

**UNIVERSIDAD NACIONAL DE ENTRE RÍOS**

**FACULTAD DE INGENIERÍA**

**Tecnicatura Universitaria en Procesamiento y Explotación de datos**

**Trabajo Integrador**

**“Análisis de archivos log”**

**Salim Taleb, Nasim A.**

**Oro Verde, 2023**

## Introducción

Los archivos log son un registro de eventos que tuvieron lugar en un cierto tiempo y contienen metadata contextualizando, los datos que contienen puede, ser de formatos estructurados, semiestructurados y desestructurados[1], lo cual puede generar problemas, ya que es difícil y tedioso extraer información de archivos log si los archivos no poseen un formato adecuado y no se posee conocimiento acerca de su manipulación[2], [3].

Actualmente, hay múltiples opciones para examinar archivos log, como hacerlo directamente por consola[3], utilizando software web[4] o software de escritorio[5]. También es posible un análisis más personalizado mediante la utilización de software propio, creándolo con lenguajes como Python[6] o R[7], que poseen librerías capaces de facilitar estas tareas.

En este trabajo se optó por usar gráficas sencillas con las librerías de PyQt5[8] y PyQt graph[9], ya que se utilizó esta misma para la interfaz del software y como un acercamiento a su utilización para futuros trabajos, puesto que es una de las librerías de mayor uso para la creación de interfaces de usuario en Python[10], [11].

## Objetivo general

* Obtener información útil a partir de archivos log.

## Objetivos específicos

* Identificar las variables de las que se pueden obtener métricas y datos cualitativos.
* Elaborar código reutilizable para procesar al menos 3 de esas variables.
* Diseñar un dashboard para representarlas.

## Métodos y técnicas

### Entorno de desarrollo

Como entorno de desarrollo se optó por Microsoft Visual Studio Code, principalmente por permitir el uso de la extensión de Copilot y Copilot Labs, que son herramientas potenciadas por Inteligencia Artificial que ayudan en gran medida en la eficiencia a la hora de codear y también a la corrección de bugs.

### Lenguaje

El lenguaje utilizado es Python, es uno de los lenguajes más utilizados y preferidos por la comunidad[12], además posee librerías que facilitan el trabajo en cuestión.

#### Librerías

PyQt5: Cómo librería principal para las interfaces gráficas y creación de algunas gráficas

PyQtGraph: Librería auxiliar para la creación de algunas gráficas de manera más simple.

Qt designer: Software externo que permite diseñar mediante una interfaz de usuario a las propias interfaces de usuario del software objetivo, también permite exportar estas interfaces en formato .ui.

Pyuic5: Librería que convierte a las interfaces .ui provistas por Qt designer en archivos de código .py que permiten ser utilizados inmediatamente en la codificación de la aplicación.

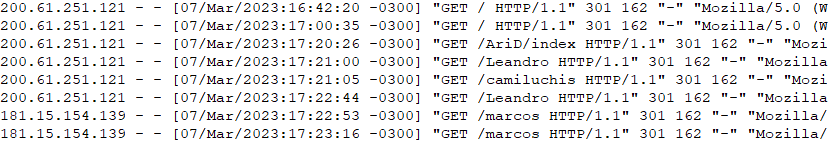
Counter: Esta librería permite realizar conteos de strings que fueron utilizados para la creación de las gráficas, su funcionamiento es similar a la de un diccionario y está basado en ellos.

Re: Para la utilización de expresiones regulares durante el parseo.

## Desarrollo

### Análisis de variables

Observando algunas líneas del archivo log se pueden identificar los datos que se encuentran en él:



En primer lugar, se puede ver la IP de quien se conecta, luego la fecha y hora del acceso, el método, la ubicación, el protocolo, el código de respuesta, el tamaño del archivo y luego información adicional sobre el navegador, entre otras cosas.

De estos datos se quitaron algunos por ser irrelevantes para el análisis, como por ejemplo el protocolo que se mantiene constante, quedando finalmente los datos de IP, fecha y hora, método, directorio y los relacionados con el navegador.

