



Escuela Técnica Superior de
Ingeniería Informática

TRABAJO FIN DE GRADO

Buscador de Información personal en Redes Sociales

Realizado por
Miguel Gómez Vázquez

Para la obtención del título de
Grado en Ingeniería Informática - Tecnologías Informáticas

Dirigido por
José Ángel Galindo Duarte

Realizado en el departamento de
Lenguajes y Sistemas Informáticos

**Convocatoria de Junio/Septiembre/Diciembre,
curso 2022/23**

Agradecimientos

A todas las personas que de una manera u otra aportaron su grano de arena para que hoy pueda estar aquí.

Resumen

El propósito de este TFG es crear una herramienta que pueda sacar de las redes sociales que hemos elegido, toda la información personal posible, es decir, elegimos una persona y que tendremos que poder obtener toda la información que dicha persona ha publicado de forma pública en redes sociales. Las redes sociales que hemos elegido han sido Twitter y Facebook.

Luego de haber conseguido la primera parte, el segundo objetivo es crear un entorno gráfico que permita visualizar los datos que hemos obtenido, de una forma limpia y ordenada, para que el cliente pueda usarla sin ningún tipo de problema.

Palabras clave: redes, sociales, facebook, twitter, scrapping, python, selenium, información, personal, buscador

Enlace al repositorio de Github: https://github.com/migueclon98/Proyecto_tfg.git

Abstract

The purpose of this TFG is to create a tool that can extract from the social networks that we have chosen, all possible personal information, that is, we choose a person and that we can obtain all the information that said person has published publicly on social networks. The social networks we have chosen have been Twitter and Facebook.

After having achieved the first part, the second objective is to create a graphic environment that allows us to visualize the data we have obtained, in a clean and orderly way, so that the client can use it without any problem.

Keywords: networks, social, facebook, twitter, scrapping, python, selenium, information, personal, search engine

Link to the repository: https://github.com/migueclon98/Proyecto_tfg.git

Índice general

1. Introducción	1
1.1. Contexto	1
1.1.1. OSINT	1
1.2. Motivación del proyecto	2
1.2.1. FOCA	3
1.3. Objetivos	5
1.3.1. Objetivos Docente	5
1.3.2. Objetivos Técnico	5
2. Requisitos	6
2.1. Introducción	6
2.2. Requisitos Funcionales	6
2.3. Requisitos NO Funcionales	7
2.4. Requisitos de Información	8
3. Planificación	9
3.1. Introducción	9
3.2. Planificación Temporal	9
3.2.1. Diagrama de GANTT	10
3.3. Estimación de costes	11
3.3.1. Coste Personal	11
3.3.2. Coste Material	11
3.3.3. Coste Total	11
4. Análisis	12
4.1. Introducción	12
4.1.1. Criterios de selección	12
4.1.2. Proceso de selección	12
4.2. Aspecto Legal	13
4.2.1. Legislación Española	13
4.2.2. Normativa Facebook	13
4.2.3. Normativa Twitter	14
4.2.4. Conclusión	14
4.3. Facebook	14
4.3.1. Web scraping	14
4.3.2. Selenium	15
4.4. Twitter	16
4.4.1. Snsrape	16
4.4.2. NLTK	17
4.5. Base de Datos	18
4.5.1. MongoDB	18
4.6. Interfaz Gráfica	20

5. Metodologías	21
5.1. Introducción	21
5.2. Metodologías ágiles	21
5.2.1. Kanban	22
5.3. Desarrollo en cascada	23
6. Tecnologías	25
6.1. Introducción	25
6.2. IDE	26
6.2.1. Eclipse	26
6.2.2. Visual Studio Code	26
6.3. Python	27
6.4. GitHub	28
7. Diseño	30
7.1. Introducción	30
7.2. Logo	30
7.3. UML	31
7.4. Facebook	33
7.4.1. Login	33
7.4.2. Busca amigos	33
7.4.3. Busca fotos	34
7.4.4. Busca info	34
7.4.5. Info work education	34
7.4.6. Info places lived	34
7.4.7. Info contact	34
7.5. Twitter	34
7.5.1. Nube de palabras	34
7.5.2. Opinión tema	35
7.6. Interfaz Gráfica	35
7.6.1. Página Principal	35
7.6.2. Página 1	35
7.6.3. Página 31	36
7.6.4. Página 2	36
7.6.5. Página 3	36
7.6.6. Página 4	36
7.6.7. Página 5	37
7.7. MongoDB	37
8. Implementación	38
8.1. Introducción	38
8.2. Facebook	38
8.3. Twitter	45
8.3.1. NLTK	45
8.3.2. Snsrape	47
8.4. Interfaz Gráfica	47
8.5. Futuros Plugin	48

9. Resultado	50
9.1. Pagina de inicio	50
9.2. Pagina búsqueda de Facebook	50
9.3. Datos de Facebook	51
9.4. Exportar TXT	52
9.5. Pagina busqueda de Twitter	53
10.Manual de Instalación	55
11.Conclusiones	56
11.1. Obtener Información	56
11.2. Ética	56
11.3. Concienciación	56
11.4. Malas Praxis	57
11.5. Objetivos	57
12.Planes Futuros	58
13.Bibliografía	59

Índice de figuras

1.1. Posibles campos de OSINT	2
1.2. Logo Aplicación FOCA	4
3.1. Diagrama de Gantt	10
3.2. Resumen Gantt	10
4.1. web scraping	15
5.1. Tabla Kanban	23
5.2. Desarrollo en cascada	24
6.1. Logo eclipse	26
6.2. Logo vsCode	26
6.3. Logo python	27
6.4. github-logo	28
7.1. logo del programa	31
7.2. UML	32
8.1. Inspeccionar página Web	40
8.2. Uso Extensión Selenium 1	41
8.3. IUso Extensión Selenium 2	42
8.4. Uso Extensión Selenium 3	43
8.5. Uso Extensión Selenium 4	44
9.1. github-logo	50
9.2. Apartado de Facebook	50
9.3. Obtención de datos Facebook	51
9.4. Exportar TXT	52
9.5. Creación del TXT	52
9.6. Apartado de Twitter	53
9.7. Rellenamos con una cuenta	53
9.8. Nube de palabras	53
9.9. Que opina sobre	54

Índice de extractos de código

8.1. Importaciones Facebook	38
8.2. Función Login	38
8.3. Función Login2	45
8.4. StopWords	46
8.5. Sentimental Analysis	46
8.6. Snsrape.Twitter	47
8.7. Interfaz Gráfica	47
8.8. Incluir Instagram a BBDD	48
8.9. Incluir Instagram a BBDD	48

Índice de cuadros

2.1. Requisitos de funcionales	6
2.2. Requisitos NO funcionales	7
2.3. Requisitos información	8

1. Introducción

1.1. Contexto

En la era digital en la que vivimos, las redes sociales se han convertido en una parte integral de la vida cotidiana de millones de personas en todo el mundo. Estas plataformas permiten a los usuarios compartir diversos aspectos de su vida, como intereses, ubicación, fotografías, relaciones personales y mucho más.

La gran cantidad de información personal disponible en las redes sociales ha generado preocupación sobre la privacidad y la seguridad de los usuarios. A medida que más y más personas comparten información personal en línea, surgen desafíos en cuanto a cómo se maneja, protege y utiliza esta información.

El objetivo de este proyecto es investigar y analizar las prácticas de recopilación de información personal en las redes sociales, examinando las implicaciones éticas y legales asociadas. Se buscará comprender cómo se recolecta y utiliza esta información, así como los posibles riesgos y beneficios asociados.

El proyecto abordará cuestiones relacionadas con la privacidad de los usuarios, la protección de datos y la seguridad en línea. Se realizará un análisis de las políticas de privacidad y las prácticas de las redes sociales más populares, con el fin de comprender cómo se obtiene, almacena y comparte la información personal de los usuarios.

Además, se evaluarán las implicaciones éticas de la recopilación de información personal y se propondrán posibles medidas y recomendaciones para proteger la privacidad de los usuarios en el entorno digital.

1.1.1. OSINT

La open source intelligence (OSINT) es una técnica usada por gobiernos y ejércitos para obtener información sobre amenazas, objetivos, países y otros a partir de datos públicamente disponibles.

Esta información puede encontrarse mediante diferentes métodos. Las páginas web y otros recursos que pueden encontrarse en Google son grandes fuentes de información open source, pero no son ni mucho menos la única. De hecho, según el exdirector de Google, Eric Schmidt, más del 99 por ciento del contenido disponible en internet no puede encontrarse con los motores de búsqueda convencionales. Esto es lo que se conoce como “deep web”. Gran parte de los contenidos de la deep web también se consideran open source, ya que están disponibles para el público mediante otros métodos.

Además, estas fuentes de información también se consideran open source:

- Contenidos publicados o emitidos para el público, por ejemplo, noticias.

- Datos disponibles mediante solicitud, por ejemplo, los datos catastrales de una vivienda.
- Datos disponibles mediante compra o suscripción, por ejemplo, publicaciones profesionales de un sector.
- Información obtenida visitando cualquier lugar o asistiendo a cualquier evento abierto al público.



Figura 1.1: Posibles campos de OSINT

Su ventaja frente a otros métodos de obtención de información es que no requiere permisos de seguridad especiales, por lo que no es necesario pertenecer a un organismo público para utilizarla.

En este proyecto solo nos adentraremos en la parte de internet, que incluye sitios web, foros y redes sociales.

1.2. Motivación del proyecto

Las motivaciones para realizar este TFG son diversas y variadas. A continuación, se presentan algunas de ellas:

1. Importancia actual: La recopilación de información personal en las redes sociales es un tema de gran relevancia en la sociedad actual. Con el crecimiento exponencial de las redes sociales y el aumento de la cantidad de información personal compartida en línea, es crucial comprender las implicaciones y los riesgos asociados.
2. Privacidad y seguridad: Existe una preocupación creciente sobre la privacidad y la seguridad de los usuarios en el entorno digital. La recopilación de información personal en las redes sociales plantea interrogantes sobre cómo se utiliza y protege esta información, y cómo puede afectar a la privacidad de los individuos.

3. Ética en la era digital: La ética en el uso de la información personal se ha convertido en un tema central en el mundo digital. Investigar y analizar las prácticas de recopilación de información personal en las redes sociales puede ayudar a identificar posibles dilemas éticos y proponer soluciones para proteger los derechos de los usuarios.

4. Cumplimiento legal y normativo: La recopilación de información personal en las redes sociales también tiene implicaciones legales y normativas. Examinar las políticas de privacidad y las prácticas de las redes sociales más populares puede permitir identificar posibles incumplimientos y proponer medidas para garantizar el cumplimiento de las leyes y regulaciones vigentes.

5. Investigación académica: El estudio de la recopilación de información personal en las redes sociales ofrece la oportunidad de contribuir al campo de investigación en informática y áreas relacionadas. Puede brindar nuevas perspectivas, enfoques o metodologías para abordar los desafíos actuales en materia de privacidad y seguridad en línea.

6. Sensibilización y concienciación: El proyecto tiene como objetivo crear conciencia entre los usuarios de las redes sociales sobre los riesgos asociados con la recopilación de información personal y promover buenas prácticas en cuanto a la protección de la privacidad.

1.2.1. FOCA

Otra gran motivación e inspiración para este trabajo es el proyecto de *FOCA*.

El proyecto *FOCA* (Fingerprinting Organizations with Collected Archives) es una herramienta de análisis y recopilación de información utilizada para la auditoría de seguridad y la evaluación de la exposición de una organización en línea. Fue desarrollado por ElevenPaths, una empresa de ciberseguridad y filial de Telefónica.

FOCA está diseñado para recopilar y analizar información disponible públicamente en diferentes fuentes, como motores de búsqueda, sitios web, redes sociales y documentos. Utiliza técnicas de reconocimiento de huellas digitales para identificar información sensible y valiosa sobre una organización, como nombres de dominio, direcciones IP, correos electrónicos, nombres de usuarios y documentos filtrados.

La herramienta proporciona una interfaz gráfica intuitiva que permite a los usuarios realizar búsquedas automatizadas, analizar meta datos de archivos, extraer información de documentos, identificar vulnerabilidades y realizar un mapeo de la infraestructura tecnológica de una organización. También puede generar informes detallados que ayudan a identificar posibles riesgos de seguridad y mejorar la postura de seguridad de la organización.

El proyecto *FOCA* ha sido utilizado tanto por profesionales de ciberseguridad como

por investigadores de seguridad para identificar posibles brechas de seguridad, obtener información relevante sobre una organización y realizar pruebas de penetración. Ayuda a las organizaciones a comprender su exposición en línea y tomar medidas para fortalecer su seguridad y proteger su información confidencial.

Es importante destacar que *FOCA* debe ser utilizado de manera ética y en conformidad con las leyes y regulaciones aplicables. Antes de utilizar la herramienta, se recomienda obtener el consentimiento adecuado y cumplir con las políticas de privacidad y seguridad de la organización objetivo.

Enlace al repositorio de Github: <https://github.com/ElevenPaths/FOCA.git>



Figura 1.2: Logo Aplicación FOCA

1.3. Objetivos

1.3.1. Objetivos Docente

- Ser capaz de aprender, estudiar y aplicar las técnicas de WebScrapping para obtener los datos necesarios de las distintas redes sociales.
- Ser capaz de una vez obtenida la información requerida poder buscar una metodología de gestión de datos suficientemente correcta para poder guardar y gestionar los datos, como por ejemplo podrían ser tecnologías SQL o NOSQL.
- Emplear métodos ágiles de manera correcta, que permitan la buena salud del proyecto y que avance correctamente.
- Describir los procedimientos en una memoria con una estructura clara. empleados para abordar los temas sugeridos además de la metodología utilizada. siempre defendiendo las decisiones tomadas.

1.3.2. Objetivos Técnico

Obtener una Aplicación Python que con una interfaz gráfica accesible que obtenga de varias redes sociales los datos públicos que las personas tenemos en las mismas.

