E.T. Nº 36
ALMIRANTE GUILLERMO BROWN

Redes

Actividad 14 - Trabajo Práctico: Escáner de Red
Documentación del desarrollo

Año: 5º	División: 1º	Turno: Tarde
Alumno: Santino, Martinez		
Docente: Oscar, Obregón		
Fecha de entrega: 18/08/2025		

Para qué sirve el programa

El programa “IPScanner” fue desarrollado para escanear un rango de direcciones **IPv4** de manera simple, mostrando información útil de cada IP, como:

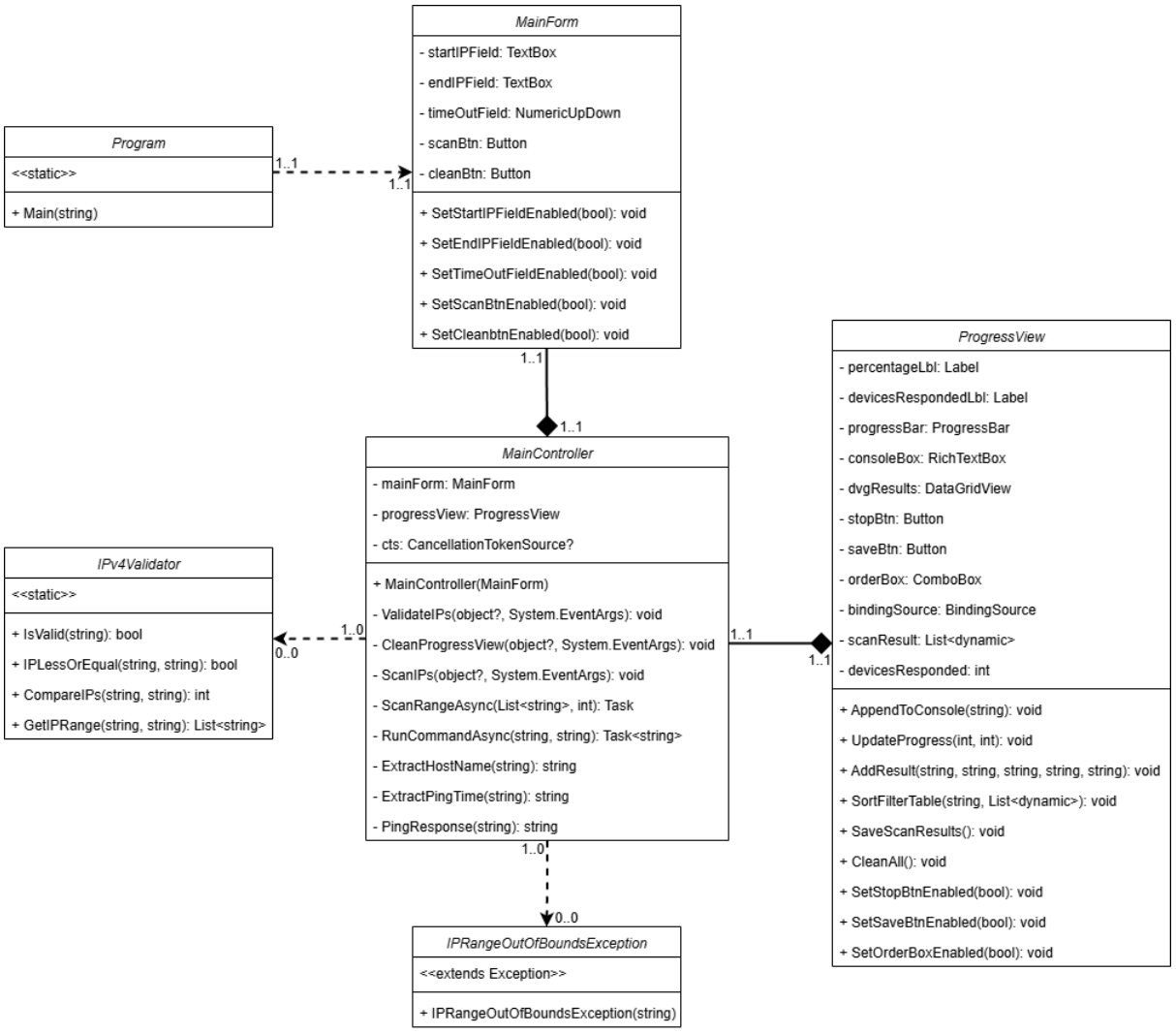
- Respuesta al **ping** (tiempo de ida y vuelta, estado del dispositivo).
- Resolución de nombre de host mediante **nslookup**.
- Respuesta completa de ambos comandos.

