

**E.T. Nº 36**  
**ALMIRANTE GUILLERMO BROWN**



**Redes**

**IPVision - Documentación**

Año: 5º

División: 1º

Turno: Tarde

Nombre: Alejo

Apellido : Guerra

Profesor: Oscar A. Obregón

Fecha de entrega: 29/10/2025

# IPVision - Documentación:

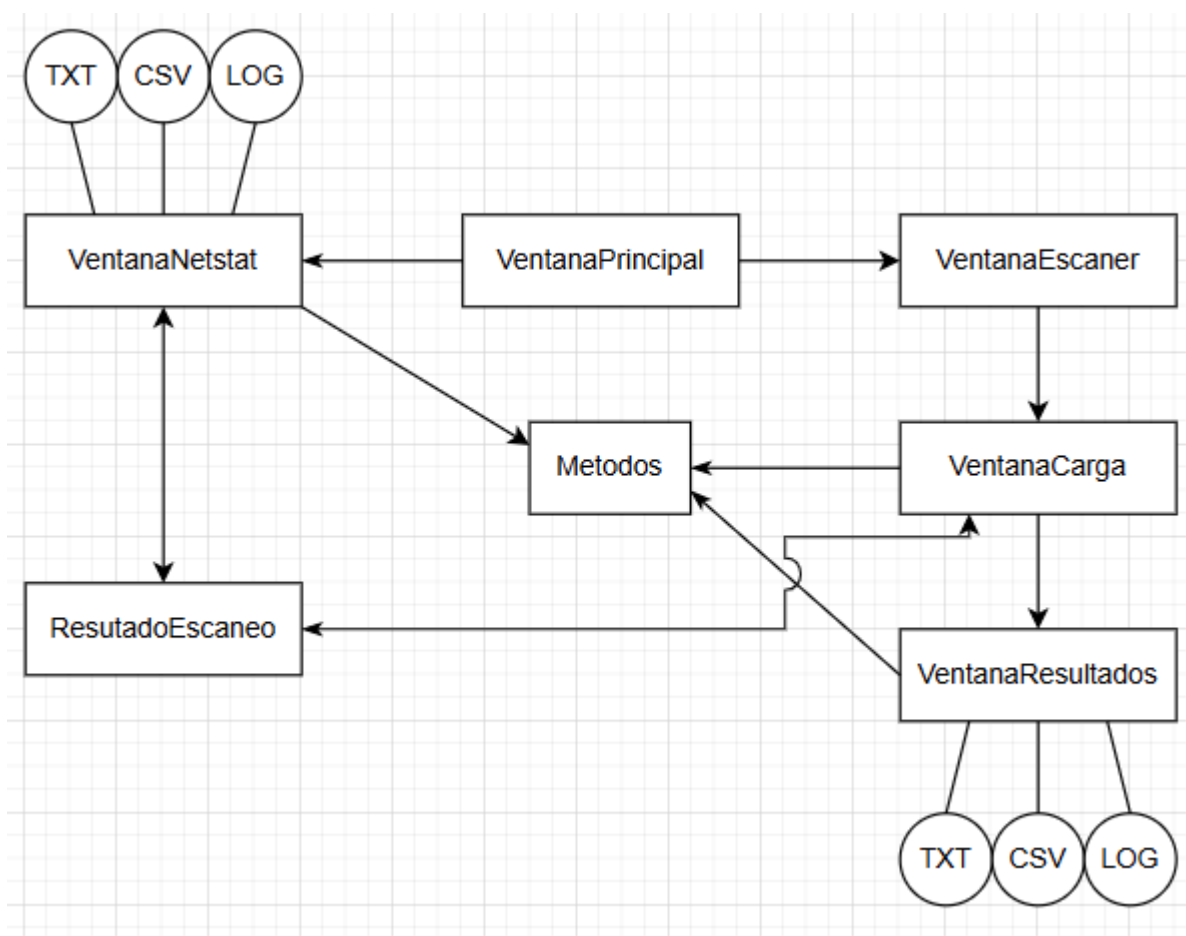
## Descripción del proyecto

IPVision es una herramienta desarrollada para simplificar el análisis de redes y la gestión de conexiones locales. Combina **dos módulos principales**:

1. **Escáner de IP**: Permite explorar rangos de direcciones IP, desde una IP inicial hasta una IP final, mostrando información organizada sobre el estado de cada equipo, tiempo de respuesta, host, servidor DNS y otros datos relevantes.
2. **Ventana Netstat**: Monitoriza las conexiones locales del equipo en tiempo real, mostrando puertos, PIDs, estados de las conexiones y procesos asociados.

El objetivo de IPVision es proporcionar una **solución integral** para monitoreo de redes, detectando equipos no autorizados, problemas de conectividad o conexiones sospechosas, con una interfaz intuitiva y amigable.