2. Requisitos

2.1. Introducción

En este proyecto, dado que no hay un cliente que proponga los requisitos para el proyecto, debo ser yo el que lo haga, para esto me basaré en varios aspectos, como por ejemplo:

- Investigación de mercado
- Experiencia personal
- Iteración y mejora continua
- Consideraciones técnicas y recursos disponibles

2.2. Requisitos Funcionales

Los requisitos funcionales describen las funciones, acciones y comportamientos particulares que debe poseer el sistema o el software. Estas especificaciones describen lo que el sistema debe lograr y cómo debe reaccionar ante diversas entradas. Los casos de uso, los escenarios o las descripciones detalladas de las interacciones entre el usuario y el sistema se utilizan para expresar los requisitos funcionales.

En Nuestro proyecto estos son los Requisitos Funcionales:

Requisito	Descripción
RF1	La aplicación debe permitir la autenticación de usuarios para acceder a las funcionalidades de obtención y organización de información.
RF2	Los usuarios deben poder realizar búsquedas y obtener información personal de las personas a través de la interfaz gráfica.
RF3	La aplicación debe proporcionar la oportunidad de investigar por una sola red social o por ambas al mismo tiempo.
RF4	Se deben implementar mecanismos para que la información obtenida tenga los datos más actualizados posibles.
RF5	La aplicación debe permitir la exportación de los datos obtenidos en diferentes formatos, como CSV o Excel.

Cuadro 2.1: Requisitos de funcionales

2.3. Requisitos NO Funcionales

Los requisitos no funcionales son estándares de calidad, restricciones y atributos del sistema que no están tangencialmente conectados a funcionalidades particulares. Estas especificaciones enfatizan factores como la disponibilidad del sistema, el rendimiento, la facilidad de uso, la seguridad y la escalabilidad. Normalmente, los requisitos no funcionales son limitaciones o requisitos que el sistema debe satisfacer.

En nuestro caso los requisitos No Funcionales son:

Requisito	Descripción
RNF1	La aplicación debe ser eficiente y tener un tiempo de respuesta rápido al obtener y procesar la información.
RNF2	La interfaz gráfica debe ser intuitiva, fácil de usar y estéticamente agradable para los usuarios.
RNF3	La aplicación debe ser compatible con diferentes sistemas operativos, como Windows, macOS y Linux.
RNF4	Se deben implementar mecanismos de seguridad para proteger los datos almacenados y garantizar la confidencialidad e integridad de la información personal.
RNF5	La aplicación debe ser escalable y capaz de manejar un aumento en el número de usuarios y volúmenes de datos sin degradar su rendimiento.

Cuadro 2.2: Requisitos NO funcionales

2.4. Requisitos de Información

Con respecto a la gestión, el almacenamiento, el procesamiento y la presentación de la información dentro del sistema, los requisitos de información son el enfoque principal. Los datos necesarios, así como su estructura, formato, flujo y reglas de validación, están especificados por estos requisitos. La capacidad del sistema para administrar los datos necesarios para satisfacer los requisitos funcionales y no funcionales depende de la capacidad del sistema para manejar los requisitos de información.

En nuestro caso los requisitos de Información son:

Requisito	Descripción
RI1	La aplicación debe ser capaz de obtener información personal de las personas, como nombres, cumpleaños, biografía, etc., a través de distintas técnicas como web scraping.
RI2	La información personal recopilada debe ser almacenada en una base de datos NoSQL, como MongoDB, de manera estructurada y organizada.
RI3	Los datos obtenidos deben incluir identificadores únicos para cada persona, permitiendo un acceso rápido y eficiente a la información.
RI4	Se deben considerar los requisitos de privacidad y cumplimiento normativo al manipular y almacenar datos personales, asegurando el cumplimiento de las leyes de protección de datos aplicables.
RI5	La aplicación debe ser capaz de manejar y almacenar grandes volúmenes de datos de forma eficiente, garantizando la escalabilidad del sistema.

Cuadro 2.3: Requisitos información

3. Planificación

3.1. Introducción

En esta sección se aborda la Planificación del proyecto, tanto temporal como de costes. Realizar una correcta planificación temporal y de costes desde el inicio del desarrollo de un proyecto es un factor determinante para el éxito de este. Para los proyectos será vital que el análisis estimado de tiempo y coste sea lo más cercano al real, para evitar realizar las entregas fuera de plazo o exceder el presupuesto establecido.

3.2. Planificación Temporal

La planificación temporal de un proyecto informático se refiere a la definición y organización de las actividades y tareas del proyecto en función del tiempo. Es un proceso que permite establecer un cronograma detallado para llevar a cabo el proyecto de manera eficiente y cumplir con los plazos establecidos.

La planificación temporal implica identificar las actividades necesarias para completar el proyecto, establecer la secuencia lógica en la que deben llevarse a cabo, estimar la duración de cada actividad y asignar recursos adecuados para llevarlas a cabo.

Al realizar la planificación temporal de un proyecto informático, se pueden utilizar diversas técnicas y herramientas, como el diagrama de Gantt o el método de la ruta crítica, para visualizar y gestionar las dependencias entre las actividades, así como establecer fechas de inicio y finalización para cada una.

3.2.1. Diagrama de GANTT

El diagrama de Gantt es una herramienta de gestión de proyectos que se utiliza para visualizar y planificar las tareas a lo largo del tiempo. Proporciona una representación gráfica de las actividades del proyecto, su duración y las dependencias entre ellas.

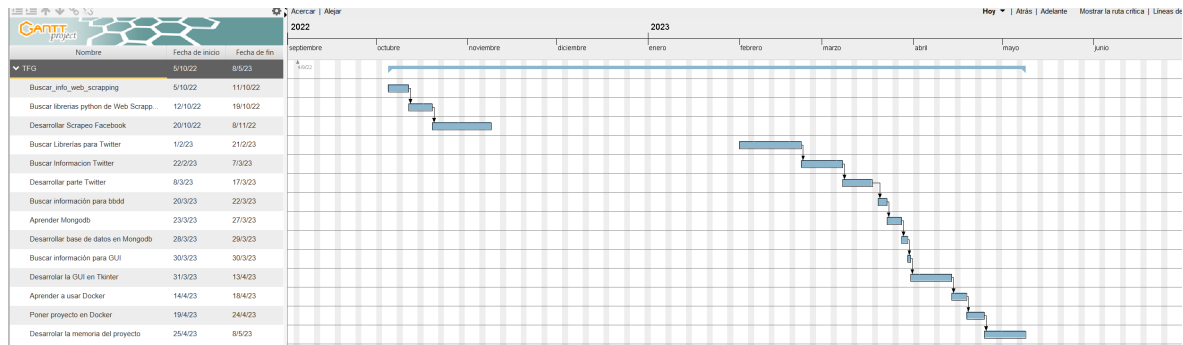


Figura 3.1: Diagrama de Gantt

Propiedades para TFG

General | Antecedentes | Recursos | Columnas personalizadas

Nombre: TFG

Proyecto: ☐

Opciones de programación: en este cuadro de diálogo ▾

Fecha de inicio: 5 de octubre de 2022

Fecha de finalización: 8 de mayo de 2023

Duración: 154

Fecha de inicio más temprana: ☐ 22 de mayo de 2023 Copiar fecha de inicio

Prioridad: El más alto

Progreso: 0

Mostrar en la línea de tiempo: ☐

Relleno:

Colores: Color Predeterminado

Página web:

Editar Notas

Figura 3.2: Resumen Gantt

3.3. Estimación de costes

3.3.1. Coste Personal

Teniendo en cuenta que solo hay encargado para todo el proyecto vamos a asignar a esa persona el cargo de Directores/as de proyecto, con esto y con el dato de que los ingenieros informáticos cobran de media en España, sacado del *INFORME FINAL SOBRE LA CONSULTA PRELIMINAR DEL MERCADO “PERFILES PROFESIONALES ÁMBITO INFORMÁTICO”*. [4] de media siendo junior, 45,40€/hora nos sale 363,2€/día, viendo el diagrama de Gantt hemos trabajado 154 días lo que nos saldría un total de 55932,8€

3.3.2. Coste Material

El costo material del proyecto se refiere a los gastos incurridos en la adquisición de los recursos físicos necesarios para llevar a cabo el proyecto. En este caso, dado que solo ha sido necesario un ordenador con un costo de 1200 euros, el costo material se limita a este único elemento.

3.3.3. Coste Total

Teniendo en cuenta los costes personales y los costes de material nos daría un total de:

$$55932,8 + 1200 = 57132,8 \text{Euros} \quad (3.1)$$

4. Análisis

4.1. Introducción

En esta sección, se presentará el análisis realizado para la selección de las herramientas y librerías utilizadas en la creación de la aplicación. El objetivo principal de este análisis fue identificar las tecnologías más adecuadas que proporcionaran las funcionalidades requeridas, optimizaran el rendimiento y facilitaran el desarrollo del proyecto. A continuación, se describirán los criterios y el proceso utilizado para tomar decisiones informadas en la elección de estas herramientas y librerías. Como base tendremos que todo el proyecto será elaborado en el lenguaje de programación Python.

4.1.1. Criterios de selección

En primer lugar, se establecieron los criterios que se tuvieron en cuenta para evaluar las distintas opciones disponibles. Estos criterios incluyeron, pero no se limitaron a, los siguientes aspectos:

Funcionalidad requerida: Se identificaron las funcionalidades necesarias para la aplicación, como la capacidad de realizar web scraping, acceder a la API de Twitter, almacenar y consultar datos en una base de datos NoSQL, y crear una interfaz gráfica amigable para el usuario.

Compatibilidad y interoperabilidad: Se consideró la compatibilidad de las herramientas y librerías con el lenguaje de programación principal utilizado en el proyecto (en este caso, Python), así como la capacidad de trabajar con las tecnologías de extracción de datos de las redes sociales y bases de datos NoSQL seleccionadas.

Documentación y comunidad de usuarios: Se examinó la disponibilidad y calidad de la documentación, así como la existencia de una comunidad activa de usuarios y desarrolladores que pudieran proporcionar soporte técnico y compartir buenas prácticas.

Rendimiento y escalabilidad: Se evaluó el rendimiento y la escalabilidad de las herramientas y librerías consideradas, teniendo en cuenta el volumen de datos a procesar y la capacidad de respuesta requerida por la aplicación.

4.1.2. Proceso de selección

Una vez establecidos los criterios de selección, se llevó a cabo un proceso de evaluación comparativa de las distintas opciones disponibles en el mercado. Este proceso incluyó las siguientes etapas:

Investigación preliminar: Se realizó una investigación exhaustiva de las herramientas y librerías disponibles en el ecosistema de desarrollo de software, identificando aquellas que cumplieran con los criterios establecidos.

Análisis y comparación: Se procedió a analizar y comparar las características, ventajas y desventajas de las herramientas y librerías preseleccionadas. Se tuvieron en cuenta aspectos como la estabilidad, la madurez, la comunidad de usuarios, las actualizaciones frecuentes y la facilidad de uso.

Pruebas de concepto: Se realizaron pruebas de concepto utilizando las herramientas y librerías más prometedoras, para evaluar su rendimiento, funcionalidad y facilidad de integración con los requisitos del proyecto.

Toma de decisiones: Finalmente, se tomaron las decisiones basadas en los resultados de las pruebas de concepto y la evaluación comparativa, seleccionando las herramientas y librerías más adecuadas para cada aspecto del proyecto.

4.2. Aspecto Legal

4.2.1. Legislación Española

En España tenemos la Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales[6]. Esta ley tiene como objetivo regular la protección de los datos personales y garantizar los derechos de los ciudadanos en el ámbito digital, se encarga de cosas como la protección de datos personales, el consentimiento informado, la responsabilidad de los responsables y encargados del tratamiento, los derechos digitales o el registro de actividades de tratamiento.

Cuando analizamos la ley vemos que se nombran a las redes sociales y la relación de los usuarios con ellas pero en ningún momento habla de terceras personas que puedan acceder de forma pública a dichos datos.

4.2.2. Normativa Facebook

Facebook, en sus políticas de datos[9] señalan que los usuarios tienen cierto control sobre la información que comparten públicamente en su perfil. Sin embargo, el acceso y uso de los datos públicos de otros usuarios puede estar sujeto a restricciones o limitaciones según las políticas y configuraciones de privacidad de cada usuario individual. Facebook también proporciona herramientas de privacidad que permiten a los usuarios controlar quién puede acceder y utilizar su información.

4.2.3. Normativa Twitter

Según los términos de servicio actuales de Twitter[3], los usuarios que publican información de forma pública en la plataforma generalmente otorgan a Twitter una licencia mundial, no exclusiva, libre de regalías y transferible para usar, modificar, reproducir y distribuir sus tweets. Esto implica que Twitter tiene el derecho de permitir el acceso y la extracción de contenido público por parte de terceros, incluyendo desarrolladores de aplicaciones.

Sin embargo, es importante tener en cuenta que los términos de servicio de Twitter pueden cambiar con el tiempo y es necesario revisar la versión más actualizada para obtener información precisa y completa sobre el uso de datos públicos de la plataforma.

Además, aunque Twitter permite el acceso a datos públicos, existen ciertas limitaciones y restricciones en el uso de su API y en la recopilación de datos.

4.2.4. Conclusión

Después de un exhaustivo análisis de las políticas de privacidad de Twitter y Facebook, así como de la legislación española, no hemos encontrado ninguna indicación que nos haga pensar que el uso de esta aplicación o su desarrollo contravengan de alguna manera dichas políticas.

Es importante tener en cuenta que esta conclusión se basa en el análisis realizado hasta la fecha de corte de nuestro conocimiento, y es recomendable mantenerse actualizado sobre los cambios en las políticas y la legislación vigente en todo momento.

4.3. Facebook

4.3.1. Web scraping

Durante el web scraping[1] (del inglés scraping = arañar/raspar) se extraen y almacenan datos de páginas web para analizarlos o utilizarlos en otra parte. Por medio de este raspado web se almacenan diversos tipos de información: por ejemplo, datos de contacto, tales como direcciones de correo electrónico o números de teléfono, o también términos de búsqueda o URL. Estos se almacenan en bases de datos locales o tablas.

