**Analizador de paquetes de red**

Se tratará de un analizador de paquetes de red, que captura y categoriza lo analizado desde una interfaz de red seleccionada. para luego reflejarlo en una base de datos.

**Descripción**

Esta será una guía para la instalación y ejecución de código en lenguaje Python a través de VScode, el cual captura y analiza paquetes de tráfico de red a nivel local desde una interfaz de red especificada en el comando, Para Windows utilizaremos Ethernet, para SO Mac utilizaremos en0 o en1, dándonos estadísticas de los siguientes parámetros:

* IP de destino con mayor tráfico
* IP de origen con mayor trafico
* Total de paquetes capturados
* Numero de paquetes por protocolo

Estos datos se reflejarán en una base de datos creada a partir de Mysql, que en esta demostración trabajara con Workbench.

**Requisitos**

Para que este código funcione en Windows, se requiere:

**1. Python 3.8 o superior**

* Instalar Python desde https://www.python.org/downloads/windows/
* Durante la instalación, marca la opción para agregar Python a las variables del sistema; para ejecutar Python desde cualquier lugar en la línea de comandos.

**2. Librerías de Python**

Este script usa algunas librerías específicas. Instálalas con este comando en tu terminal:

pip install scapy mysql-connector-python argparse

* **Scapy**: para capturar y manejar paquetes de red.
* **mysql-connector-python**: para conectarse y guardar datos en MySQL.
* **argparse**: para gestionar argumentos que le das al script desde la línea de comandos.

**Compatibilidad de Scapy en windows**: es recomendable tener instalado **Npcap** para evitar errores en controladores de red desde : https://npcap.com/dist/npcap-1.80.exe

