



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Ingeniería Informática - Tecnologías Informáticas

OsintSpector: Recopilador de información pública online sobre personas

**Realizado por
Juan Luis Aguilera Rivero
DNI 4xxxxxxxR**

**Dirigido por
Ángel Jesús Varela Vaca**

**Departamento
Lenguajes y Sistemas Informáticos**

Sevilla, (06/2023)

Resumen

El proyecto busca recopilar de manera segura y legal información pública disponible en Internet sobre una persona objetivo dada por el cliente. Utiliza dos métodos de búsqueda: a través de la web y analizando su cuenta de Twitter. Al proporcionar datos previos conocidos, se generan informes completos que abarcan distintas secciones relacionadas con la información suministrada, generando a su vez inteligencia a partir de dichos datos recopilados. El proyecto garantiza la confidencialidad y accesibilidad del usuario y no almacena datos personales.

Palabras clave: OSINT, SOCMINT, Información, Inteligencia.

The project aims to securely and legally collect publicly available information on the Internet about a target person given by the client. It uses two search methods: via the web and by analysing their Twitter account. By providing known background data, comprehensive reports are generated covering different sections related to the information provided, generating intelligence from the collected data. The project guarantees user confidentiality and accessibility and doesn't store personal data.

Keywords: OSINT, SOCMINT, Information, Intelligence.

Agradecimientos

En primer lugar, me gustaría agradecer a mi tutor Ángel Jesús Varela, por haberme guiado durante el desarrollo del proyecto y sobre todo por la amabilidad y paciencia con la que siempre me ha tratado en todo este trayecto.

También me gustaría agradecer a mis amigos tanto los que están como los que ya no, ya que de una forma u otra me han ayudado en ciertas etapas de la carrera.

Por último y lo más importante, a mi familia, la cual siempre me estuvo apoyando desde los inicios de entrar en la vida universitaria, por todo el esfuerzo y amor que me han brindado para poder llegar aquí.

Sobre todo, dedicárselo a mi abuelo Rafael, el cual desde pequeño me aportó valores muy importantes en esta vida, como la dedicación y el esfuerzo a la hora de trabajar para conseguir metas, ayudar siempre a todo aquel que lo necesite y siempre ser humildes en esta vida, para así saber en todo momento de donde venimos y quiénes somos. Descansa en paz abuelo.

Índice general

Índice general	V
Índice de tablas	VII
Índice de figuras	XI
Índice de código	XV
1 Introducción	1
1.1 Contexto y motivación	2
1.2 Objetivos	2
2 OSINT	5
2.1 Orígenes	5
2.2 Definición	6
2.3 Retos y oportunidades	7
2.3.1 Retos	7
2.3.2 Oportunidades	7
2.4 El ciclo de inteligencia	9
2.5 Legalidad	10
3 Planificación y costes	13
3.1 Análisis temporal	13
3.2 Diagrama de Gantt	14
3.3 Análisis de Costes	19
3.3.1 Costes directos	19
3.3.2 Costes reales	19
3.3.3 Costes indirectos	20
3.3.4 Costes totales	22
4 Elicitación del problema	23
4.1 Introducción	23
4.1.1 Alcance	23
4.1.2 Actores del sistema	23

4.2	Requisitos Específicos	25
4.2.1	Requisitos funcionales	25
4.2.2	Requisitos de información	38
4.2.3	Requisitos de negocio	49
4.2.4	Requisitos no funcionales	51
5	Análisis y diseño	53
5.1	Modelo de casos de uso	54
5.2	Descripción de diagramas de secuencia	63
5.3	Modelo de datos	66
5.4	Arquitectura del sistema	68
6	Implementación y pruebas	71
6.1	Arquitectura tecnológica	71
6.1.1	Tecnologías usadas en el front-end	71
6.1.2	Tecnologías usadas en el back-end	74
6.2	Herramientas adicionales	75
6.3	Esquema tecnológico	76
6.4	Detalles de la implementación	77
6.4.1	Estructura de paquetes	77
6.4.2	Código desarrollado	80
6.4.3	Vistas	87
6.5	Pruebas	118
6.5.1	Pruebas de aceptación	119
6.5.2	Intrusión	125
7	Conclusiones	127
7.1	Retrospectiva	127
7.1.1	Aspectos a repetir	127
7.1.2	Aspectos a mejorar o evitar	127
7.2	Lecciones aprendidas	128
7.3	Posibles mejoras del sistema	129
8	Manuales	131
8.1	Manual de instalación y despliegue	131
8.1.1	Tecnologías a descargar previa a la instalación	131
8.1.2	Instalación	132
8.2	Manual de usuario	134
8.2.1	Documentación previa a las funciones principales	134
8.2.2	Análisis de Twitter	137
8.2.3	Búsqueda de personas	147
	Bibliografía	167