El web scraping se utiliza para una gran variedad de tareas, por ejemplo, para recopilar datos de contacto o información especial con gran rapidez. En el ámbito profesional, el scraping se utiliza a menudo para obtener ventajas respecto a la competencia. De esta forma, por medio del harvesting de datos, una empresa puede examinar todos los productos de un competidor y compararlos con los propios. El web scraping también resulta valioso en relación con los datos financieros: es posible leer datos desde un sitio web externo, organizarlos en forma de tabla y después analizarlos y procesarlos.

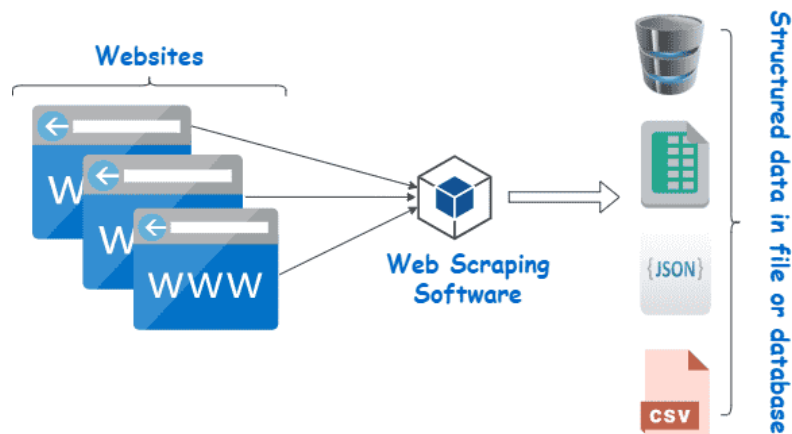


Figura 4.1: web scraping

Finalmente nos decidimos por esta herramienta ya que algunas de las redes sociales de las que queríamos obtener información no tiene una API publica que poder usar sin limitaciones, aunque esto sea una penalización a la hora de la ejecución ya que usar una API sería infinitamente más rápido no disponemos de ellas. Por ejemplo Twitter tiene un plan de usos para su API que tiene demasiadas restricciones y si queremos que nos eleven el proyecto para disponer de más endpoints tenemos que hacer una petición directamente a Twitter cosa que hicimos en un principio pero que por la falta de agilidad y la aparición de nuevas tecnologías que solucionaban el problema se descartó su uso. Por otro lado Facebook ni siquiera tiene una API publica a la que poder conectarse.

4.3.2. Selenium

La biblioteca Selenium[2] es una herramienta de automatización de pruebas de software que se utiliza principalmente para realizar pruebas funcionales y pruebas de regresión en aplicaciones web. Proporciona una interfaz de programación que permite controlar un navegador web de manera automatizada, imitando las acciones que un usuario realizaría en un navegador real.

La biblioteca Selenium es una herramienta de automatización de extracción de datos de sitios web. Desde el punto de vista del web scraping, Selenium se utiliza para simular la interacción de un usuario real con un navegador web y acceder a los datos deseados en una página web.

A diferencia de otras bibliotecas de web scraping que se basan en la extracción directa del código fuente HTML de una página web, Selenium permite interactuar con páginas web dinámicas o aquellas que requieren acciones específicas para cargar o mostrar contenido relevante.

Al utilizar Selenium para web scraping, se puede escribir código que simula las acciones del usuario, como hacer clic en botones, desplazarse por páginas, llenar formularios y esperar a que se cargue el contenido deseado. Esto permite acceder a datos que se

generan dinámicamente mediante JavaScript o que requieren acciones específicas del usuario.

Selenium proporciona métodos y funciones que permiten localizar elementos en una página web utilizando selectores CSS, XPath u otros métodos de búsqueda. Una vez que se han localizado los elementos deseados, se pueden extraer los datos o realizar acciones adicionales, como interactuar con formularios o hacer clic en enlaces.

Además de la funcionalidad básica de web scraping, Selenium también puede combinarse con otras bibliotecas y herramientas de web scraping, como BeautifulSoup, para procesar los datos extraídos y realizar operaciones más avanzadas.

En resumen, desde el punto de vista del web scraping, Selenium se utiliza para simular la interacción de un usuario real con una página web, permitiendo acceder a datos generados dinámicamente o que requieren acciones específicas del usuario. Esto lo convierte en una herramienta útil para extraer información de sitios web complejos o interactivos.

Aparte de todas estas razones Selenium dispone una extensión para chrome que te permite hacer una grabación de todos los pasos que haces en una web, desde pulsar un botón hasta rellenar un espacio en blanco, lo interesante de esta herramienta es que te indica los indicadores característicos de dichos actos, te especifica que path exacto tiene un boton o el CSS de una zona en específico, lo que nos será de mucha utilidad.

4.4. Twitter

4.4.1. Snsrape

Snsrape[7] es una herramienta de línea de comandos y una biblioteca de Python diseñada para realizar web scraping en redes sociales, centrándose principalmente en Twitter e Instagram. Esta herramienta permite extraer datos públicos de perfiles, publicaciones, comentarios y otros elementos de estas plataformas sociales.

La característica distintiva de Snsrape es su capacidad para obtener datos estructurados directamente de las páginas web de las redes sociales. A diferencia de otras bibliotecas o herramientas que dependen de APIs oficiales para acceder a los datos, Snsrape aprovecha la estructura HTML de las páginas web de Twitter e Instagram para extraer información.

Al utilizar Snsrape, puedes realizar consultas de búsqueda, especificar filtros y extraer datos relevantes de los perfiles de usuario, tweets, retweets, respuestas, hashtags, menciones y mucho más. También es posible realizar un seguimiento de las estadísticas de un perfil, como el número de seguidores, seguidos y me gusta.

Snsrape proporciona una interfaz de línea de comandos fácil de usar, lo que per-

mite realizar rápidamente consultas y extraer datos sin necesidad de escribir mucho código. También se puede utilizar como una biblioteca de Python, lo que proporciona flexibilidad y la posibilidad de integrarlo en proyectos de desarrollo.

Es importante tener en cuenta que Snsrape se adhiere a las políticas de uso aceptable de las redes sociales y no permite el acceso a datos privados o restringidos sin el consentimiento del propietario de la cuenta.

A pesar que a lo largo del desarrollo del proyecto la propia librería ha estado inhabilitado durante un cierto tiempo ya que la situación de la API oficial de Twitter ha ido cambiando, dejaba esta sección inhabilitada, decidimos finalmente usarla ya que cuando funciona es una herramienta realmente poderosa.

En resumen, Snsrape es una herramienta de web scraping específicamente diseñada para extraer datos públicos de redes sociales como Twitter e Instagram. Permite obtener información estructurada de perfiles, publicaciones y otros elementos, utilizando la estructura HTML de las páginas web de estas plataformas.

4.4.2. NLTK

NLTK[12] (Natural Language Toolkit) es una biblioteca de Python que proporciona un conjunto de herramientas y recursos para procesamiento del lenguaje natural (PLN). Es una de las bibliotecas más utilizadas y populares en el campo del PLN debido a su facilidad de uso, flexibilidad y variedad de funcionalidades.

NLTK incluye una amplia gama de módulos y recursos que abarcan diversas áreas del PLN, como tokenización, etiquetado de partes del discurso, análisis de sentimientos, extracción de información, análisis de concordancia, clasificación de textos, generación de lenguaje natural, entre otros. Además, también proporciona acceso a una variedad de conjuntos de datos lingüísticos, como corpus y léxicos.

Algunas de las características y funcionalidades clave de NLTK son:

1. Tokenización: NLTK ofrece métodos para dividir el texto en unidades más pequeñas, como palabras o frases, lo que facilita el procesamiento y análisis posterior.
2. Etiquetado de partes del discurso: Proporciona herramientas para asignar etiquetas gramaticales a las palabras, como sustantivos, verbos, adjetivos, etc., lo que permite realizar análisis gramaticales y sintácticos.
3. Análisis de sentimientos: NLTK incluye herramientas para analizar el tono emocional de un texto, como determinar si un texto es positivo, negativo o neutral.
4. Extracción de información: Permite extraer información estructurada y relevante de un texto, como nombres de personas, ubicaciones, fechas, etc.

5. Clasificación de textos: Ofrece algoritmos y métodos para clasificar textos en categorías predefinidas, lo que es útil en tareas como la clasificación de opiniones, detección de spam, entre otras.

6. Generación de lenguaje natural: NLTK proporciona herramientas para generar texto en lenguaje natural a partir de estructuras de datos y modelos lingüísticos.

NLTK también es utilizado como una herramienta educativa en el campo del PLN, ya que ofrece una amplia documentación y ejemplos de código que ayudan a comprender los conceptos y técnicas del procesamiento del lenguaje natural.

En resumen, NLTK es una biblioteca de Python ampliamente utilizada en el procesamiento del lenguaje natural. Proporciona una amplia gama de herramientas y recursos para tareas como tokenización, etiquetado de partes del discurso, análisis de sentimientos, extracción de información y clasificación de textos. Es una herramienta versátil y poderosa para el procesamiento de textos y análisis lingüístico.

4.5. Base de Datos

Las bases de datos no relacionales, también conocidas como bases de datos NoSQL (Not Only SQL), son sistemas de gestión de bases de datos que difieren de las bases de datos relacionales tradicionales en su estructura de almacenamiento y modelo de datos. A diferencia de las bases de datos relacionales, que utilizan tablas con filas y columnas interrelacionadas, las bases de datos NoSQL almacenan los datos de forma más flexible, como documentos, gráficos, clave-valor o columnas.

4.5.1. MongoDB

MongoDB[10] es una base de datos NoSQL popular y ampliamente utilizada, que se basa en el modelo de documentos. En lugar de almacenar datos en tablas, MongoDB almacena datos en documentos BSON (Binary JSON), que son estructuras de datos flexibles y jerárquicas similares a JSON.

La elección de MongoDB y la librería PyMongo para nuestro proyecto ha estado respaldada por varias razones:

1. Flexibilidad y escalabilidad: MongoDB permite almacenar datos no estructurados o semiestructurados, lo que brinda flexibilidad al adaptarse a cambios en los requerimientos de datos. Además, es escalable y puede manejar grandes volúmenes de datos y cargas de trabajo.

2. Modelo de datos documental: El modelo de datos basado en documentos de MongoDB facilita el almacenamiento y recuperación de datos complejos y anidados, lo que

puede ser beneficioso para tu proyecto que implica la organización de información personal de personas de redes sociales.

3. Fácil integración con Python: La librería PyMongo proporciona una interfaz sencilla y fácil de usar para interactuar con MongoDB en Python. PyMongo te permite realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en la base de datos, realizar consultas y manipular los documentos almacenados.

4. Amplia comunidad y soporte: MongoDB cuenta con una comunidad activa y en constante crecimiento, lo que significa que puedes encontrar recursos, documentación, ejemplos y soporte para resolver problemas o dudas que puedan surgir durante el desarrollo de tu proyecto.

Al elegir MongoDB y PyMongo, estás aprovechando las ventajas de una base de datos NoSQL flexible y potente, y la facilidad de uso y la compatibilidad con Python a través de la librería PyMongo.

4.6. Interfaz Gráfica

Tkinter[5] es una biblioteca estándar de Python que se utiliza para crear interfaces gráficas de usuario (GUI, por sus siglas en inglés). Proporciona un conjunto de herramientas y widgets que permiten desarrollar aplicaciones con una interfaz visual, lo que facilita la interacción del usuario con el programa.

Algunas características y razones por las que se puede haber utilizado Tkinter en tu proyecto son las siguientes:

1. Facilidad de uso: Tkinter es fácil de aprender y utilizar, especialmente para aquellos que ya están familiarizados con Python. Proporciona una API intuitiva y sencilla para construir interfaces gráficas de usuario de manera rápida y eficiente.
2. Compatibilidad multiplataforma: Tkinter es compatible con múltiples plataformas, lo que significa que las aplicaciones desarrolladas con Tkinter pueden ejecutarse en diferentes sistemas operativos, como Windows, macOS y Linux, sin necesidad de realizar cambios significativos en el código.
3. Amplia gama de widgets: Tkinter ofrece una amplia variedad de widgets (elementos visuales) predefinidos, como botones, etiquetas, cuadros de texto, listas desplegables, etc. Estos widgets permiten crear interfaces interactivas y personalizadas para satisfacer las necesidades específicas del proyecto.
4. Flexibilidad de diseño: Tkinter proporciona un sistema de geometría de ventanas que permite organizar y colocar los widgets en la interfaz de manera flexible. Esto facilita la creación de diseños personalizados y la disposición de los elementos según las preferencias de diseño del proyecto.
5. Integración con Python: Al ser parte de la biblioteca estándar de Python, Tkinter se integra perfectamente con el resto de las capacidades de Python. Esto significa que se puede utilizar el poder de Python para realizar operaciones lógicas, procesamiento de datos y otras tareas dentro de la aplicación GUI.

5. Metodologías

5.1. Introducción

En un proyecto informático, las metodologías se refieren a enfoques sistemáticos y estructurados utilizados para planificar, organizar, ejecutar y controlar el desarrollo de software. Estas metodologías proporcionan un marco de trabajo para abordar el ciclo de vida completo del proyecto, desde la concepción hasta la implementación y el mantenimiento.

En esta sección hablaremos de las metodologías que se han usado en el proyecto.

5.2. Metodologías ágiles

Las metodologías ágiles[8] son enfoques de desarrollo de software que se caracterizan por ser iterativos, colaborativos y flexibles. Estas metodologías ponen énfasis en la adaptabilidad, la entrega temprana de valor y la respuesta rápida a los cambios.