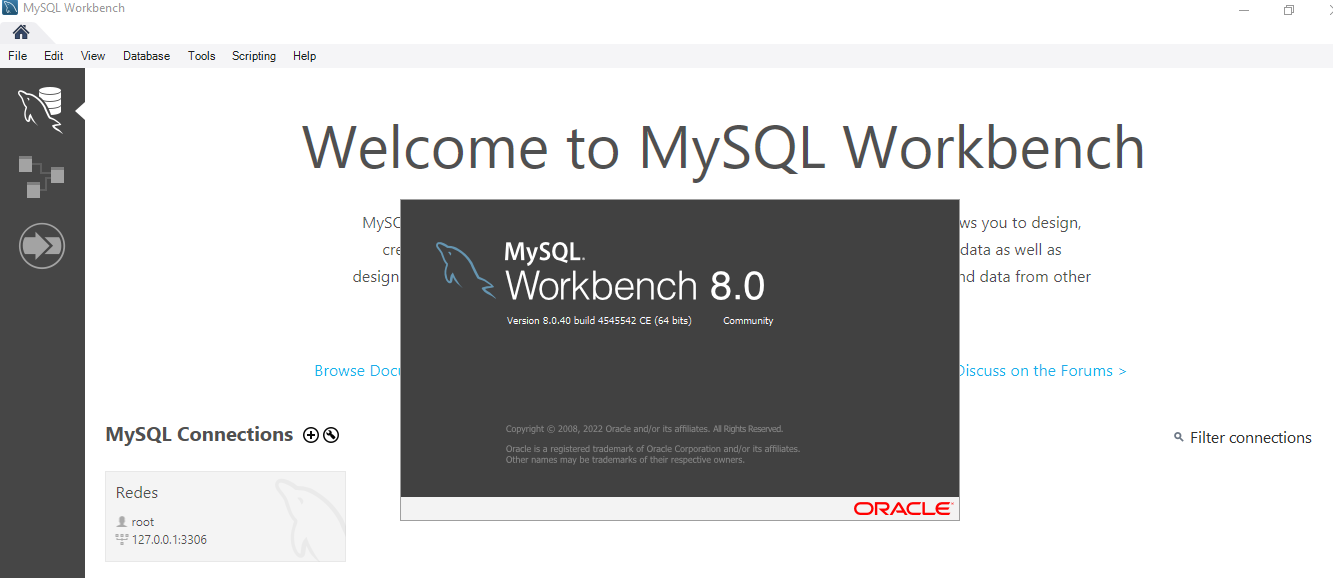
**3. MySQL Server**

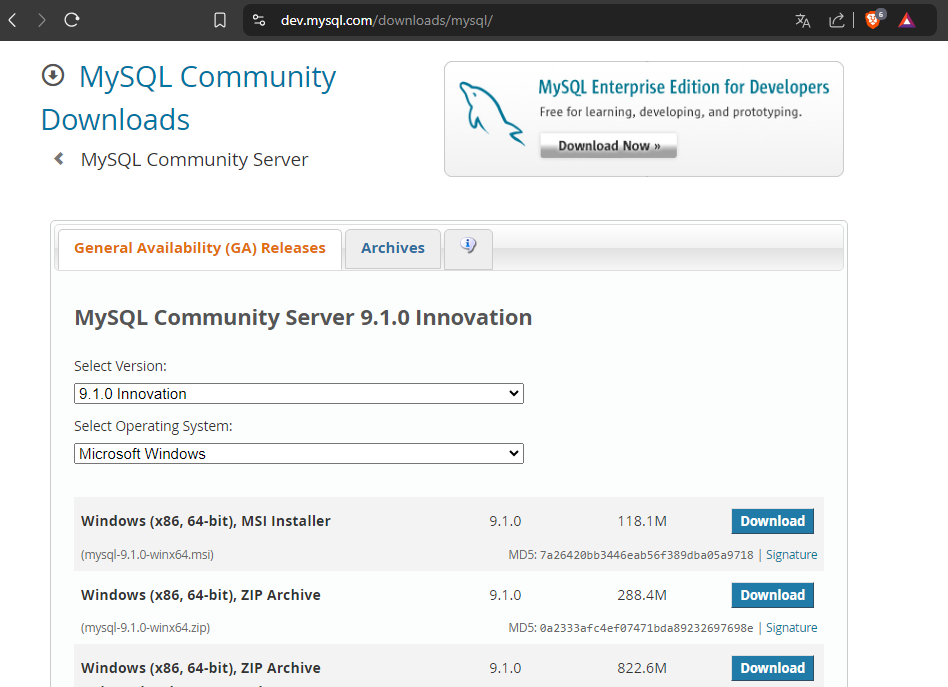
Necesitas tener MySQL instalado y en ejecución

* Crear una base de datos llamada trafficanalysis y asegurase de que esté escuchando en el puerto 3306 (o cambiar el puerto en el código si se utiliza otro).

**Empezando**

Para comenzar debemos tener instalado: Workbench 8.0 (en esta demostración utilizaremos este cliente) o posteriores versiones

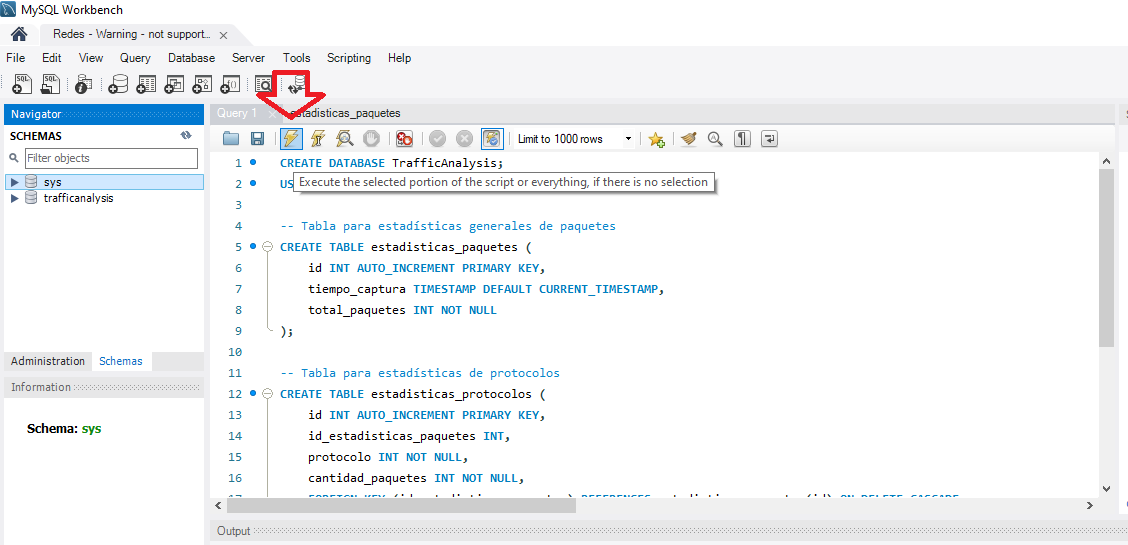


Para mysql server utilizaremos Mysql server 9.1 <https://dev.mysql.com/downloads/mysql/> podremos elegirlo tanto para Windows o para Mac