Índice de tablas

3.1	Estimación temporal del proyecto.	14
3.2	Salarios de las posiciones en el proyecto.	19
3.3	Costes directos estimados.	19
3.4	Costes reales por tarea.	20
3.5	Costes del hardware utilizado en el proyecto.	21
3.6	Costes del software utilizado en el proyecto.	21
4.1	Usuario	24
4.2	Sistema	24
4.3	Información previa a las búsquedas.	25
4.4	Petición Twitter.	25
4.5	Análisis de Twitter.	26
4.6	Extracción de datos para la nube de palabras.	26
4.7	Elaboración y visualización de la nube de palabras.	26
4.8	Extracción de datos para el análisis de sentimientos.	27
4.9	Elaboración y visualización de la nube de palabras.	27
4.10	Extracción de datos para el grafo de interacciones.	27
4.11	Elaboración y visualización del grafo de interacciones.	28
4.12	Extracción de datos para el grafo de comunidades.	28
4.13	Elaboración y visualización del grafo de comunidades.	29
4.14	Extracción de datos para el estudio de localizaciones.	29
4.15	Elaboración y visualización del estudio de localizaciones.	30
4.16	Descarga de la información generada en el análisis de Twitter.	30
4.17	Petición búsqueda persona.	30
4.18	Búsqueda de persona.	31
4.19	Extracción de datos del INE sobre nombre y/o apellido(s).	31
4.20	Elaboración y visualización de los datos del INE sobre nombre y/o apellido(s).	32
4.21	Extracción de datos sobre el nombre y/o apellido(s) y ciudad en motores de búsqueda.	32
4.22	Elaboración y visualización de los datos sobre el nombre y/o apellido(s) y ciudad en motores de búsqueda.	33

4.23 Extracción de datos sobre el nickname en motores de búsqueda.	33
4.24 Elaboración y visualización de los datos sobre el nickname en motores de búsqueda.	34
4.25 Extracción de datos sobre el correo electrónico en motores de búsqueda.	34
4.26 Elaboración y visualización de los datos sobre el correo electrónico en motores de búsqueda.	35
4.27 Extracción de datos sobre el número de teléfono en motores de búsqueda.	35
4.28 Elaboración y visualización de los datos sobre el número de teléfono en motores de búsqueda.	36
4.29 Extracción de datos en la Darknet sobre los atributos rellenados.	36
4.30 Elaboración y visualización de los datos de la Darknet sobre los atributos rellenados.	37
4.31 Descarga de la información generada en la búsqueda de la persona.	37
4.32 Proxies.	38
4.33 User-Agents.	38
4.34 Motores Nickname.	39
4.35 API keys.	40
4.36 Username Twitter.	40
4.37 Recopilación Twitter.	41
4.38 Wordcloud Twitter.	41
4.39 Sentimientos Twitter.	42
4.40 Interacciones Twitter.	42
4.41 Comunidades Twitter.	43
4.42 Información ordenada e inteligencia sobre el usuario de Twitter para el apartado de localizaciones.	43
4.43 Persona.	44
4.44 Recopilación Nombre-Apellido(s)-Ciudad.	44
4.45 Inteligencia Nombre-Apellido(s)-Ciudad.	45
4.46 Recopilación Nickname.	45
4.47 Inteligencia Nickname.	46
4.48 Recopilación e-mail.	46
4.49 Inteligencia e-mail	47
4.50 Recopilación Teléfono.	47
4.51 Inteligencia Teléfono.	48
4.52 Recopilación Darknet.	48
4.53 Búsqueda trivial de personas.	49
4.54 Búsqueda usuarios no válidos en Twitter.	49
4.55 Atributos de búsqueda de personas opcionales.	50
4.56 Diseño responsive y simple.	51

