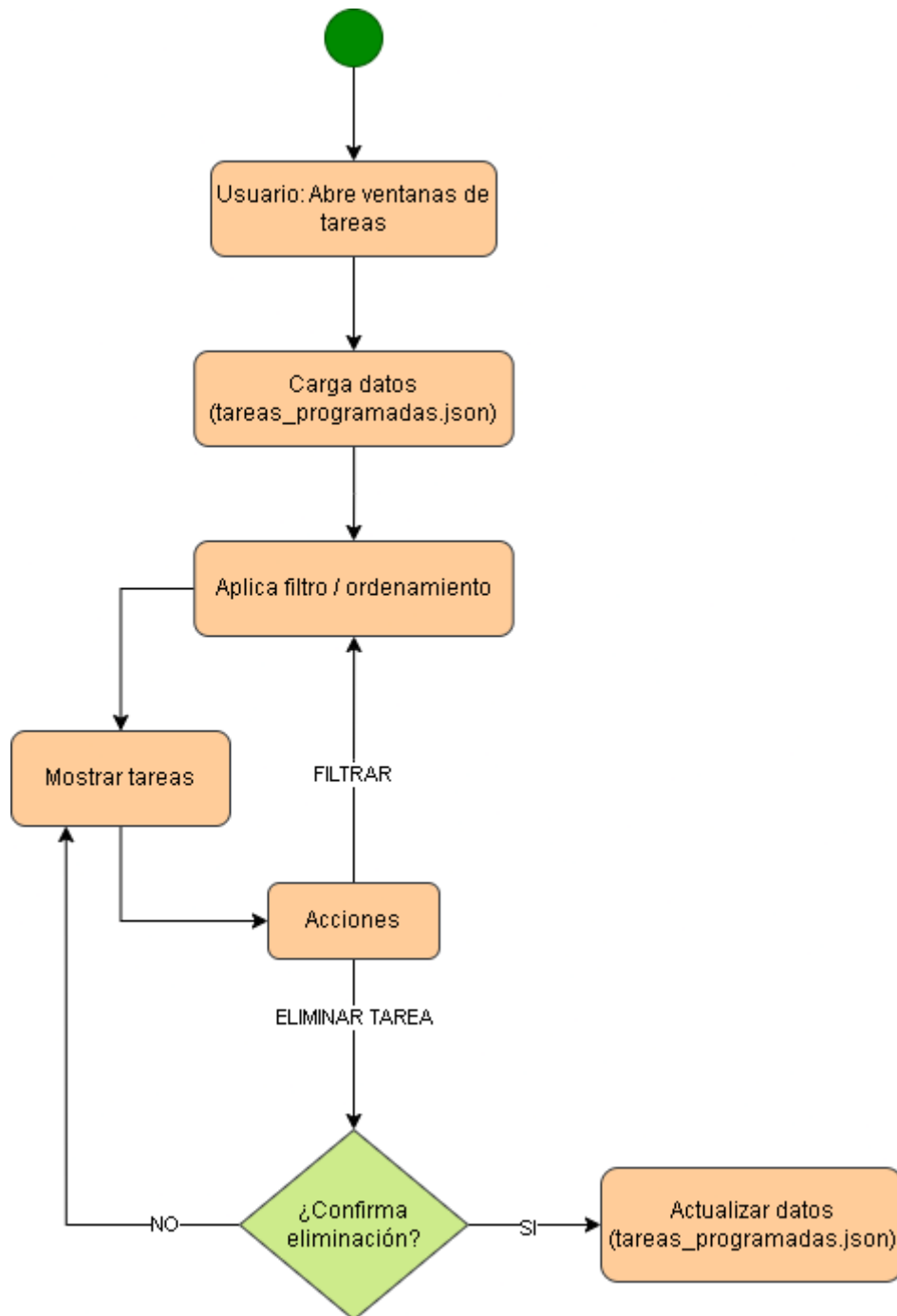


Figura 4: Diagrama de flujo de tareas programadas

b. La interfaz incluye:

- i. Filtrado por ubicación
- ii. Ordenamiento por IP (ascendente/descendente)
- iii. Visualización de próxima ejecución

iv. Eliminación individual con confirmación

Flujo Principal del Sistema

Descripción General:

El sistema sigue un flujo estructurado que garantiza la persistencia de datos, ejecución automática de tareas y una experiencia de usuario coherente. Combina carga inicial de datos, gestión en segundo plano e interacción mediante una interfaz intuitiva.