A diferencia de las metodologías tradicionales, como el modelo en cascada, las metodologías ágiles se centran en la interacción continua con el cliente y en la entrega incremental de software funcional. Los principios fundamentales de las metodologías ágiles incluyen:

Colaboración y comunicación constante: Fomentan la colaboración activa entre los miembros del equipo de desarrollo, los stakeholders y el cliente. Se prioriza la comunicación frecuente y efectiva para garantizar una comprensión clara de los requisitos y expectativas.

Entrega temprana y continua de valor: Se busca proporcionar al cliente entregas periódicas y funcionales del software para obtener retroalimentación temprana y adaptar el producto según las necesidades cambiantes.

Adaptabilidad y flexibilidad: Las metodologías ágiles aceptan y responden a los cambios en los requisitos y prioridades del proyecto. Permiten ajustes y modificaciones a medida que se avanza en el desarrollo.

Enfoque en equipos autorganizados: Los equipos ágiles tienen autonomía para tomar decisiones y organizarse internamente. Se promueve la responsabilidad compartida y la toma de decisiones colaborativa.

Algunas de las metodologías ágiles más conocidas son Scrum, Kanban, Extreme Programming (XP) y Lean Agile. Cada una tiene sus propias prácticas y marcos de trabajo

específicos, pero comparten la filosofía de adaptabilidad y entrega de valor de manera iterativa e incremental.

Las metodologías ágiles han ganado popularidad en la industria del desarrollo de software debido a su capacidad para responder a los cambios rápidos y ofrecer resultados más rápidamente. Ayudan a minimizar los riesgos y a mejorar la satisfacción del cliente al involucrarlo activamente en el proceso de desarrollo.

5.2.1. Kanban

Kanban es un sistema visual de gestión y control del flujo de trabajo, utilizado principalmente en entornos de desarrollo de software y gestión de proyectos. Se originó en Toyota como parte del sistema de producción Lean, y posteriormente se adoptó en otros contextos más allá de la fabricación.

El término Kanban significa tarjeta o tablero visual en japonés, y se basa en el uso de tarjetas o notas adhesivas para representar tareas o elementos de trabajo. Estas tarjetas se colocan en un tablero dividido en columnas que representan diferentes etapas del flujo de trabajo, desde el inicio hasta la finalización.

El objetivo principal del sistema Kanban es visualizar el trabajo, limitar la cantidad de trabajo en progreso y optimizar el flujo de trabajo para mejorar la eficiencia y la productividad. Algunos de los principios y conceptos clave de Kanban incluyen:

1. Tablero Kanban: Se utiliza un tablero físico o digital dividido en columnas que representan diferentes estados o etapas del trabajo, como "Por hacer", "En progreso" y "Completado". Las tarjetas o notas adhesivas se mueven de una columna a otra a medida que se avanza en el trabajo.
2. Límite de trabajo en progreso (WIP): Se establece un límite máximo de tarjetas permitidas en cada columna para evitar la sobrecarga de trabajo y el bloqueo del flujo. Esto fomenta la finalización de las tareas antes de comenzar nuevas, evitando así la acumulación de trabajo inacabado.
3. Pull System (Sistema de extracción): El trabajo se extrae de la columna anterior a medida que se liberan recursos para trabajar en él. Esto se realiza basándose en la capacidad y la disponibilidad del equipo, evitando así la asignación excesiva de tareas y el exceso de trabajo.
4. Mejora continua: Kanban promueve la mejora continua al permitir una visibilidad clara de los cuellos de botella y los problemas en el flujo de trabajo. Los equipos pueden identificar y abordar los obstáculos y buscar formas de optimizar y agilizar el proceso.

Además de su uso en el ámbito del desarrollo de software, Kanban también se aplica en diversos contextos, como la gestión de proyectos, el flujo de trabajo en equipos de trabajo, la gestión de tareas personales y la planificación de actividades.



Figura 5.1: Tabla Kanban

5.3. Desarrollo en cascada

El desarrollo de un proyecto informático en cascada es un enfoque tradicional y lineal para la gestión y ejecución de proyectos. También se le conoce como el modelo de ciclo de vida en cascada. En este enfoque, el proyecto se divide en etapas secuenciales, donde cada etapa debe completarse antes de pasar a la siguiente.

Las etapas principales del modelo de cascada:

1. Requisitos: En esta etapa, se recopilan y documentan todos los requisitos del proyecto, es decir, qué debe hacer el sistema o software a desarrollar. Se definen las funcionalidades, características y restricciones.

2. Diseño: Una vez que los requisitos están claros, se pasa a la etapa de diseño. Aquí se crea la arquitectura del sistema, se diseñan los componentes y se establecen las interfaces entre ellos. También se definen las bases de datos, el flujo de datos y las interfaces de usuario.

3. Implementación: En esta fase, se lleva a cabo la codificación del software siguiendo el diseño previamente establecido. Los programadores desarrollan el código fuente utilizando el lenguaje de programación adecuado y las mejores prácticas de desarrollo.

4. Pruebas: Después de la implementación, se realizan pruebas exhaustivas para verificar que el software funcione correctamente. Se identifican y corrigen los errores y se asegura que se cumplan los requisitos establecidos.

5. Despliegue: Una vez que el software ha pasado las pruebas y se considera estable, se procede a su despliegue en el entorno de producción. Esto implica la instalación y configuración del software en los sistemas del cliente o en los servidores correspondientes.

6. Mantenimiento: En esta etapa, se brinda soporte continuo y se realizan mejoras al software. Se pueden corregir errores, agregar nuevas funcionalidades o realizar actualizaciones según las necesidades del cliente.

Una característica importante del modelo en cascada es que cada etapa debe

completarse antes de pasar a la siguiente, y los cambios en etapas anteriores pueden ser costosos y difíciles de implementar una vez que se ha avanzado en el proceso. Esto significa que este enfoque puede no ser adecuado para proyectos donde los requisitos son propensos a cambios o para aquellos que requieren una entrega rápida y iterativa.

Aunque sabemos que esta metodología usada de forma aislada no es una metodología ágil, usada al mismo tiempo que la explicada anteriormente, Kanban, puede ser una buena opción. El desarrollo en cascada no suele ser una buena metodología ya que el cliente no puede ver un producto medianamente terminado hasta el final del proyecto, pero como en este proyecto el cliente y el desarrollador son la misma persona no ha habido ese problema, y dada la situación ha sido el sistema elegido para el desarrollo del proyecto.

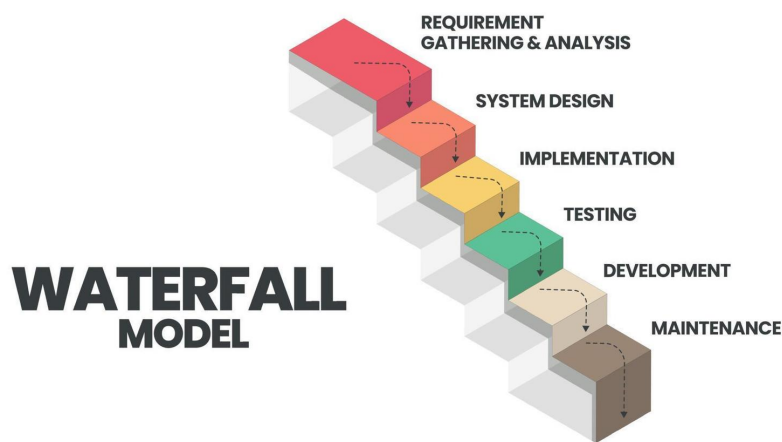


Figura 5.2: Desarrollo en cascada

6. Tecnologías

6.1. Introducción

En esta sección se expondrá las distintas tecnologías que se han usado para el desarrollo del proyecto. Las tecnologías utilizadas en un proyecto informático son las herramientas, lenguajes de programación, frameworks, bibliotecas y plataformas que se emplean para desarrollar, implementar y mantener el software. Estas tecnologías pueden variar según los requisitos del proyecto y el tipo de aplicación que se esté desarrollando. Algunas de las tecnologías comunes en un proyecto informático pueden incluir:

Lenguajes de programación: Los lenguajes de programación son la base para el desarrollo de software. Algunos lenguajes comunes son Python, Java, C++, JavaScript, Ruby, entre otros.

Frameworks y bibliotecas: Los frameworks y bibliotecas proporcionan un conjunto de herramientas y funcionalidades predefinidas que facilitan el desarrollo de software. Ejemplos populares incluyen Django y Flask para Python, React y Angular para JavaScript, Spring para Java, etc.

Bases de datos: Las bases de datos se utilizan para almacenar y gestionar la información en un proyecto informático. Algunas opciones populares son MySQL, PostgreSQL, MongoDB, SQLite, entre otras.

Herramientas de control de versiones: Las herramientas de control de versiones, como Git, son utilizadas para gestionar y controlar los cambios en el código fuente y colaborar en equipo.

Plataformas en la nube: Las plataformas en la nube, como AWS, Azure y Google Cloud, ofrecen servicios y recursos para el despliegue, escalabilidad y gestión de aplicaciones.

Herramientas de desarrollo integrado (IDE): Los IDEs, como PyCharm, Eclipse, Visual Studio Code, proporcionan un entorno de desarrollo completo con características y herramientas para programar, depurar y gestionar proyectos.

Tecnologías web: Para proyectos web, se utilizan tecnologías como HTML, CSS y JavaScript para el desarrollo de la interfaz de usuario y la interactividad en el navegador.

La elección de las tecnologías depende de los requisitos del proyecto, la experiencia del equipo de desarrollo, la escalabilidad necesaria y otros factores. Es importante evaluar las ventajas, desventajas y compatibilidad de las tecnologías para tomar decisiones informadas y construir soluciones eficientes.

6.2. IDE

6.2.1. Eclipse



Figura 6.1: Logo eclipse

Eclipse es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) de código abierto ampliamente utilizado para el desarrollo de software. Proporciona una plataforma robusta y extensible que admite múltiples lenguajes de programación, incluyendo Java, C++, Python, PHP, y más.

Eclipse se caracteriza por su capacidad para facilitar el desarrollo de aplicaciones complejas al ofrecer una amplia gama de características y herramientas.

La principal razón por la cual he usado este IDE ha sido por su sistema de *DEBUG* ya que era al que más habituado estaba, al acabar con el periodo de pruebas y por lo tanto no necesitar más el sistema de *DEBUG* migré el proyecto hacia Visual Studio Code ya que hay algunos aspectos de Eclipse que no son compatibles con el resto de IDE's por ejemplo la forma de llamar a los paquetes. Esto fue un problema por lo que a partir de ahora para futuros proyectos usaremos directamente Visual.

6.2.2. Visual Studio Code



Figura 6.2: Logo vsCode

Visual Studio Code (VS Code) es un entorno de desarrollo de código abierto y multiplataforma creado por Microsoft. Es ampliamente utilizado por desarrolladores de software para escribir, depurar y administrar código en diversos lenguajes de programación, como JavaScript, Python, C++, Java y muchos otros.

VS Code se destaca por su ligereza, su interfaz de usuario intuitiva y su amplia gama de extensiones que permiten personalizar y ampliar sus funcionalidades

6.3. Python



Figura 6.3: Logo python

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general. Fue creado por Guido van Rossum y lanzado por primera vez en 1991. Python se destaca por su sintaxis clara y legible, lo que lo hace muy adecuado para principiantes en programación, así como para desarrolladores experimentados.

Algunas características y puntos destacados de Python incluyen:

Sintaxis sencilla: Python utiliza una sintaxis simple y fácil de leer, lo que facilita la comprensión y el aprendizaje del lenguaje. Esto hace que sea rápido de escribir y de mantener el código.

Lenguaje interpretado: Python es un lenguaje interpretado, lo que significa que no necesita ser compilado antes de ser ejecutado. Esto permite un desarrollo más rápido y una mayor flexibilidad en el desarrollo.

Amplia biblioteca estándar: Python cuenta con una biblioteca estándar extensa que proporciona módulos y funciones para una amplia gama de tareas, como manipulación de archivos, acceso a bases de datos, networking, procesamiento de texto, entre otros. Esto evita tener que escribir código desde cero para muchas tareas comunes.

Multiplataforma: Python es compatible con múltiples plataformas, lo que significa que los programas escritos en Python se pueden ejecutar en diferentes sistemas operativos, como Windows, macOS y Linux, sin necesidad de realizar cambios significativos.

Enfoque en la legibilidad: Python pone énfasis en el código legible y bien estructurado. Utiliza sangrías (indentación) en lugar de llaves para delimitar bloques de código, lo que fomenta la escritura de código limpio y organizado.

Amplia comunidad y soporte: Python cuenta con una gran comunidad de desarrolladores activos que contribuyen con bibliotecas, frameworks y herramientas. Esto proporciona una amplia gama de recursos y soporte para el desarrollo de proyectos en Python.

Python es utilizado en una amplia variedad de aplicaciones, como desarrollo web, ciencia de datos, inteligencia artificial, automatización de tareas, scripting y más. Su popularidad se debe a su facilidad de uso, versatilidad y la gran cantidad de recursos disponibles que facilitan el desarrollo de software eficiente y de calidad.

6.4. GitHub



Figura 6.4: github-logo

GitHub es una plataforma web de desarrollo colaborativo basada en el sistema de control de versiones Git. Permite a los desarrolladores y equipos de desarrollo trabajar de manera conjunta en proyectos de software, realizar un seguimiento de los cambios realizados en el código fuente y coordinar la colaboración en un entorno centralizado.

Algunos aspectos destacados de GitHub incluyen:

Control de versiones distribuido: GitHub utiliza el sistema de control de versiones distribuido Git, que permite a los desarrolladores realizar un seguimiento de los cambios en el código fuente de manera eficiente y realizar ramificaciones y fusiones de forma sencilla. Cada desarrollador tiene una copia completa del repositorio, lo que facilita el trabajo en paralelo y la colaboración.

Repositorios públicos y privados: En GitHub, se pueden crear repositorios tanto públicos como privados. Los repositorios públicos son visibles para todos y fomentan la colaboración y el intercambio de código abierto, mientras que los repositorios privados son accesibles solo para los colaboradores seleccionados y brindan mayor privacidad y seguridad.