4.57	Recopilación de proxies rotatorios.	51
4.58	Ayuda en el anonimato del usuario.	51
5.1	Información previa a las búsquedas.	55
5.2	Información sobre el desarrollador.	55
5.3	Visualización de la nube de palabras.	56
5.4	Visualización sobre el análisis de sentimientos.	56
5.5	Visualización del grafo de interacciones.	57
5.6	Visualización del grafo de comunidades.	57
5.7	Visualización de la recopilación de localizaciones.	58
5.8	Descarga de la información generada.	58
5.9	Visualización de los datos sobre el nombre y/o apellido(s) del INE.	59
5.10	Visualización de los datos sobre el nombre y/o apellido(s) en motores de búsqueda.	59
5.11	Visualización de los datos sobre el nickname en motores de búsqueda.	60
5.12	Visualización de los datos sobre el correo electrónico en motores de búsqueda.	60
5.13	Visualización de los datos sobre el número de teléfono en motores de búsqueda.	61
5.14	Visualización de los datos indexados en la Darknet sobre los atributos rellenados.	61
5.15	Descarga de la información resultante de la persona.	62
6.1	Prueba sobre información previa a las búsquedas.	119
6.2	Prueba sobre información sobre el desarrollador.	119
6.3	Prueba sobre la nube de palabras.	120
6.4	Prueba sobre el análisis de sentimientos.	120
6.5	Prueba sobre el grafo de interacciones.	121
6.6	Prueba sobre el grafo de comunidades.	121
6.7	Prueba sobre el grafo de la recopilación de localizaciones.	122
6.8	Prueba sobre la descarga de información generada de Twitter.	122
6.9	Prueba sobre los datos recogidos del INE a partir del nom- bre y/o apellido(s).	123
6.10	Prueba sobre los datos recogidos del nombre y/o apellido(s) y ciudad en motores de búsqueda.	123
6.11	Prueba sobre los datos recogidos del nickname en motores de búsqueda.	124
6.12	Prueba sobre los datos recogidos del correo electrónico en motores de búsqueda.	124
6.13	Prueba sobre los datos recogidos del número teléfono en motores de búsqueda.	124

6.14 Prueba sobre los datos indexados en la Darknet sobre los atributos rellenos.	125
6.15 Prueba sobre la descarga de la información resultante de la persona.	125

Índice de figuras

2.1	Ejemplo actividades realizadas en Internet en 1 minuto en 2021.	9
3.1	Diagrama de Gantt completo.	15
3.2	Vista paquete planificación.	15
3.3	Vista paquete análisis y diseño.	16
3.4	Vista paquete ejecución.	17
3.5	Vista paquete seguimiento y control.	17
3.6	Vista paquete cierre.	18
5.1	Diagrama casos de uso.	54
5.2	Diagrama de secuencia para el análisis de Twitter.	64
5.3	Diagrama de secuencia para la búsqueda y recopilación de información de una persona objetivo.	65
5.4	Modelo de datos.	66
5.5	Vista análisis de Twitter.	67
5.6	Vista búsqueda de Persona.	67
5.7	Arquitectura del sistema.	69
6.1	HTML5.	72
6.2	CSS3.	72
6.3	Javascript.	72
6.4	Jinja2.	72
6.5	Bootstrap.	73
6.6	JQuery.	73
6.7	ApexCharts.	73
6.8	ChartJS.	73
6.9	Vis-NetworkJS.	74
6.10	Luxon.	74
6.11	Leaflet.	74
6.12	Python.	75
6.13	Esquema de tecnologías usadas.	76
6.14	Vista global de los paquetes.	77

