Proyecto de: Esteban Gonzalez

Materia: Redes

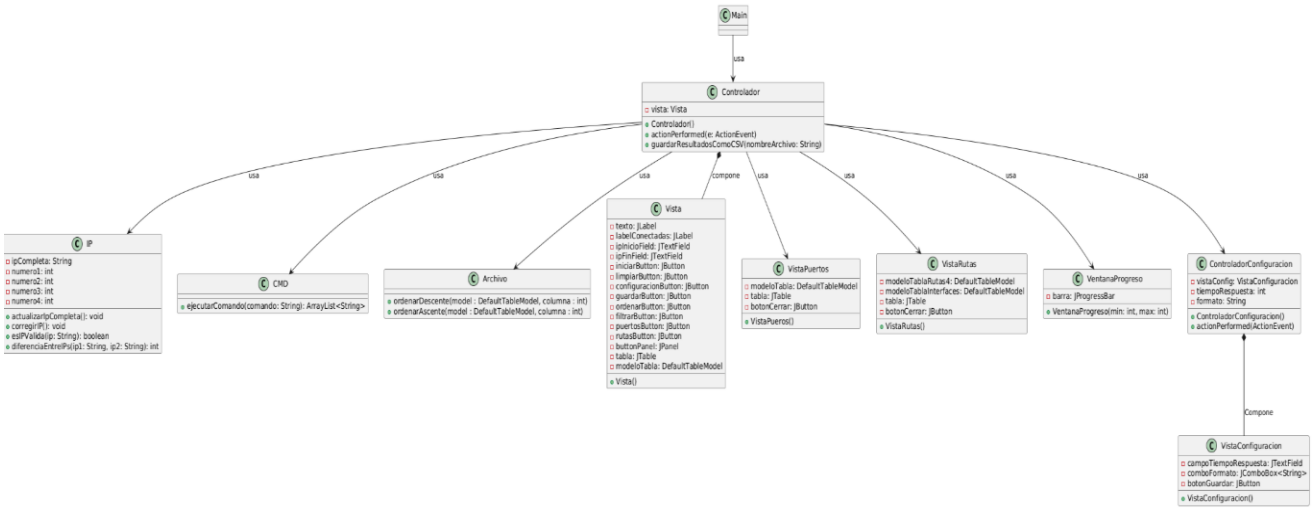
Profesor: Oscar Obregon

Escaner de red

Acerca del programa:

El programa es un escáner de red, que dadas dos IPs se encargará de analizar las IPs que estén dentro de ese rango, obteniendo su nombre, si se encuentra conectada o no y cuánto tardó en responder, guardando estos resultados y mostrandolos en una tabla. Luego es usuario puede decidir guardar el resultado en un archivo. También el usuario puede configurar ciertos aspectos del proceso como cuánto espera como máximo al hacer solicitudes a las direcciones y en qué tipo de archivo guarda los resultados.

Diagrama del sistema:



Representación escrita del diagrama

```
class IP {
    -ipCompleta: String
    -numero1: int
    -numero2: int
    -numero3: int
    -numero4: int

    +actualizarIpCompleta(): void
    +corregirIP(): void
    +esIPValida(ip: String): boolean
    +diferenciaEntreIPs(ip1: String, ip2: String): int
}

class CMD {
    +ejecutarComando(comando: String): ArrayList<String>
}
```

```
class Archivo {
    +ordenarDescente(model : DefaultTableModel, columna : int)
    +ordenarAscente(model : DefaultTableModel, columna : int)
}

class Vista {
    -texto: JLabel
    -labelConectadas: JLabel
    -ipInicioField: JTextField
    -ipFinField: JTextField
    -iniciarButton: JButton
    -limpiarButton: JButton
    -configuracionButton: JButton
    -guardarButton: JButton
    -ordenarButton: JButton
    -filtrarButton: JButton
    -puertosButton: JButton
    -rutasButton: JButton
    -buttonPanel: JPanel
    -tabla: JTable
    -modeloTabla: DefaultTableModel

    +Vista()
}

class VistaPuertos{
    -modeloTabla: DefaultTableModel
    -tabla: JTable
    -botonCerrar: JButton

    +VistaPueros()
}

class VistaRutas{
    -modeloTablaRutas4: DefaultTableModel
    -modeloTablaInterfaces: DefaultTableModel
    -tabla: JTable
    -botonCerrar: JButton

    +VistaRutas()
}

class VistaConfiguracion {
    -campoTiempoRespuesta: JTextField
    -comboFormato: JComboBox<String>
    -botonGuardar: JButton

    +VistaConfiguracion()
}

class VentanaProgreso {
```

```
    -barra: JProgressBar

    +VentanaProgreso(min: int, max: int)
}

class ControladorConfiguracion {
    -vistaConfig: VistaConfiguracion
    -tiempoRespuesta: int
    -formato: String

    +ControladorConfiguracion()
    +actionPerformed(ActionEvent)
}

class Controlador {
    -vista: Vista

    +Controlador()
    +actionPerformed(e: ActionEvent)
    +guardarResultadosComoCSV(nombreArchivo: String)
}

Main --> Controlador : usa
Controlador --> CMD : usa
Controlador --> Archivo : usa
Controlador --> IP : usa
Controlador --> VentanaProgreso : usa
Controlador --> VistaRutas : usa
Controlador --> VistaPuertos : usa
Controlador *-- Vista : compone
Controlador --> ControladorConfiguracion : usa
ControladorConfiguracion *-- VistaConfiguracion : Compone
```

Métodos usados:

Los métodos que usé para hacer el programa fueron la programación orientada a objetos, ya que me sirvió para organizar el código y separar las diferentes funcionalidades, como la parte gráfica de la lógica. También use lo llamada vibe coding ya que utilicé la ia saber como hacer cosas que no sabía (cómo ciertas cosas con hilos y la ejecución de comandos cmd) y para acelerar ciertas cosas manuales.

Tecnologías utilizadas:

-El lenguaje utilizado fue Java. Decidí usarlo ya que contaba con cierto conocimiento básico sobre este, además como está fuertemente orientado a objetos era una buena opción.
-También use librerías internas para el apartado gráfico, la implementación de hilos y ciertos tipos de datos.

Problemas durante el desarrollo:

Los problemas que surgieron fueron:

- La barra de progreso y tabla de resultados no se actualizaban si no hasta el final. Para poder arreglar esto tuve que implementar un hilo que se encargue por separado de esto.
- Los datos de la configuración se guardaban antes de ser ingresados por el usuario, lo que los volvía erróneos. Para evitar esto VistaConfiguración cambio de JFrame a JDialog.

Mejoras a futuro:

Posibles mejoras que podría recibir el programa a futuro son:

- Optimización, el programa es bastante lento incluso cuando trabaja con un rango chico. Además podría mejorar el uso de hilos para lograr esto.
- Mejora visual, el apartado gráfico funciona, pero podría mejorarse para que sea más lindo y cómodo de usar.
- Adaptar el programa a múltiples sistemas operativos, ya que actualmente solo funciona en windows.