Colaboración y gestión de proyectos: GitHub proporciona herramientas y funciona-

lidades para facilitar la colaboración entre desarrolladores en un proyecto. Los colaboradores pueden realizar contribuciones al código a través de solicitudes de extracción (pull requests), realizar comentarios en el código, realizar seguimiento de problemas y tareas (issues), y utilizar tableros de proyectos para la gestión del flujo de trabajo.

Integraciones y servicios: GitHub se integra con una amplia gama de herramientas y servicios populares utilizados en el desarrollo de software, como integración continua (CI/CD), sistemas de seguimiento de errores, servicios de despliegue y más. Estas integraciones permiten automatizar tareas y mejorar la eficiencia en el flujo de trabajo de desarrollo.

Comunidad y código abierto: GitHub es conocido por su vibrante comunidad de desarrolladores, donde se comparten y colaboran en una amplia variedad de proyectos de código abierto. Los desarrolladores pueden descubrir proyectos interesantes, contribuir con su código y aprender de otros miembros de la comunidad.

[11]

7. Diseño

7.1. Introducción

El diseño de un proyecto informático es una etapa crucial para asegurar el éxito y la eficiencia en el desarrollo del mismo. En esta sección veremos como se ha diseñado el proyecto, que clases de relaciones y dependencias habrá con los distintos módulos python que vayamos creando.

Para empezar lo primero que tendremos en cuenta serán los [objetivos](#) y [requisitos](#) fijos anteriormente. Ya analizados las herramientas que vamos a usar, vamos a describir como pensamos usarlas.

Como el objetivo del proyecto es obtener la máxima información pública que los usuarios de las redes sociales ofrecen públicamente, se ha decidido que la extracción de los datos de Facebook irán orientados a la información personal, datos como edad, vivienda, conocidos... Y por otro lado Twitter servirá para obtener datos después de procesar los distintos mensajes publicos(Tweets) que se puedan obtener para sacar conclusiones sobre le tono (positivo/neutro/negativo).

7.2. Logo

Como se ha explicado al inicio de este proyecto, una de las inspiraciones que ha servido para idear este proyecto es la herramienta *FOCA*, que de sus sigla *Fingerprinting Organizations with Collected Archives*, sale una palabra en español. En un alarde de originalidad y en señal de aprecio y homenaje a dicha aplicación, siendo el acrónimo de este proyecto BIRS, se ha diseñado el siguiente logo:



Figura 7.1: logo del programa

7.3. UML

Se utiliza una representación visual conocida como diagrama UML (Lenguaje de modelado unificado) para modelar y comunicar varios aspectos de un sistema o proceso. El desarrollo de software y otros campos relacionados con la ingeniería de software suelen utilizar el lenguaje de modelado estándar conocido como UML.

Los diversos componentes y relaciones de un sistema se pueden ver, especificar, construir y documentar mediante diagramas UML. Estas representaciones de las estructuras, los comportamientos y las relaciones de un sistema se basan en un conjunto de convenciones y elementos gráficos uniformes que facilitan su lectura y comprensión.

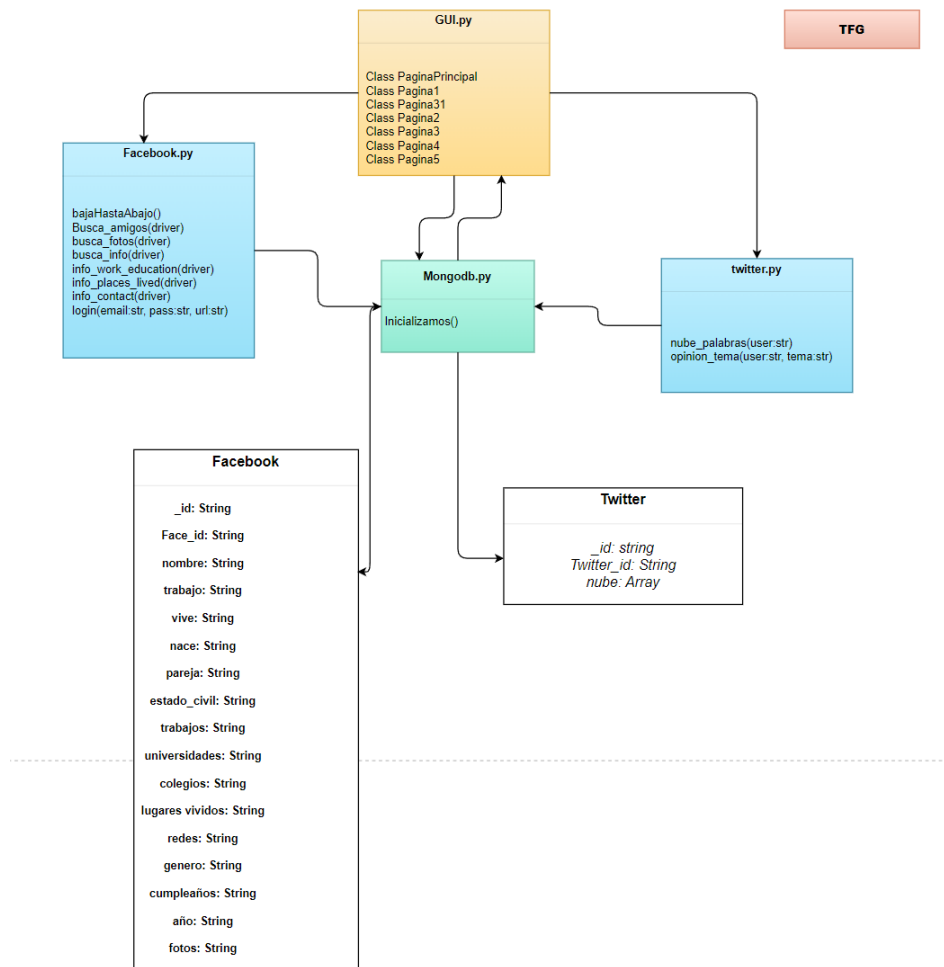


Figura 7.2: UML

7.4. Facebook

Para la parte de Facebook tendremos que desarrollar una serie de funciones que nos permitan recorrer todos los puntos claves que nos ofrece las paginas de información de los usuarios en la red social. Serán:

- login(email, password,url)
- bajaHastaAbajo().
- busca amigos(driver)
- busca fotos(driver)
- busca info(driver)
- info work education(driver)
- info places lived(driver)
- info contact(driver)

7.4.1. Login

Esta será la parte más critica de esta zona ya que usaremos esta función para crear el Driver que nos dará acceso al navegador y desde donde serán llamadas el resto de funciones. En esta función tenemos como entradas el usuario y contraseña de la persona que va a usar la aplicación ya que todo el scrapeo de la misma será llevado acabo desde su cuenta personal, se intentó otros modos que no implicasen un logueo pero Facebook restringe prácticamente toda la información de los perfiles si los observas sin estar logueado, lo que no nos dejo otra opción que tener que loguearnos.

Con esta parte nos aseguráramos de haber cumplido el requisito RF1, para poder obtener la información se debe identificar el usuario en la red social.

Teniendo en cuenta que la obtención de los datos se obtienen de la página web en el mismo momento que se realiza la búsqueda podemos dar por realizado el requisito RF4.

7.4.2. Busca amigos

Esta función utiliza Selenium para acceder a la página de amigos de una persona en Facebook y recopila el nombre y el enlace de los primeros 5 amigos encontrados, devolviendo una lista de tuplas con esta información.

7.4.3. Busca fotos

Esta función utiliza Selenium para acceder a la página de amigos de una persona en Facebook y recopila el nombre y el enlace de los primeros 5 amigos encontrados, devolviendo una lista de tuplas con esta información.

7.4.4. Busca info

Esta función utiliza Selenium para acceder a la página de información de una persona en Facebook y recopilar todo los datos que allí encontramos, como trabajo actual, donde estudió, residencia..., devolviendo una lista de tuplas con esta información.

7.4.5. Info work education

Esta función utiliza Selenium para acceder a la página de Work and Education de una persona en Facebook y recopilar todo los datos que allí encontramos, devolviendo una lista de tuplas con esta información.

7.4.6. Info places lived

Esta función utiliza Selenium para acceder a la página de Places Lived de una persona en Facebook y recopila todo los datos que allí encontramos, devolviendo una tupla con esta información.

7.4.7. Info contact

Esta función utiliza Selenium para acceder a la página de Contact and basic info de una persona en Facebook y recopilar todo los datos que allí encontramos, devolviendo una tupla con esta información.

7.5. Twitter

Para la parte de Facebook tendremos que desarrollar una serie de funciones que nos permitan recorrer todos los últimos X tweets de una cuenta, luego tendremos que tratarlos para obtener información.

7.5.1. Nube de palabras

Esta función, tiene como entrada una cuenta de Twitter y tendrá como objetivo recopilar de los últimos X tweets las palabras más usadas, tendremos que usar algunas

de las librerías definidas anteriormente para quitar palabras que no aporten valor como lo podrían ser: pronombres, artículos ...

7.5.2. Opinión tema

Esta función tendrá como entrada una cuenta de Twitter y una palabra, su objetivo será recorrer todos los tweets publicados por dicha cuenta en los que haya usado la palabra dada como entrada, por cada tweet le daremos una valoración de si usa la palabra en tono despectivo, neutro o positivo, nos ayudaremos de las librerías anteriormente explicadas. Una vez se hayan procesado todos los tweets se devolverá la media aritmética de todos ellos y cual es el tono general que usa.

7.6. Interfaz Gráfica

La parte de la Interfaz gráfica la iremos explicando através de las distintas pantallas que nos vamos encontrando, explicaremos como funcionará cada una y que se encuentra en ellas.

Una vez esté completada la interfaz gráfica tendremos cumplido el requisito RF2.

7.6.1. Página Principal

Será la página de inicio de la aplicación en la que nos encontraremos solamente tres objetos, que serán botones con los dos símbolos de las dos redes sociales que vamos a explorar, uno de Facebook y uno de Twitter y un botón con el símbolo de un texto que nos llevará a una página para poder extraer en un .TXT la información. Tendrá también un titulo de inicio.

Dando opciones de que red social queremos investigar podemos decir que se cumple el requisito RF3.

7.6.2. Página 1

A la página uno llegaremos después de haber presionado el botón de Facebook, una vez en la misma nos encontraremos:

1. Boton de vuelta a la página principal.
2. Rellenable pidiendo la cuenta de la persona buscada.
3. Rellenable pidiendo el correo de la cuenta desde la cual se va a scrapear.
4. Rellenable pidiendo la contraseña de la cuenta desde la cual se va a scrapear.
5. Boton de lanzamiento para la búsqueda.

El Botón de lanzamiento para la búsqueda nos llevará a la página 31, pasaremos en la misma acción de botón la información rellena por la cuenta que queremos buscar, el correo y contraseña de la persona que quiere investigar.

7.6.3. Página 31

En esta página se lanza una petición a la base de datos buscando información de la cuenta recibida, en caso positivo la página será automáticamente rellena con los datos que estaban almacenados, en caso contrario se lanza una instancia a la función *LOGIN* con los datos recibidos la cual se encargará de comenzar la extracción de los mismos. Una vez terminada la función esos datos se guardarán en la base de datos.

7.6.4. Página 2

La página dos es la página de inicio de twitter en la cual podremos encontrar:

1. Botón de vuelta a la página principal.
2. Rellenable pidiendo la cuenta de la persona buscada.
3. Boton de lanzamiento para la búsqueda.

El Botón de lanzamiento para la búsqueda nos llevará a la página 31, pasaremos en la misma acción de botón la información rellena por la cuenta que queremos buscar.

7.6.5. Página 3

En la página 3 haremos una llamada a la base de datos buscando por el nombre de la cuenta, en caso negativo se hará una llamada a la función *Nube de Palabras* la cual una vez terminada guardará los datos en la base de datos, luego de esto aparecerá por pantalla el Top 10 de palabras más usadas por la cuenta, cada palabra estará en un tipo botón, si pulsamos cualquiera de ellas nos llevará a la página 4.

7.6.6. Página 4

En la página cuatro recibimos como variables la palabra anteriormente pulsada y la cuenta de la que se trata, por lo tanto lo que haremos en esta pantalla será una llamada a la función *QueOpina* que una vez terminada nos dará una pequeña ventana que nos dirá en que tono suele usar la palabra cuando la usa en los tweets.

7.6.7. Página 5

La página 5 será una pagina donde daremos los datos que dábamos en las anteriores páginas pero en este caso con el objetivo de recopilar toda esa información en un .TXT que el usuario recibirá en la dirección que se encuentre el proyecto.

Dada esta posibilidad de obtención de los datos tenemos completado el requisito RF5.

7.7. MongoDB

Como se ha ido viendo a lo largo de la explicación del diseño el mayor uso de la base de datos está en el propio uso de la interfaz gráfica, solo hay un módulo que inicia la base de datos y lo usaremos la primera vez que iniciemos la app.

Dado que el tiempo de ejecución del programa dependerá si el dato buscado está ya en la base de datos o no, tener este almacenamiento al menos reduce radicalmente el tiempo de búsqueda a partir de la segunda vez que se busque dichos datos, por lo tanto podemos dar por realizado los requisito RNF1 y RNF5 ya que esta forma de agrupar los datos nos permite escalar la aplicación con gran margen.

8. Implementación

8.1. Introducción

El objetivo principal de esta implementación es materializar los conceptos teóricos y las metodologías propuestas en la parte inicial del TFG. Aquí, se traducen las ideas en código funcional que permite validar y verificar las hipótesis planteadas, así como proporcionar un análisis cuantitativo y cualitativo de los resultados obtenidos.

Se explorarán las herramientas y lenguajes de programación utilizados, así como las bibliotecas y frameworks empleados para la implementación del código. Asimismo, se explicarán con claridad las partes más repetitivas del código y los patrones usados, para que luego en su lectura pueda estar todo más claro.

Lo dividiremos en tres partes que serán las implementaciones de las funciones de Facebook, las de Twitter y de la interfaz gráfica.