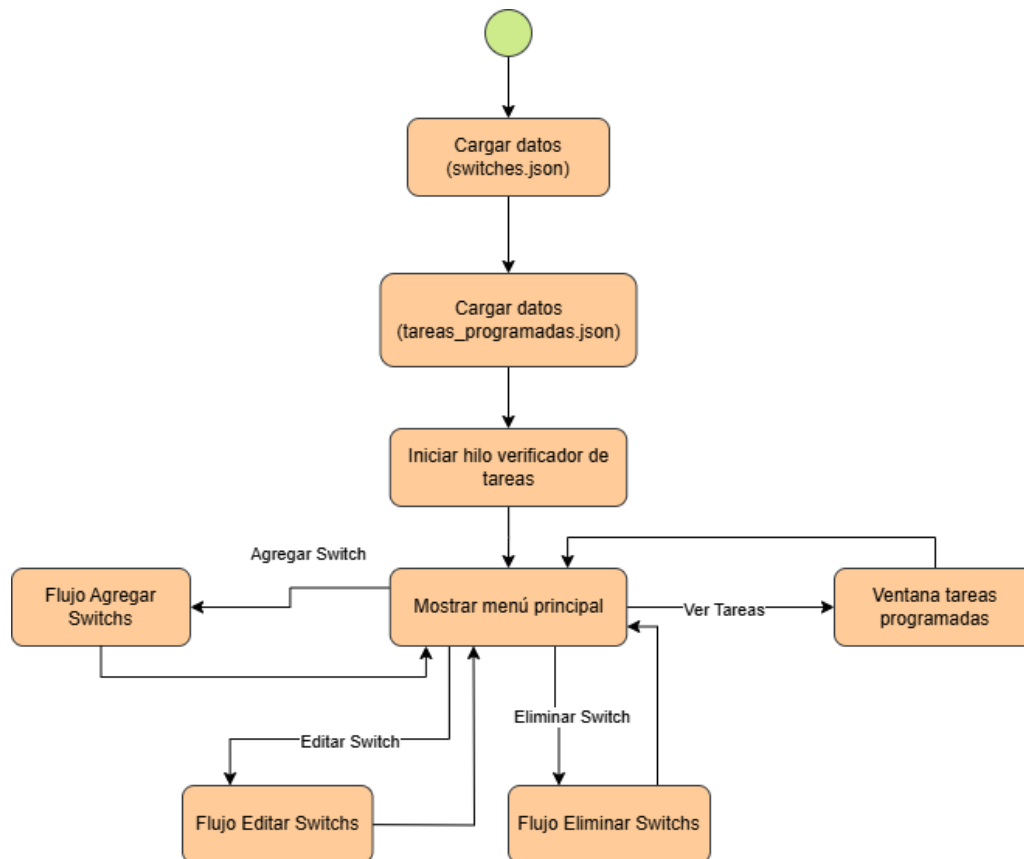


Figura 5: Diagrama de flujo principal del sistema

Desglose de Pasos:

- **Inicio**
 - La aplicación se lanza y prepara el entorno
 - Inicializa componentes esenciales (SSH, FTP, UI)
- **Cargar datos existentes**
 - Lee switches.json → Carga switches preconfigurados
 - Lee tareas_programadas.json → Recupera backups programados
 - Si no existen archivos, crea estructuras vacías
- **Iniciar planificador de tareas**
 - Lanza hilo en segundo plano que cada 30 segundos:

- Verifica tareas pendientes
- Ejecuta backups programados
- Reprograma tareas recurrentes
- Actualiza archivos JSON
- **Interfaz de usuario**
 - Construye ventana principal con:
 - Terminal de salida (derecha)
 - Panel de control (izquierda)
 - Configura tema oscuro y esquema de colores

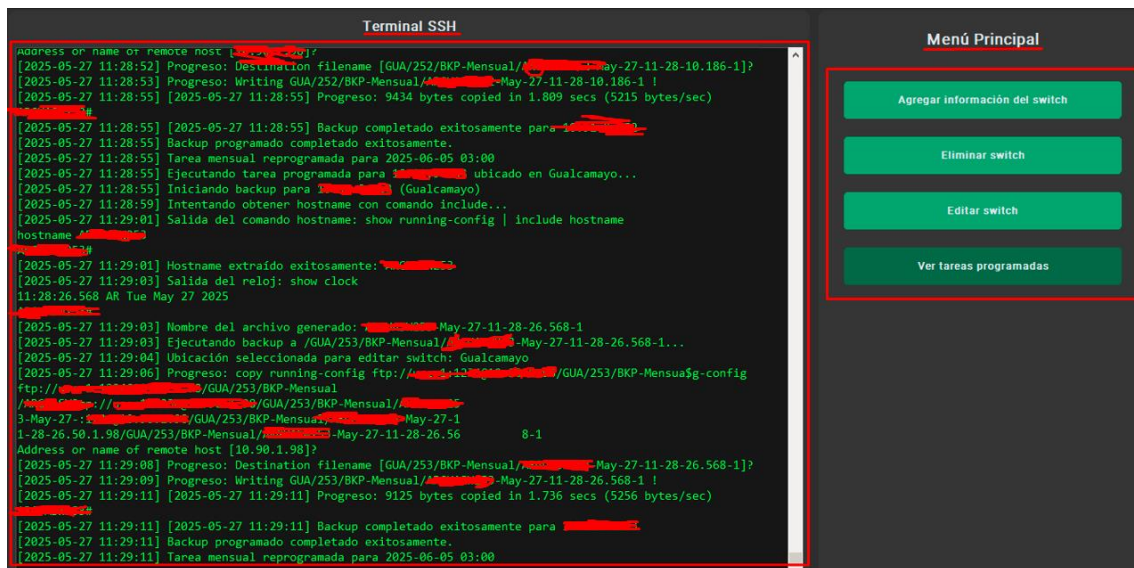


Figura 6: Interfaz de usuario

- **Menú Principal**
 - Ofrece 4 opciones principales

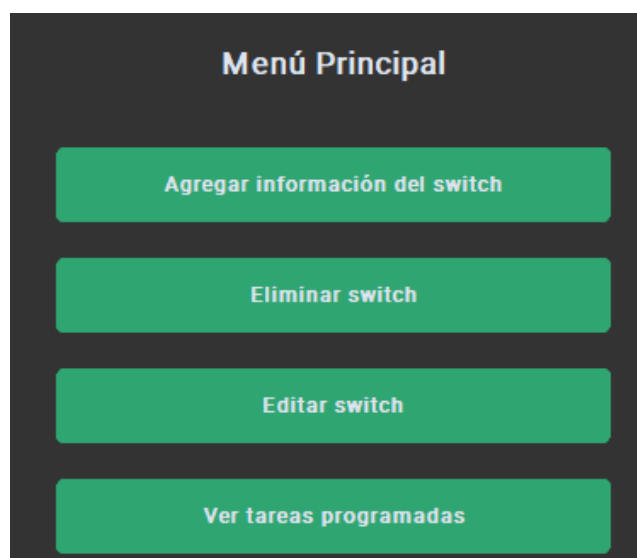


Figura 7: Iterfaz (Menú Principal)