Además, la aplicación presenta una interfaz gráfica sencilla que permite al usuario:

- Ingresar un rango de IPs a escanear.
- Visualizar los resultados en una tabla.
- Seguir el progreso del escaneo con las respuestas de los comandos, y una barra de progreso.

Cómo está armado el sistema

El sistema sigue el modelo vista controlador para lograr independencia entre la lógica del programa y la vista del usuario. Además, se usa orientación a objetos, y el UML se vería de esta manera:



Qué métodos se usaron y por qué

Al ver que el trabajo práctico solicitaba un programa con una ventana para que el usuario interactúe, se pensó en usar el modelo vista controlador (aprendido de la clase de Programación Orientada a Objetos), el cual asegura una independencia entre la parte lógica y visual del programa, además de mejorar la legibilidad y orden del código. También el trabajo pedía utilizar programación orientada a objetos, por ende, el programa está completamente hecho en este paradigma.

Nunca se pensó en un diseño previo, todo fue pensado mientras se programaba en el momento, y para ello, se utilizó IA para hacer las bases de las vistas. Además, los contenidos de los que no se sabían con certeza, se preguntaban a una IA para acelerar el proceso de búsqueda.

Qué tecnologías se eligieron y por qué

El trabajo práctico daba a elegir al estudiante entre dos lenguajes de programación para hacer el programa, Java y C#. Se decidió programar en C#, debido a que se quería sumar experiencia de un lenguaje no utilizado anteriormente.

Se utilizaron namespaces del BCL(Base Class Library). El BCL tiene colecciones de clases, interfaces y tipos que proporciona .NET. Los namespaces utilizados para el programa fueron:

- System.Diagnostics
- System.Text

Qué problemas aparecieron y cómo se solucionaron

Uno de los problemas principales que hubo fue la sintaxis de C#, ya que no se tenía experiencia alguna con este lenguaje. Aun así, este fue el problema más fácil de solucionar, porque al entender la lógica de programación, lo único que se tuvo que hacer fue buscar cómo se usaban las herramientas del lenguaje, además de que este es muy parecido a Java.
Otro problema fueron los procesos asincrónicos, tareas, delegados, e hilos, debido a que no aprendimos nada sobre estos en la escuela, pero por suerte ya se contaba con algo de conocimiento que fue de gran ayuda para saber en donde investigar.

No hubieron muchos errores en compilación, porque el paradigma y el modelo utilizados lo tenemos muy presente en la escuela, y entonces era fácil adaptarlo al C#.

Qué se podría mejorar en el futuro

Lo que se podría mejorar es:

- Agregar un botón de reanudar el escaneo.
- Agregar la posibilidad de guardar en un archivo el resultado de varios escaneos.
- Agregar la posibilidad de guardar en un archivo el resultado de un escaneo con un criterio de orden o filtro.
- Permitir al usuario asignar parámetros a los comandos “ping” y “nslookup” utilizados por el programa.
- Unir los componentes de ambas ventanas en una sola.
- Adaptar el programa para varios sistemas operativos.
- Permitir ingresar en el inicio y fin del rango un nombre de host, el cual el programa tendrá que investigar cual es su IP, y así armar el rango de IPs.
- Permitir al usuario insertar un tiempo máximo de espera para cada comando ejecutado.
- Añadir más tipos de archivos a la hora de guardar los resultados.
- Permitir usar algunas expresiones regulares para hacer rangos de IPs más dinámicos.
- Permitir hacer “saltos” en los rangos de IPs para hacerlos más dinámicos.