*Diagrama del código:*



### *Metodologías del trabajo: ¿Cuáles fueron los métodos utilizados?*

El primer paso consistió en definir claramente el objetivo del proyecto y las especificaciones que debía cumplir. A continuación, diseñé y desarrollé las ventanas y clases necesarias para la interfaz visual. Una vez finalizada la parte gráfica, investigué el funcionamiento del recorrido de direcciones IP y los comandos esenciales para garantizar que el programa operara correctamente.

Para conceptualizar la lógica principal, primero la modelé en Python, dado que me resultaba más sencillo entender y probar el flujo, y posteriormente la trasladé a Java. Con el apoyo de inteligencia artificial y mis conocimientos previos, desarrollé toda la lógica del programa, para luego enfocarme en optimizar la interfaz, el código y la experiencia del usuario.

Durante todo el proceso, se realizaron mejoras continuas en términos de eficiencia, corrección de errores, velocidad de ejecución y estética general del programa.

### *Tecnologías utilizadas: ¿Cuáles son las tecnologías utilizadas? ¿Por qué?*

**javax.swing:** Construcción de la interfaz gráfica (ventanas, paneles, botones, etiquetas, tablas, diálogos, comboboxes, campos de texto).

**java.awt:** Layouts (BorderLayout, GridBagLayout, FlowLayout), manejo de dimensiones, colores, fuentes y cursores.

**java.awt.event / javax.swing.event:** Manejo de eventos de usuario (ActionListener, MouseAdapter, DocumentListener).

**java.io / java.nio.charset:** Lectura y escritura de archivos (TXT, CSV, LOG), codificación UTF-8.

**java.util:** Colecciones, utilidades de fecha y hora, comparadores para ordenamiento.

**java.text.SimpleDateFormat:** Formateo de fechas y horas en reportes.

**java.net.InetAddress / ProcessBuilder:** Ping, resolución de host y ejecución de comandos externos (`nslookup`, `netstat`).

**java.util.concurrent:** `ExecutorService`, `Executors` y `SwingWorker` para multihilo y concurrencia en UI.

**javax.swing.filechooser.FileNameExtensionFilter:** Filtrado de archivos en diálogos de guardado.

**javax.swing.border:** Bordes estéticos y `TitledBorder`.

**DefaultTableCellRenderer:** Para colorear filas según estado en Ventana Netstat.

**JPopupMenu / JTabbedPane:** Funcionalidades adicionales en la interfaz de Netstat (copiar fila, estadísticas).

## Otras librerías de utilidad

- **javax.swing.filechooser.FileNameExtensionFilter** para filtrar tipos de archivo en diálogos de guardar (TXT, CSV, LOG).
- **javax.swing.border** para bordes estéticos y TitledBorder en paneles.

## A medida que iba haciendo el proyecto, surgieron estos problemas:

- *Problemas con exportación en excel:* Estuve mucho tiempo tratando de que se exportara directamente a formato excel, porque tenía que importar un montón de módulos manualmente o pasar a proyecto Maven. Esto se solucionó cambiando el tipo de archivo a CSV, que era mucho más simple.
- *Problemas con los componentes de JSwing:* Muchos componentes no se adaptaban bien a la interfaz que planteaba, por lo que tuve que estar mucho tiempo para lograr que queden correctamente.
- *Falta de conocimiento:* Java es demasiado extenso, y tiene muchas funcionalidades. Pasé mucho tiempo entendiendo todo lo que yo pensaba añadir a mi proyecto, además del funcionamiento de las IPs y los comandos de consola
- *MultiThreading :* Buscando acelerar mi proyecto, recurrí a usar hilos, esto hizo que muchas funciones de mi programa en ese momento dejaran de funcionar correctamente. Hacer el traslado costó mucho tiempo.
- *Netstat y parsing de datos:* Adaptar la salida del comando a diferentes sistemas fue complejo.

## Escalabilidad: ¿Qué se podría mejorar en el futuro?

El programa está hecho con la idea de poder ampliarse a futuro, por lo que tiene una base que en caso de querer expandir no se tendrá que modificar ni rediseñar.

**Disponibilidad para todas las versiones de Java:** Mi programa se desarrolló en java 1.8, por lo que muchas funciones no funcionan en versiones más recientes, y si necesitas usarlo, tendrías que adaptarte a la versión.

## Internacionalización y adaptabilidad

Hoy el programa está en español, pero podría ser adaptado para varios idiomas, lo que permitiría su uso en entornos más amplios. También se podrían implementar configuraciones regionales de formato de fechas y red.

## Documentación y mantenimiento

Aunque el código está organizado, agregar comentarios más detallados facilitaría futuras modificaciones, la incorporación de nuevos desarrolladores al proyecto y la resolución de errores.