- **Gestión de Switchs**

- Flujos interactivos para:
 - Agregar: Valida SSH/FTP → Guarda → Programa backup
 - Editar: Individual/masivo → Actualiza credenciales
 - Eliminar: Confirmación → Borra switch y tareas asociadas

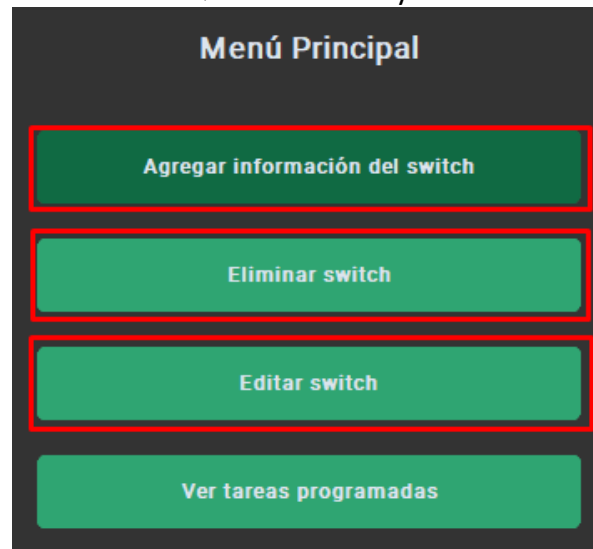


Figura 8: Gestión de Switchs

- **Gestión de tareas**

- Ventana independiente con:
 - Filtrado por ubicación
 - Ordenamiento por IP
 - Eliminación selectiva
 - Visualización de próximas ejecuciones

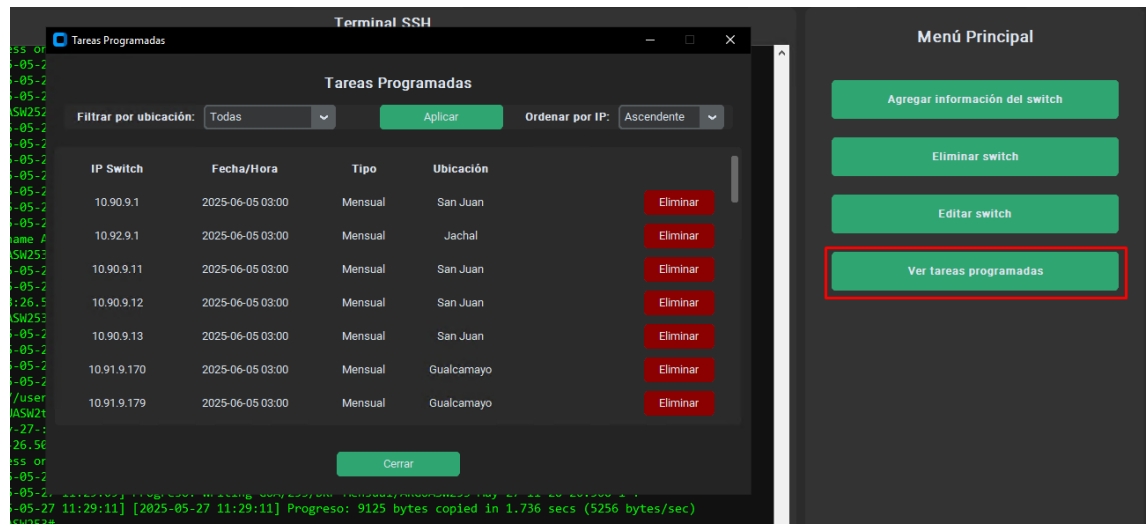


Figura 9: Iterfaz (Gestión de Tareas)

- **Persistencia de datos**

- Guardado inmediato tras cualquier cambio en:
 - switches.json (datos de dispositivos)
 - tareas_programadas.json (cronograma)
- Garantiza consistencia entre sesiones