8.2. Facebook

```
from sys import exit
import os
from selenium import webdriver
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
import time
import wget
```

Extracto de código 8.1: Importaciones Facebook

Estas son las primeras líneas de código que nos encontramos en el archivo Facebook.py que nos servirán para importar las herramientas que estaremos usando a lo largo del código. En ellas podemos ver que incluimos varias instancias distintas de la librería Selenium, la cuál nos servirá para las técnicas de WebScrapping.

```
def login(email, password, url):
    global driver
    global persona
    persona=url
    options = Options()
    #Code to disable notifications pop up of Chrome Browser
    options.add_argument("--disable-notifications")
    options.add_argument("--disable-infobars")
```



```

options.add_argument("--mute-audio")
options.add_argument('--headless')
try:
    #driver = webdriver.Chrome(service=Service(
        ChromeDriverManager().install()))
    driver = webdriver.Chrome("Selenium_drivers\
        chromedriver.exe")
    print("you_logged_in._Let's_rock")
except:
    print("you_need_web_driver!")
    exit()
driver.get("https://facebook.com")
# filling the form
driver.find_element(By.XPATH, "/html/body/div[3]/div[2]/
    div/div/div/div/div[4]/button[2]").click()

```

Extracto de código 8.2: Función Login

Este es el inicio de la función principal del modulo de Facebbok, en la que vemos que recibimos como parámetro los datos de acceso y la cuenta que queremos investigar.

La cuenta deberá ser la que aparezca en la url del perfil, es decir:

<https://www.facebook.com/profile.profile> pues el dato que debemos ingresar será el **profile.profile**

Si seguimos con el código vemos las opciones que le daremos al navegador entre ellas podemos ver algunas como mutear las notificaciones o el audio. Seguidamente creamos la variable driver que será el corazón de la aplicación ya que será la variable sobre la que orbitará todas las demás funciones. Una vez creada necesitamos rellenar el formulario que nos da Facebook para loguearnos. Para ello necesitamos saber las direcciones de todos los elementos para saber sobre que hacer click o donde tenemos que rellenar, para solucionar esto tenemos dos opciones:

Inspeccionar

Si hacemos click derecho en una página y pulsamos la opción de inspeccionar podemos ver la descomposición HTML de la propia página, si en esta nueva sección pulsamos el botón de flecha situado en la parte superior izquierda de la nueva parte oscura, podremos señalar exactamente que queremos señalar y en la parte de la derecha nos dirá exactamente su dirección es decir su XPATH, sabiendo esto y haciendo click derecho donde se sitúa la acción que estemos buscando nos dará la opción de copiar el PATH que es lo que usaremos en nuestro proyecto cuando estemos buscando objetos por el XPATH, por ejemplo:

driver.findElement

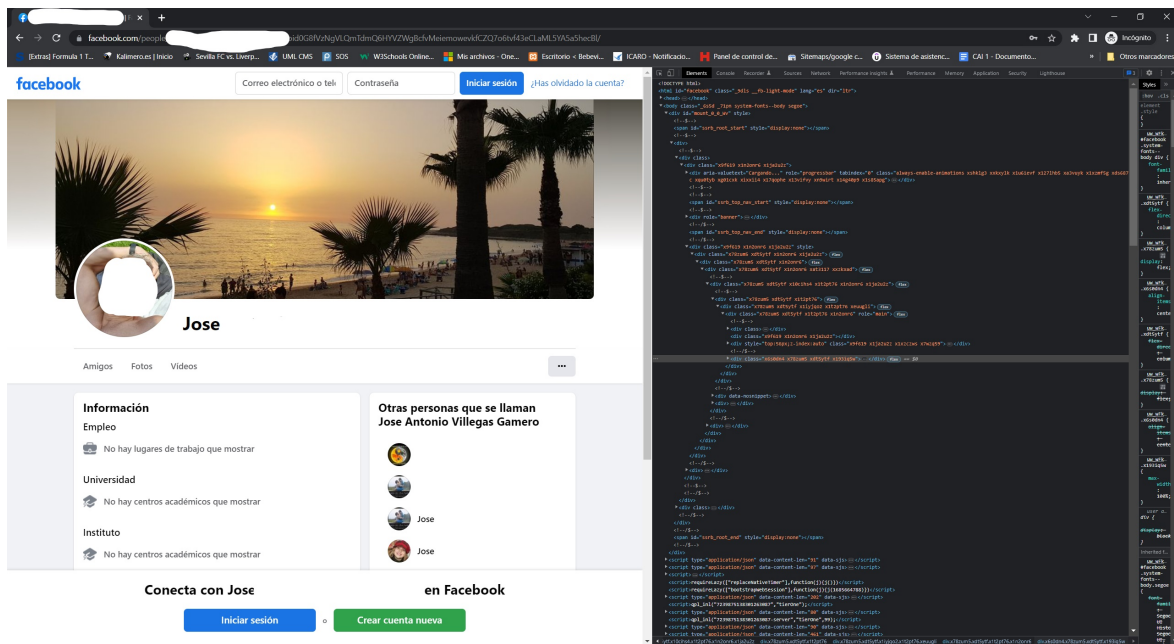


Figura 8.1: Inspeccionar página Web

(By.XPATH,/html/body/div[3]/div[2]/div/div/div/div/div[4]/button[2]).click()

En este ejemplo estamos haciendo click sobre el botón inicial para aceptar las cookies.

Selenium Chrome Extension

Esta extensión de Chrome agiliza mucho la función de objetos difíciles de seleccionar ya que no solo los identifica por su path, también puede darte su código CSS y demás identificadores. Para terminar una de las funciones más usadas para este proyecto es la capacidad de dada una grabación de uso en una web puede exportar paso por paso las acciones en el lenguaje de programación que estes usando, en este caso, Python. aquí dejo un ejemplo de uso:

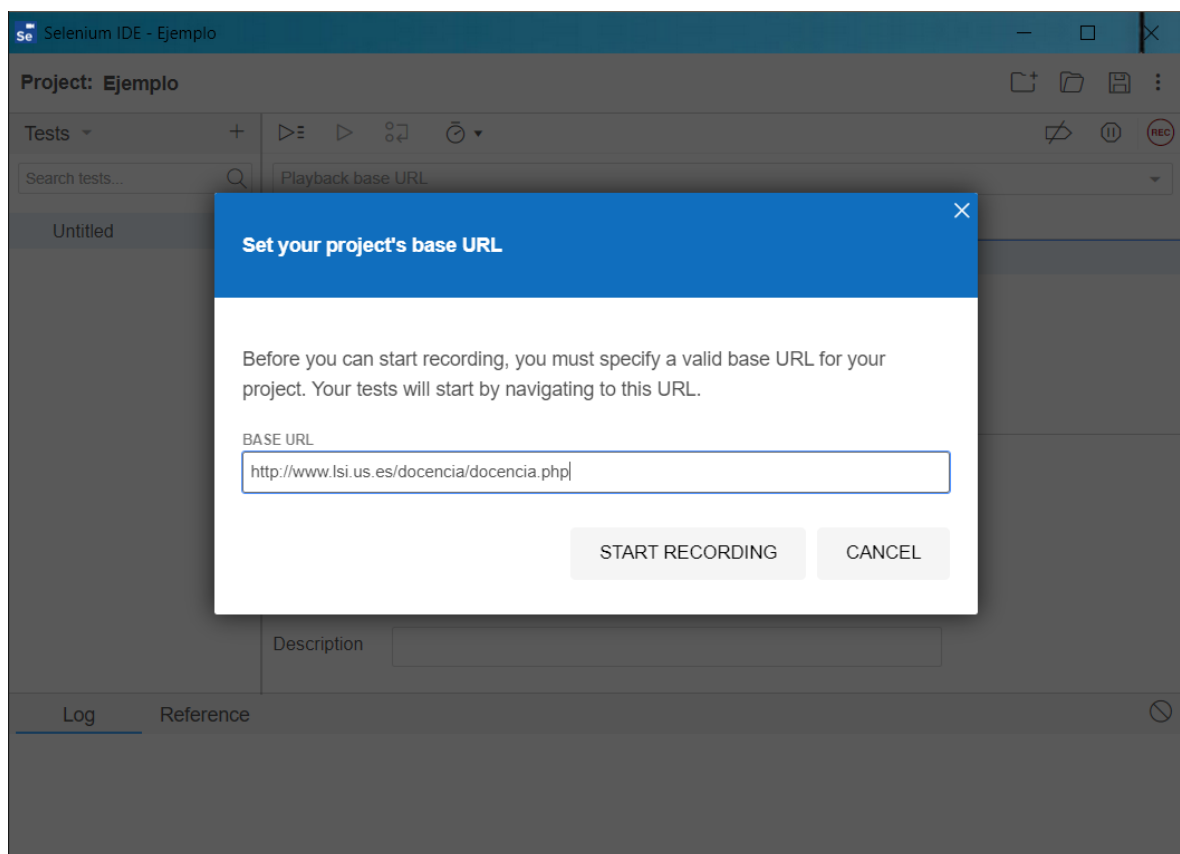


Figura 8.2: Uso Extensión Selenium 1

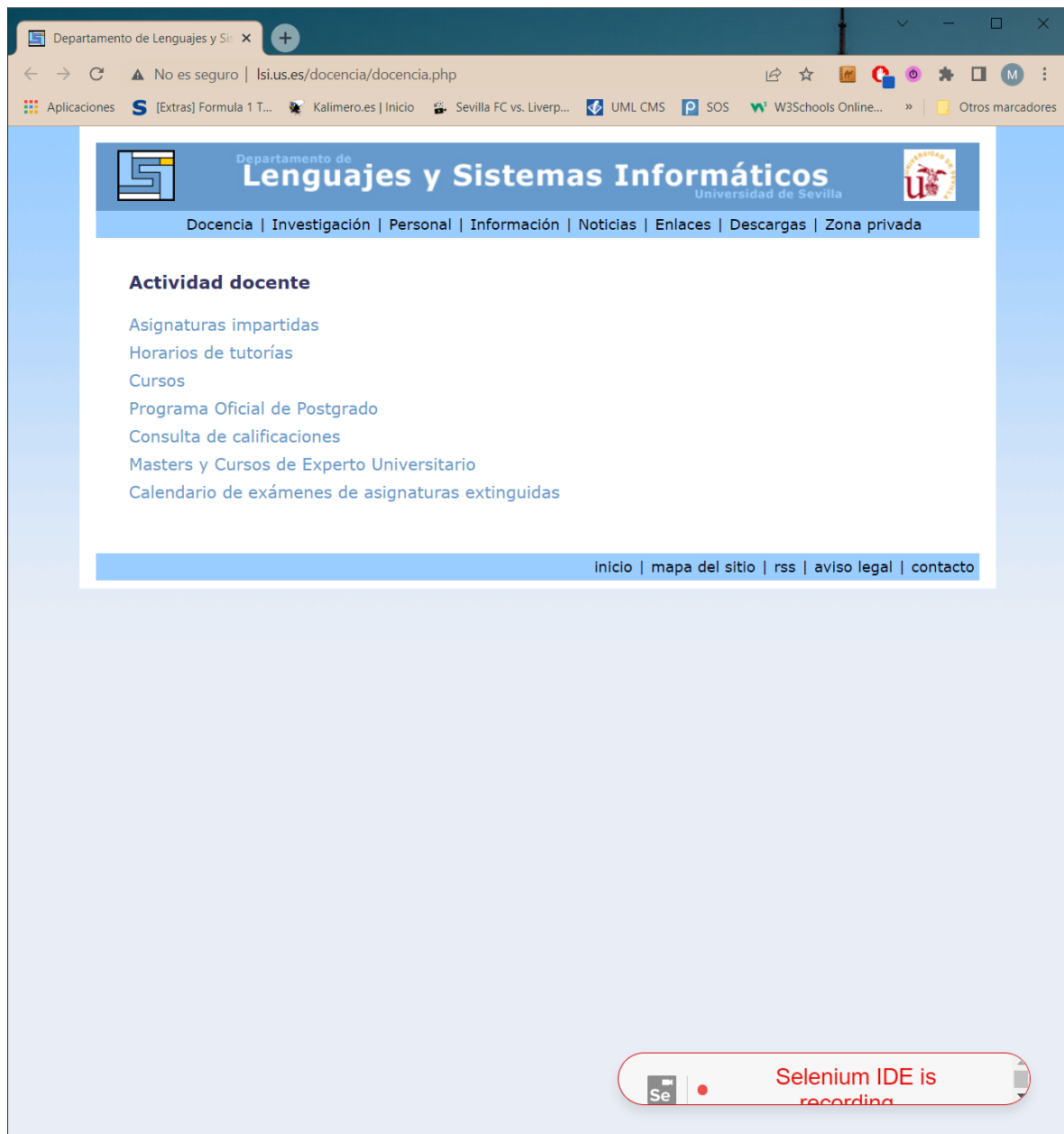


Figura 8.3: IUso Extensión Selenium 2

Aquí hacemos el acto de pinchar sobre la pestaña de docencia.