De estos se decidió analizar para este trabajo solamente los tres primeros, dejando lugar en la interfaz para futuras ampliaciones. Para el caso de la IP se optó por mostrar las primeras 10 IP que más ocurrencias se encuentre en el archivo, las ocurrencias y detalles de esta IP mediante una solicitud con una API. En cuanto a la fecha y hora, se elaboraron dos gráficas que muestran las ocurrencias por hora y por día del mes, de tal manera que permita ver horas y días picos de tráfico. Finalmente, en caso del método se creó un gráfico de torta que representé que proporción de solicitudes pertenecen a cada método.

### Codificación

Los archivos utilizados se encuentran anexados a este trabajo, consta de 3 archivos con interfaces, el menú principal que da acceso a los otros dos, el de graficación y el de parseo, a su vez también un módulo para la creación de cada gráfico según cada variable y uno más para la función de parseo. En el directorio también se cuenta el archivo log utilizado para probar el software.

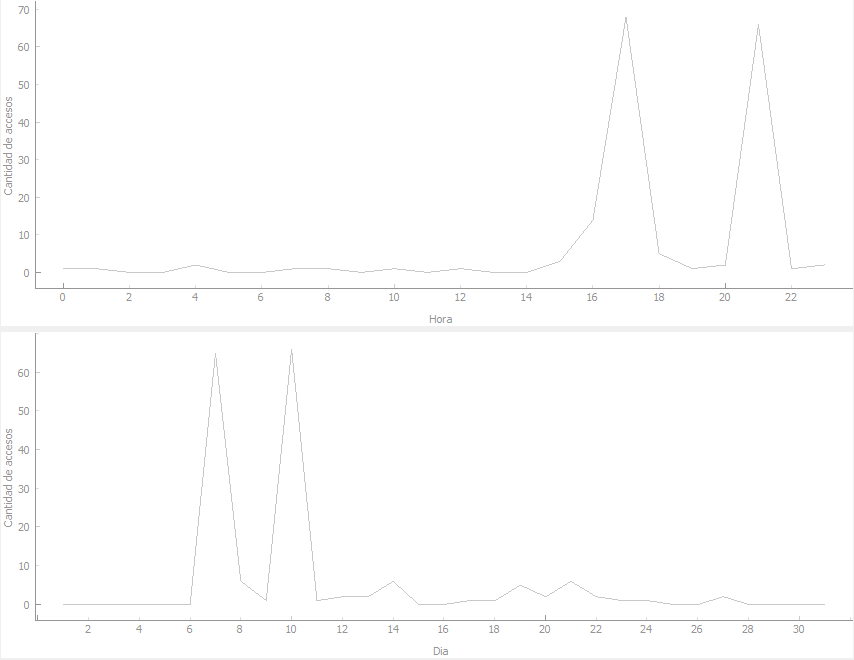
Para el proceso de diseño de las interfaces se utilizó principalmente Qt designer y se realizaron modificaciones mínimas luego al código, como el agregado de los comportamientos de los botones y las llamadas a funciones, en cuanto a las funcionalidades se usó en gran parte Copilot y la Bing Chat.

### Resultados

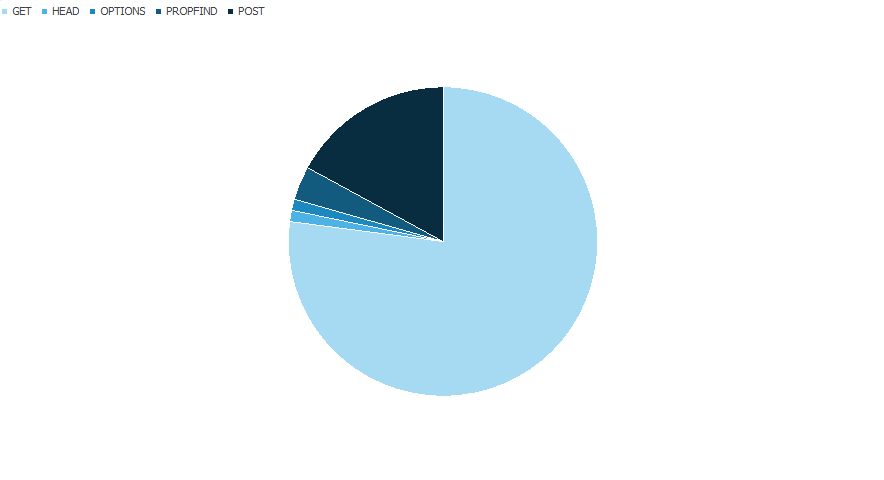
Desplegando los gráficos generados por el log de prueba se observa lo siguiente:



Se ve que las IP de mayor tráfico son de Estados Unidos y Europa, probablemente bots por presentar una gran cantidad de visitas, luego hay IP de Sudamérica que indicaría serían los verdaderos usuarios con una cantidad de accesos moderada.



En estas gráficas se puede apreciar que los picos de tráfico suceden alrededor de las 17 y 21 horas y los días 7 y 10 del mes.



En cuanto a los métodos, se ve que más del 75% de los accesos son utilizando un método GET, aproximadamente un 20% con el método POST y el resto en cantidades bajas por los métodos HEAD, OPTIONS y PROPFIND.

## 

## Referencias

[1] «Log Files: Definition, Types, and Importance | CrowdStrike», *crowdstrike.com*. https://www.crowdstrike.com/cybersecurity-101/observability/log-file/ (accedido 15 de abril de 2023).

[2] R. Irvine, «How to Troubleshoot Windows Problems Using Event Viewer Logs», *MUO*, 20 de noviembre de 2020. https://www.makeuseof.com/how-troubleshoot-windows-event-viewer-logs/ (accedido 15 de abril de 2023).

[3] SentinelOne, «How To Search Log Files: 3 Approaches To Extract Data | Scalyr», *SentinelOne*, 3 de noviembre de 2020. https://www.sentinelone.com/blog/how-search-log-files-extract-data/ (accedido 15 de abril de 2023).

[4] «Sematext Logs | Cloud Log Management Service | Hosted ELK», *Sematext*. https://sematext.com/logsene/ (accedido 15 de abril de 2023).

[5] M. communications@manageengine.com, «ManageEngine Log360», *ManageEngine Log360*. https://www.manageengine.com/log-management/ (accedido 15 de abril de 2023).

[6] A. McDonald, «Enhancing Visualization of Well Logs With Plot Fills», *Medium*, 16 de agosto de 2021. https://towardsdatascience.com/enhancing-visualization-of-well-logs-with-plot-fills-72d9dcd10c1b (accedido 15 de abril de 2023).

[7] «Reading and analyzing log files in the RRD database format». https://andrie.github.io/rrd/articles/using\_rrd.html (accedido 15 de abril de 2023).

[8] «PyQt5: Python bindings for the Qt cross platform application toolkit». Accedido: 15 de abril de 2023. [En línea]. Disponible en: https://www.riverbankcomputing.com/software/pyqt/

[9] «PyQtGraph - Scientific Graphics and GUI Library for Python». https://www.pyqtgraph.org/ (accedido 15 de abril de 2023).

[10] P. C. L. updated FAQ, «Which Python GUI library should you use in 2023?», *Python GUIs*, 24 de julio de 2022. https://www.pythonguis.com/faq/which-python-gui-library/ (accedido 15 de abril de 2023).

[11] M. F. P. FAQ, «PyQt vs. Tkinter: Which Should You Choose for Your Next Python GUI?», *Python GUIs*, 12 de abril de 2023. https://www.pythonguis.com/faq/pyqt-vs-tkinter/ (accedido 15 de abril de 2023).

[12] «Stack Overflow Developer Survey 2020», *Stack Overflow*. https://insights.stackoverflow.com/survey/2020/?utm\_source=social-share&utm\_medium=social&utm\_campaign=dev-survey-2020 (accedido 15 de abril de 2023).