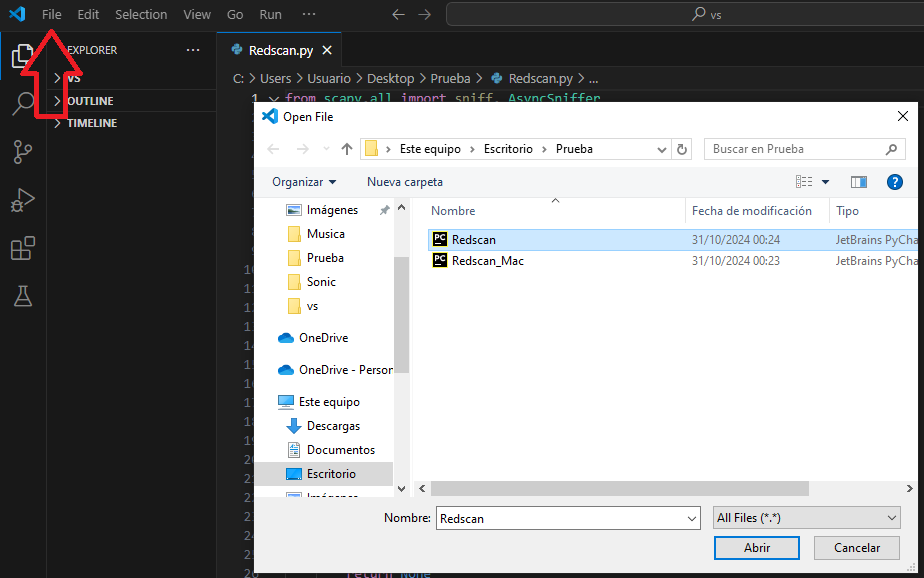
Una vez instalados procederemos a crear la base de datos

Con el código adjunto llamado DATABASE.sql Configuraremos la base de datos en nuestro cliente SQL.

Abriremos Workbench, crearemos la base de datos llamada Redes y en una query pegaremos lo contenido en el archivo y ejecutaremos. Lo que nos creara las tablas necesarias que reflejaran los datos que guardaremos al ejecutar nuestro código:



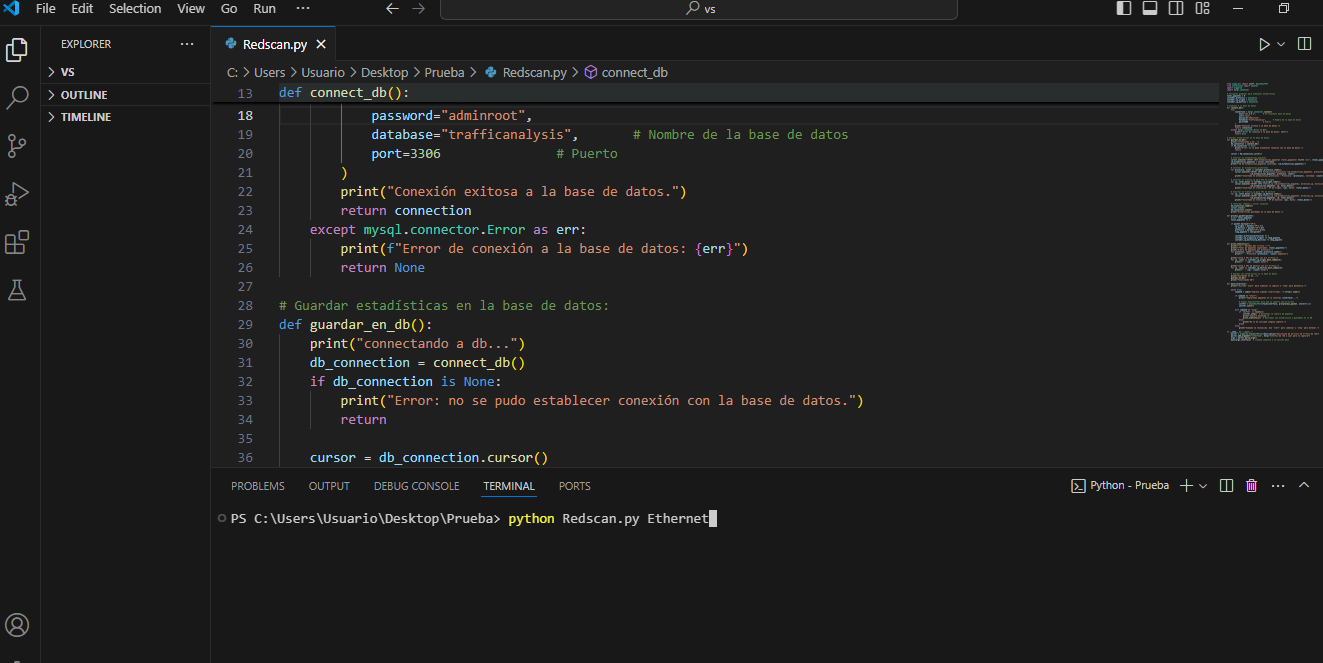
Una vez creada la base, abriremos Visual Studio Code, para este paso necesitamos descargar el archivo que contiene el código a ejecutar, con Vscode iremos a File, open file y seleccionaremos la ruta del archivo descargado,( dependiendo nuestro sistema operativo tomaremos el código de Windows o Mac.) En este caso tomaremos el de Windows:



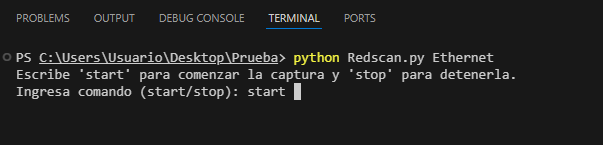
Una vez tengamos nuestro código abierto, procederemos a abrir una terminal dentro del vscode, nos posicionaremos en la carpeta donde hayamos descargado el archivo y con las librerías mencionadas anteriormente instaladas corremos el comando para que nuestro código se ejecute:

**Python Redscan.py Ethernet**

Python (nombre del archivo) (interfaz a escanear)

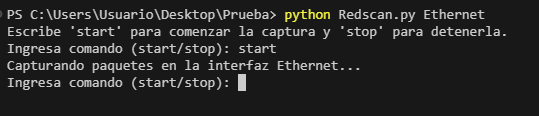


Comienza la ejecución:



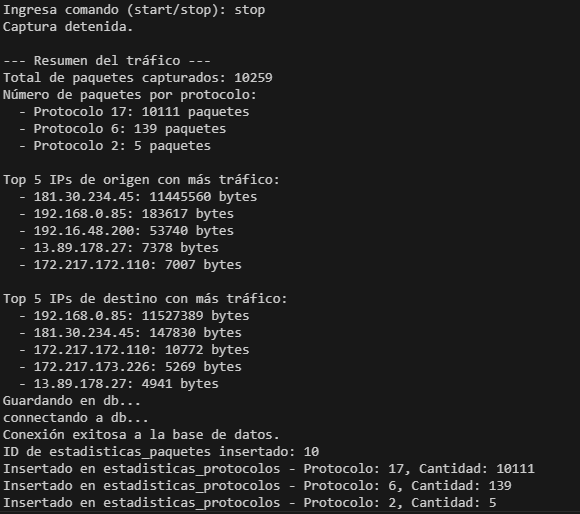
Con el comando: start

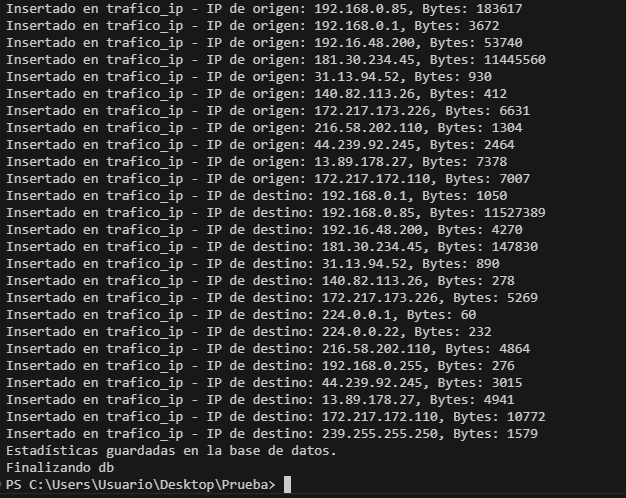
Comenzara la lectura de paquetes:



Con el comando: stop

Terminara la captura y nos devolverá los datos de los paquetes capturados:

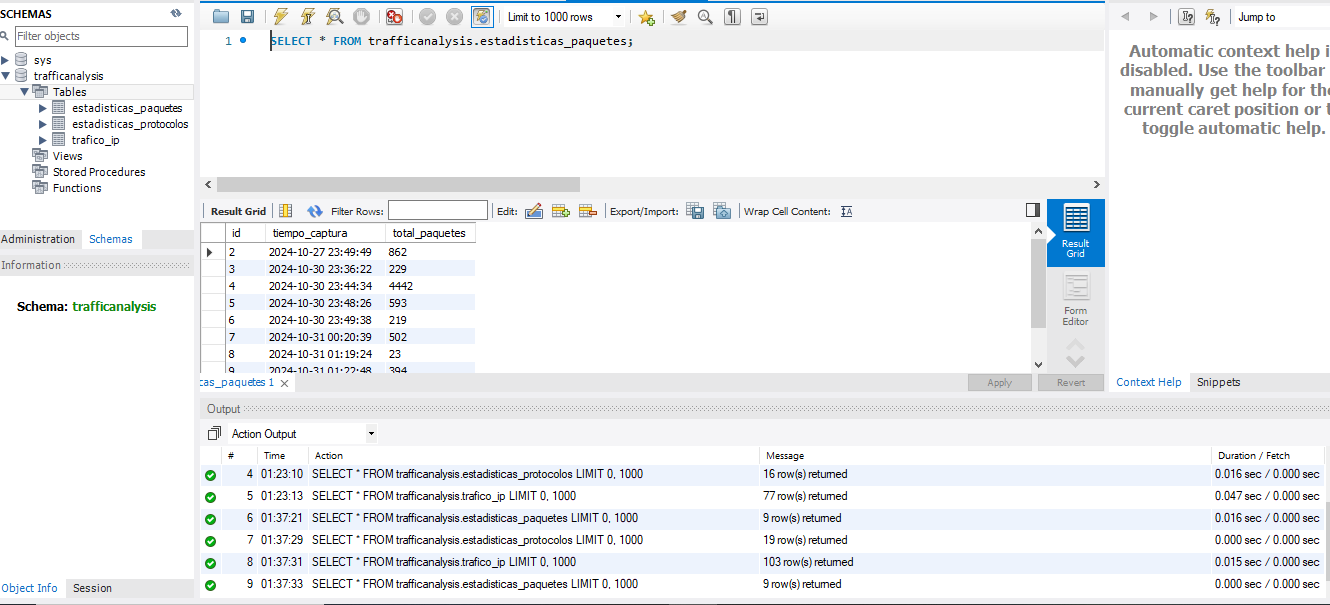




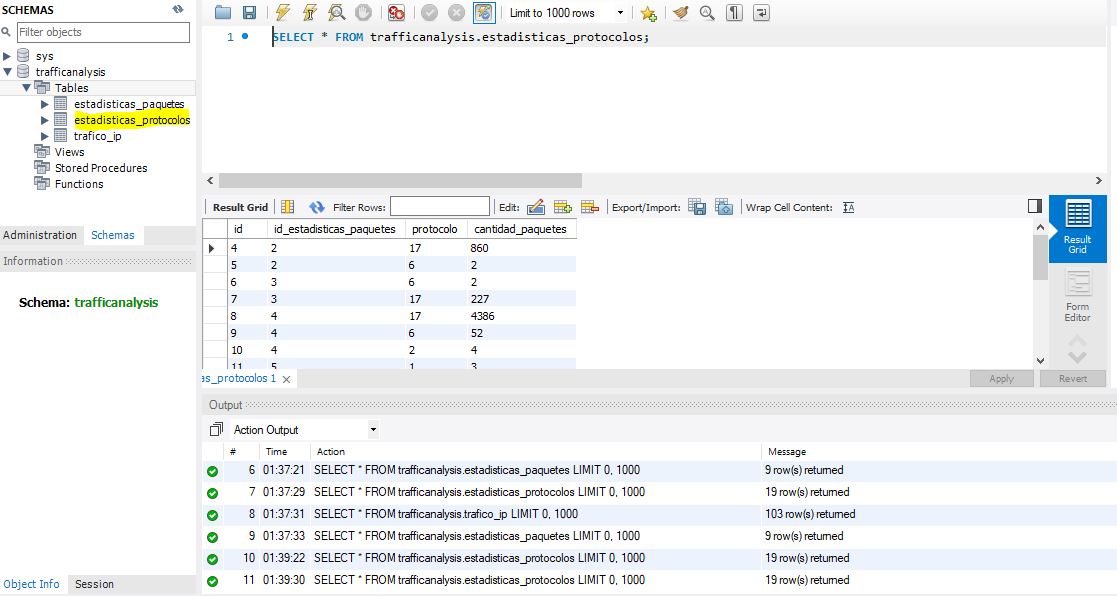
**Validando los datos en la Base de datos ⚙️**