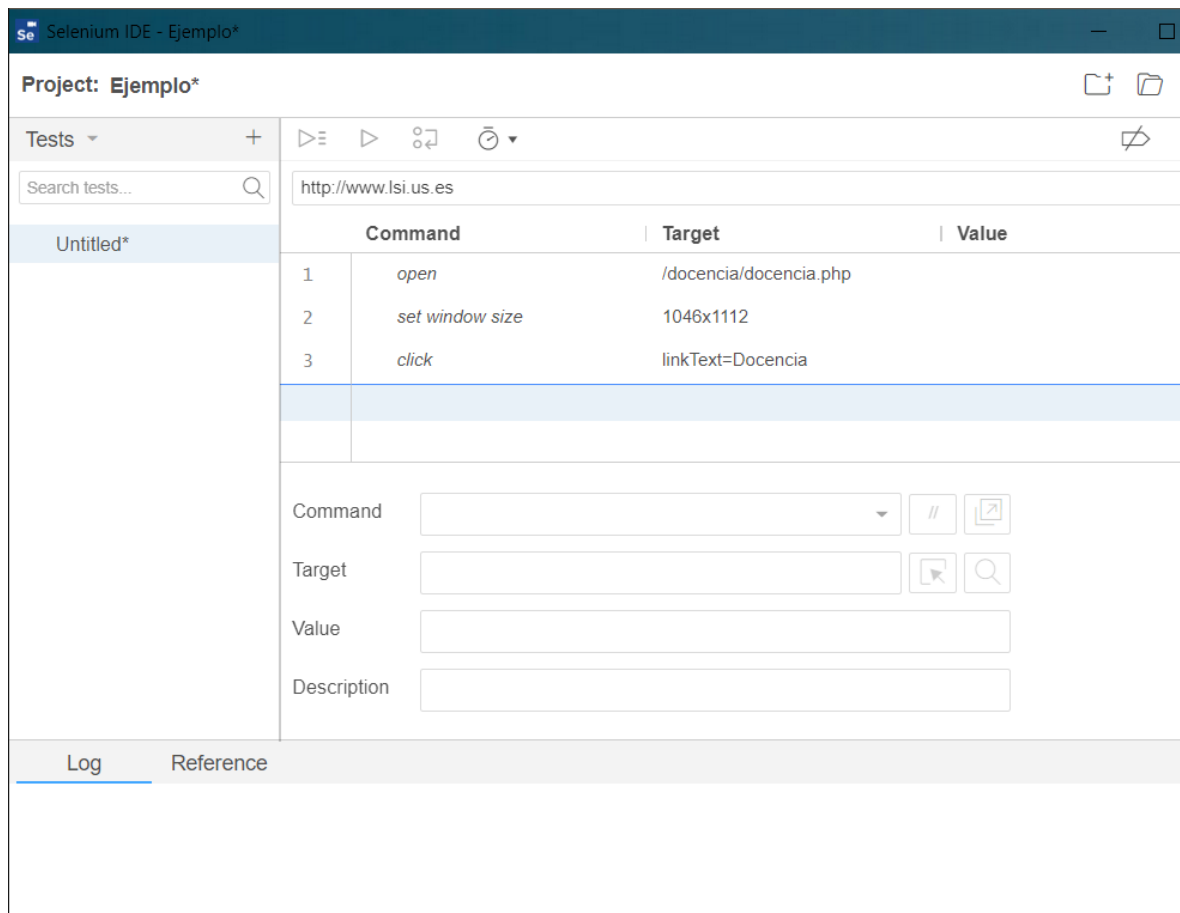


Figura 8.4: Uso Extensión Selenium 3

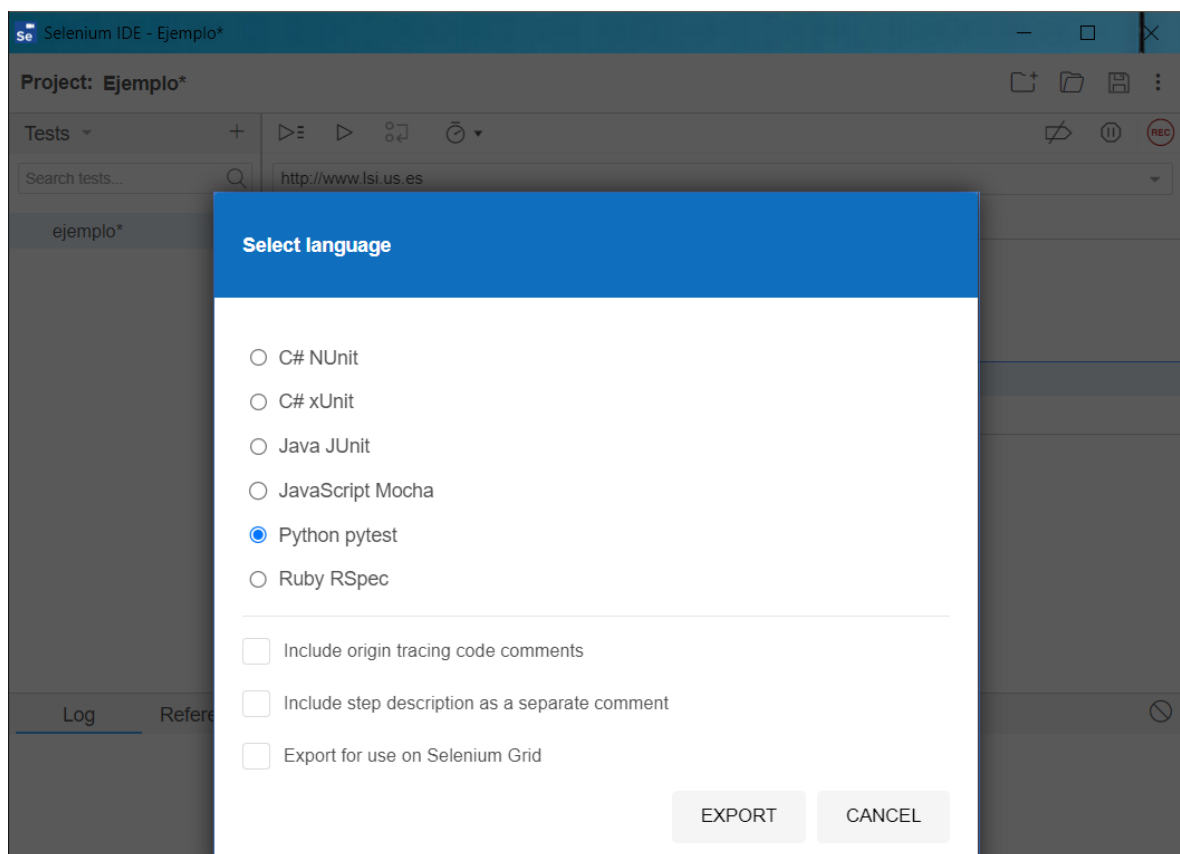


Figura 8.5: Uso Extensión Selenium 4

Explicado ya como vamos a obtener a partir de ahora la información la explicación del código simplemente estaremos refiriéndonos a los objetos que estamos usando, dando por sabido que el método para saber su dirección será uno de los dos anteriormente explicados.

Continuando con el código:

```
driver.find_element(By.XPATH, "//input[@placeholder='Correo_electr nico_o_n mero_de_tel fono']").send_keys(email)
driver.find_element(By.XPATH, "//input[@placeholder='Contrase a']").send_keys(password)
# clicking on login button
#driver.find_element(By.NAME, "login").click()
time.sleep(1.5)
driver.minimize_window()
driver.find_element(By.XPATH, "/html/body/div[1]/div[1]/div[1]/div/div/div/div[2]/div/div[1]/form/div[2]/button").click()
time.sleep(2.5)
```

Extracto de código 8.3: Función Login2

Vemos que rellenamos el formulario de inicio de sesión aparte de esto vemos que le damos la orden a la aplicación de minimizarse, también de que espere 1.5 y 2.5 segundos, aun que obviamente todo tiempo que sumemos hará que el programa se ralentice, pero cada vez que veamos unos de estos en el código tiene una explicación lógica, se necesitan ya que estamos entrando en páginas web que no aparecen instantáneamente sino que tienen un tiempo de carga variable, estos tiempos que el código está esperando es para suplir este problema, ya que sino buscará objetos que aún no han cargado saltará un error.

8.3. Twitter

En la implementación de la parte de Twitter hay dos claves fundamentales, que son la acción de recoger los Tweets de las cuentas y la otra la herramienta que nos permite hacer una Sentimental Analysis, las cuales implementamos aquí.

8.3.1. NLTK

Como se ha explicado en puntos anteriores esta librería será la encargada de varias cosas entre ellas:

StopWords

Las StopWords serán aquellas palabras que cuando analizamos cada Tweet no queremos contabilizar, como pueden ser los artículos, los determinantes, las comas, los puntos etc.

Para poder usar esta funcionalidad la importamos en el código, luego ya que este proyecto se realiza con un lenguaje español tendremos que especificar que las palabras que no queremos usar serán en español ya que el idioma por defecto es el inglés. Por último añadimos las palabras que más se suelen usar que no están normalmente en las stopwords.

```
import nltk
from nltk.corpus import stopwords
nltk.download('vader_lexicon')
nltk.download('stopwords')
nltk.download('punkt')
stopwords_es = set(stopwords.words('spanish'))
stopwords_es.add("http")
stopwords_es.add("https")
```

Extracto de código 8.4: StopWords

Después de esto lo único que tendremos que hacer es a la hora de ir analizando palabra por palabra, que casualmente lo haremos con la misma librería NLTK, discriminaremos si la palabra pertenece a las llamadas Stopwords.

Sentimental Analysis

El análisis de sentimientos se basa en algoritmos y técnicas de aprendizaje automático que analizan el lenguaje y extraen características relevantes para determinar el tono emocional del texto. Estas características pueden incluir el uso de palabras positivas o negativas, la intensidad de las palabras, las estructuras gramaticales utilizadas o incluso el contexto en el que se encuentra el texto.

La librería NLTK nos permite usar el módulo VADER(Valence Aware Dictionary for Sentiment Reasoning). Para poder usarla la importamos y luego simplemente le pasamos como argumento el texto que queremos que en nuestro caso será los Tweets de alguien.

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer

sent_analyzer.polarity_scores("texto")
```

Extracto de código 8.5: Sentimental Analysis

8.3.2. Snsrape

La librería Snscape tiene múltiples módulos pero para nuestro caso solo usaremos el de Twitter. Para ello tendremos que importar la librería y luego hacer una petición con los datos que necesitamos, hacer la búsqueda en sí y finalmente saber como recoger el texto de un Tweet.

```
import snsrape.modules.twitter as sntwitter
query = "(from:"+user+")_until:"+str(now.year)+"-"+str(now.
        month)+"-"+str(now.day)+"_since:2010-01-01"
sntwitter.TwitterSearchScraper(query).get_items()
```

Extracto de código 8.6: Snscape.Twitter

Esto último nos dará la lista de todos los Tweets de la cuenta `user` desde 2010 hasta hoy, para poder recoger el texto plano del Tweet simplemente tendremos que aplicarle a cualesquiera de los objetos de la lista la función `.rawContent`

8.4. Interfaz Gráfica

La forma de implementar la interfaz gráfica será sencilla, tendremos un root que será el que haga el loop de la aplicación y con el que daremos en diseño de la ventana y de ahí se llamará a la página principal y de esa irá derivando a las demás.

```
class PaginaPrincipal
class Pagina1
class Pagina2
class Pagina31
class Pagina3
class Pagina4
class Pagina5

root = Tk()
root.iconbitmap("img/logo_birds.ico")
root.config(width="800", height="600")
mi_app = PaginaPrincipal(root)
root.mainloop()
```

Extracto de código 8.7: Interfaz Gráfica

8.5. Futuros Plugin

Como una de las razones del desarrollo de este proyecto es dejar la posibilidad a que en años posteriores la gente pueda seguir nutriendo al proyecto añadiéndole nuevas posibilidades, redes sociales que nos permitan investigar más aspecto de la vida de las personas, como pueden ser por ejemplo Instagram, LinkedIn o TikTok. Para ello daremos una serie de indicaciones de que tendríamos que hacer para integrar ese nuevo plugin al proyecto ya existente.

Teniendo en cuenta que ya se han realizado los módulos correspondiente a la nueva red social y que por lo tanto ya la hemos podido scrapear.

Lo primero que haremos es crear la colección en la base de datos que nos permita identificar a esta nueva red social que llamaremos por ejemplo Instagram.

```
# Conectarse al servidor de MongoDB
cliente = pymongo.MongoClient("mongodb://localhost:27017/")

# Crear una base de datos llamada "mi_base_de_datos"
mi_base_de_datos = cliente['TFG']

# Crear una colección llamada "Instagram"
Instagram = mi_base_de_datos['Instagram']
```

Extracto de código 8.8: Incluir Instagram a BBDD

Lo segundo que haremos es crear la forma de acceder desde la interfaz gráfica a las funciones creadas. Lo primero tendremos que crear una clase para generar una nueva pantalla, en nuestro caso la llamaremos página 6. Así que en la página principal añadiremos algo así, un botón que llame a una función `abrirpagina6(self)`.

```
self.boton_pagina6 = Button(button_frame, bg=fondo,
                             command=self.abrir_pagina6)
self.boton_pagina6.pack(side="left", padx=50)

def abrir_pagina6(self):
    self.master.withdraw()
    pagina6 = Toplevel(self.master)
    Pagina6(pagina6)
```

Extracto de código 8.9: Incluir Instagram a BBDD

Con esto habremos conseguido crear una página exclusiva para nuestra nueva red social, una vez en la clase de la página y ya habiendo pedido al usuario los datos que quiera investigar, comprobaremos en la base de datos si se encuentra previamente o hay que ir a buscarlo.

Se pueden añadir muchos más detalles de implementación como, por ejemplo para mejorar la estética, pero me he querido centrar en los puntos más claves de la implementación de un nuevo plugin, todos los detalles que se puedan añadir luego son más innecesarios.

9. Resultado

El resultado del proyecto ha desembocado en lo que un inicio buscábamos que era crear una aplicación python que obtenga información personal a través de las redes sociales. Como vamos a hacer un repaso bastante extenso de como funciona la aplicación, este apartado servirá también como guía de uso.

Ahora navegaremos un poco por la interfaz gráfica y veremos lo que hemos obtenido.

9.1. Pagina de inicio

Accedemos a la página principal donde vemos las tres opciones que describíamos en el diseño, que es investigar facebook, Twitter o ambas para exportar en un TXT.



Figura 9.1: github-logo

9.2. Pagina búsqueda de Facebook

Cuando accedemos al apartado de Facebook te pide las credenciales de facebook y la cuenta de la persona que queremos investigar. le daremos al boton de investigala una vez los datos estén rellenos.



Figura 9.2: Apartado de Facebook

9.3. Datos de Facebook

Aquí encontramos los datos de la cuenta que hemos buscado anteriormente.

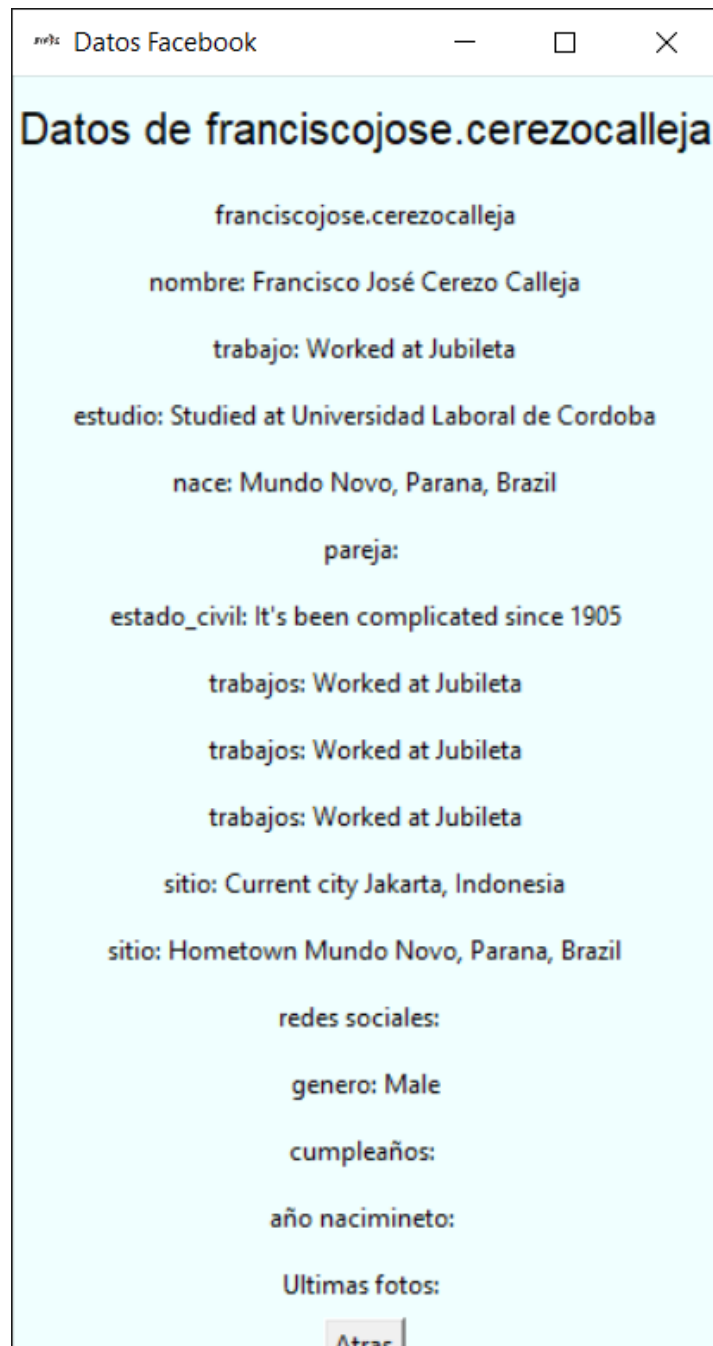


Figura 9.3: Obtención de datos Facebook

9.4. Exportar TXT

Cambiando de pestaña, volvemos a la página principal y entramos en exportar TXT. Aquí indicaremos al igual que la página de facebook los datos necesarios para buscar y también podemos incluir los de Twitter para reunirlos ambos en un archivo txt que se generará en la carpeta en la que tengamos el proyecto.



The screenshot shows a web browser window titled "Exportar TXT". The page has a light blue background and contains the following elements:

- Exportar TXT** (Section Header)
- Tu Correo:** (Text input field)
- Contraseña:** (Text input field)
- Cuenta de Facebook:** (Text input field)
- Cuenta de Twitter:** (Text input field)
- Exportar Información** (Submit button)

Figura 9.4: Exportar TXT

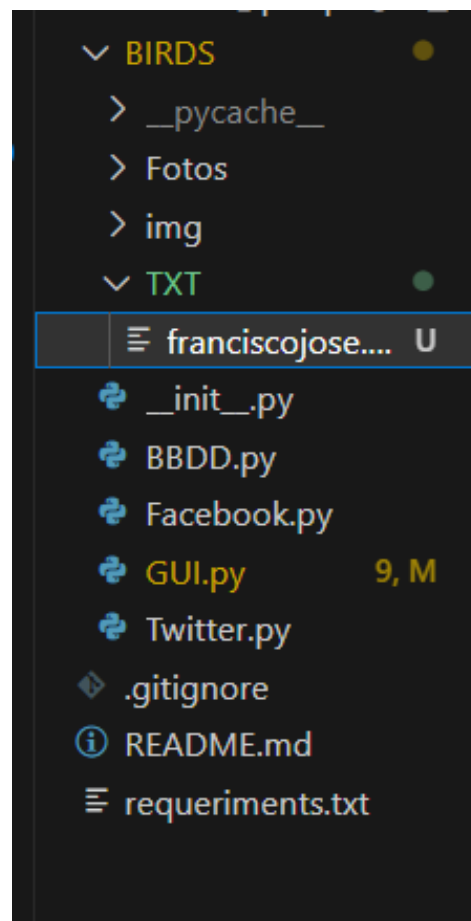


Figura 9.5: Creación del TXT

9.5. Pagina busqueda de Twitter

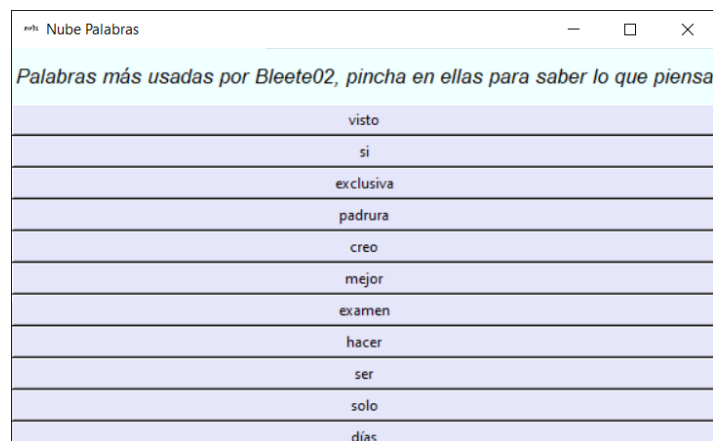


Figura 9.6: Apartado de Twitter



Figura 9.7: Rellenamos con una cuenta

Cuando iniciamos la búsqueda nos devolverá una lista ordenada de mayor a menor uso (top,down) de las palabras que más usa la cuenta investigada. Si pulsamos en



Palabras más usadas por Bleete02, pincha en ellas para saber lo que piensa
visto
si
exclusiva
padrura
creo
mejor
examen
hacer
ser
solo
días

Figura 9.8: Nube de palabras

alguna de las palabras nos hará un estudio de los Tweets en los que usa dicha palabra para finalmente darte el tono con el que suele usar dicha palabra(Negativo, Neutro o Positivo).

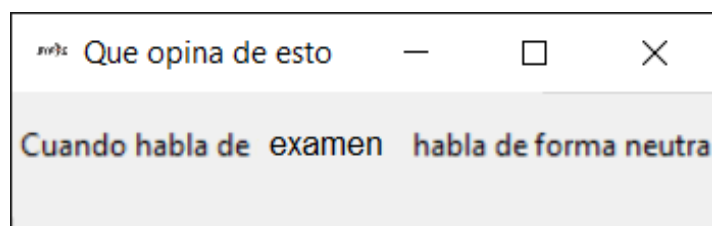


Figura 9.9: Que opina sobre

10. Manual de Instalación

Este proyecto está pensado para un entorno Windows 10/11.

Lo primero que hay que tener instalado es la última versión de Python, la cual se puede descargar e instalar desde [aquí](#).

Lo segundo que tenemos que tener instalado es la base de datos de MongoDB la cual podemos descargar desde [aquí](#). Importante declarar en la descarga el Sistema Operativo del equipo, la versión dejamos la que viene por defecto y descargamos el archivo en formato .msi .

En tercer lugar descargamos el repositorio del proyecto y una vez colocados en su carpeta iniciamos una consola que esté en la misma dirección y ejecutamos el comando:

PIP INSTALL -r requirements.txt

El cuál servirá para instalar todas los módulos y dependencias que se encuentran dentro del código.

Para la ejecución entraremos en la carpeta de BIRDS, y ejecutaremos el siguiente código:

Python GUI.py

11. Conclusiones

11.1. Obtener Información

Una vez terminado el proyecto podemos ver que el obtener la información personal de la gente nunca ha supuesto un problema, la cantidad de datos personal que se pueden ver a simple vista es más que sorprendente.

Este proyecto no se ha centrado nunca en obtener datos demasiado profundos o difíciles de obtener, su funcionalidad ha sido la de recabar, organizar y presentar la más superficial de los datos en las redes sociales, y aún así podemos ver la cantidad de datos que podemos obtener con esta herramienta.

11.2. Ética

El proyecto plantea cuestiones éticas importantes que deben ser consideradas cuidadosamente. La extracción de información personal en redes sociales plantean preocupaciones sobre el respeto a la privacidad y el consentimiento de los usuarios.

¿El hecho que los usuarios publiquemos datos accesibles ya sea por un grupo restringido de personas o públicamente le da derechos a un tercero a obtenerlos y tratarlos como quiera? Hemos visto en este proyecto que este es un punto que ni las políticas de uso de las redes sociales estudiadas ni la legislación española toman en consideración, puede que este sea uno de los puntos en los que la normativa tenga que avanzar para poder evitar posibles malos usos.

La transparencia también juega un papel fundamental en la ética del proyecto. Debe ser claro y transparente en relación con los métodos utilizados para obtener los datos, así como en el almacenamiento y gestión de los mismos. Esto no supone un problema real ya que al ser un proyecto *OPEN SOURCE* se puede ver directamente el código y lo que hace en cada una de las líneas.

11.3. Concienciación

La población en general cree que el uso de las redes sociales son gratuitas, pero nada más lejos de la realidad, los datos que le damos y que solo algunos aquí obtenemos es realmente el precio a pagar. Espero que para toda persona que haya podido leer este proyecto haya respetado más su propia intimidad y por lo menos sea consciente de donde están sus datos y como de públicos son, que sin ánimo que las personas dejen de usar redes sociales espero que sean así más consecuentes al menos.

11.4. Malas Praxis

No es nuevo que el espionaje por la redes sociales o Stalkear, no es nada nuevo, no son pocas las noticias que se publican donde esclarecen que los robos a domicilios se disparan cuando pueden saber por redes sociales y la familia se ha ido de vacaciones por ejemplo. Quiero dejar claro entonces que para nada es este el fin del proyecto y por lo tanto no me responsabilizo de los malos usos que a esta herramienta se les pueda dar.

11.5. Objetivos

Una vez acabado el TFG, creo que los objetivos propuestos inicialmente se han cumplido con creces, ya que salgo de aquí con unos conocimientos muy amplios sobre técnicas como el web scrapping, como la creación de interfaces gráficas o la creación y gestión de bases de datos no relacionales, técnicas que si bien algunas podemos aprenderlas en el grado, no han sido posible en mi caso.

12. Planes Futuros

Soy plenamente consciente que el proyecto tiene variedad de fallos o cosas que podrían estar bastante mejor realizadas, pero tampoco era este el fin de este TFG.

Una de las ideas que ayudó a tomar este proyecto como TFG fue dejar la posibilidad que otros alumnos que vengan en años posteriores y lo mejoren, ya sea perfeccionando la seguridad de la aplicación como poder seguir añadiendo futuras redes sociales como pueden ser Instagram, LinkedIn, TikTok ... Para que en un esta pueda ser una herramienta aún más interesante.

13. Bibliografía

- [1] Shweta A. Gode Anand V. Saurkar, Kedar G. Pathare. An overview on web scraping techniques and tools, 2018. URL <https://www.ijfrcsce.org/index.php/ijfrcsce/article/view/1529>.
- [2] Software Freedom Conservancy. Selenium documentation, 2023. URL <https://www.selenium.dev/documentation/>.
- [3] X Corp. Términos de servicio de twitter, 2023. URL <https://twitter.com/es/tos>.
- [4] Junta de Andalucía. Informe final sobre la consulta preliminar del mercado “perfiles profesionales Ámbito informático”, 2018. URL <https://www.juntadeandalucia.es/contratacion/document/download?refCode=2018-0000038357&refDoc=2018-0000038357-2>.
- [5] Python Software Foundation. Graphical user interfaces with tk, 2023. URL <https://docs.python.org/3/library/tk.html>.
- [6] Las Cortes Generales. Ley orgánica 3/2018, de 5 de diciembre, de protección de datos personales y garantía de los derechos digitales., 2018. URL <https://www.boe.es/buscar/act.php?id=B0E-A-2018-16673>.
- [7] JustAnotherArchivist. Git de la librería snsrape, 2020. URL <https://github.com/JustAnotherArchivist/snsrape>.
- [8] Alistair Cockburn Ward Cunningham Martin Fowler James Grenning Jim Highsmith Andrew Hunt Ron Jeffries Jon Kern Brian Marick Robert C. Martin Steve Mellor Ken Schwaber Jeff Sutherland Dave Thomas Kent Beck Mike Beedle, Arie van Bennekum. Manifiesto Ágil, 2001. URL <https://agilemanifesto.org/iso/es/manifesto.html>.
- [9] Meta. Términos de servicio de facebook, 2023. URL <https://www.facebook.com/legal/terms>.
- [10] Inc. MongoDB. MongoDB documentation, 2023. URL <https://www.mongodb.com/docs/>.
- [11] OpenAI, 2018. URL <https://openai.com/>.
- [12] NLTK Project. Nltk documentation, 2023. URL <https://www.nltk.org/>.