Abriremos Workbench e ingresaremos en la base de datos, para corroborar los datos en las tablas correspondientes:

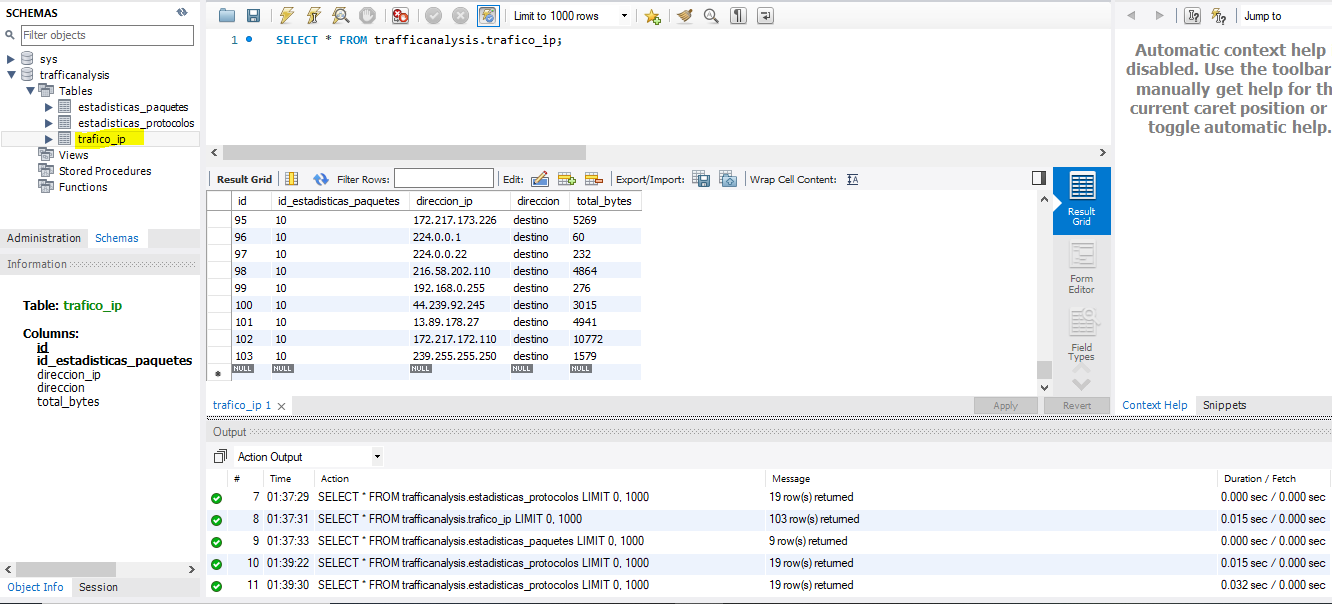
Estadística paquetes:



Estadisticas protocolo:

****

Trafico IP:



**Construido Con 🛠️**

Explica qué tecnologías usaste para construir este proyecto. Aquí algunos ejemplos:

* [Ruby](https://www.ruby-lang.org/es/) - El lenguaje utilizado
* [Ruby on Rails](https://rubyonrails.org/) - El framework web utilizado
* [Ruby gems](https://rubygems.org/) - Gestión de dependencias
* [Postgresql](https://www.postgresql.org/) - Sistema de base de datos
* [Bulma IO](https://bulma.io/) - Framework de CSS

Librería scapy, Python, mysql