6.15	Paquete searchScripts.	78
6.16	Paquete utils.	78
6.17	Paquete templates.	79
6.18	Paquete static.	79
6.19	Clase Análisis de Twitter.	80
6.20	Inicialización clase Sentimental Analysis.	82
6.21	Proceso de análisis de sentimientos de cada tweet.	82
6.22	Clase Grafo comunidades.	83
6.23	Clase del gestor de proxies.	84
6.24	Módulo para la recopilación de información en la Darknet.	85
6.25	Módulo para la recopilación de información en la Darknet.	85
6.26	Distintas plantillas base de OsintSpector.	86
6.27	Error Scraping solicitud de búsqueda HIBP.	87
6.28	Error Scraping 403 HIBP.	87
6.29	Página principal.	88
6.30	Página de sobre nosotros.	89
6.31	Página de documentación previa a las búsquedas.	90
6.32	Páginas del formulario análisis de Twitter.	91
6.33	Opción descarga información generada Twitter.	92
6.34	JSON generado.	92
6.35	Wordcloud del Twitter de Pedro Sánchez.	93
6.36	Resultado análisis de sentimientos del Twitter de Pedro Sánchez	94
6.37	Resultado grafo TOP interacciones del Twitter de Pedro Sánchez	95
6.38	Resultado grafo de comunidades del Twitter de Pedro Sánchez	96
6.39	Resultado grafo de comunidades del Twitter de Pedro Sánchez	97
6.40	Leyenda explicativa sobre los valores de la tabla de comunidades	98
6.41	Tabla de recopilación de todas las localizaciones y fechas.	99
6.42	Mapamundi con todas las localizaciones.	100
6.43	Formulario búsqueda de personas.	101
6.44	Descarga resultados de la búsqueda del objetivo.	102
6.45	Resultados búsqueda de datos del INE sobre el nombre y/o apellido(s).	103
6.46	Resultado búsqueda de datos sobre el nombre y/o apellido(s) y ciudad.	104
6.47	Gráficas sobre datos recopilados del nombre y/o apellido(s) y ciudad.	105
6.48	Resultado de la Darknet sobre el nombre y/o apellido(s).	106
6.49	Resultado búsqueda de datos sobre el nickname.	107
6.50	Resultado de la Darknet sobre el nickname.	108
6.51	Dashboard de resultados de IntelX sobre el correo.	109
6.52	Dashboard de resultados de HIBP sobre el correo.	110

6.53	Resultado búsqueda de datos sobre el correo en IntelX. . . .	111
6.54	Resultado búsqueda de datos sobre el correo en HIBP. . . .	112
6.55	Resultado de la Darknet sobre el email.	113
6.56	Dashboard de resultados de IntelX sobre el número de teléfono.	114
6.57	Dashboard de resultados de HIBP sobre el número de teléfono.	115
6.58	Resultado búsqueda de datos sobre el número de teléfono en IntelX.	116
6.59	Resultado búsqueda de datos sobre el número de teléfono en HIBP.	117
6.60	Resultado de la Darknet sobre el número de teléfono. . . .	118
8.1	Clonación del repositorio.	132
8.2	Creación entorno virtual y descarga de paquetes.	132
8.3	Todos los paquetes instalados finalmente.	132
8.4	Variables de entorno.	133
8.5	Página principal.	134
8.6	Página de documentación previa a las búsquedas.	135
8.7	Página de sobre nosotros.	136
8.8	Navegación a las funciones principales.	137
8.9	Página del formulario análisis de Twitter.	138
8.10	Opción descarga información generada Twitter.	139
8.11	Wordcloud de Twitter.	140
8.12	Resultado análisis de sentimientos de Twitter.	141
8.13	Resultado grafo TOP de interacciones.	142
8.14	Resultado grafo de comunidades.	143
8.15	Resultado tabla de valores de las comunidades.	144
8.16	Leyenda explicativa sobre los valores de la tabla de comu- nidades.	145
8.17	Tabla de recopilación de todas las localizaciones y fechas. .	146
8.18	Mapamundi con todas las localizaciones.	147
8.19	Formulario búsqueda de personas.	148
8.20	Descarga resultados de la búsqueda del objetivo.	149
8.21	Resultados búsqueda de datos del INE sobre el nombre y/o apellido(s).	150
8.22	Resultado búsqueda de datos sobre el nombre y/o apelli- do(s) y ciudad.	151
8.23	Gráficas sobre datos recopilados del nombre y/o apellido(s) y ciudad.	152
8.24	Resultado de la Darknet sobre el nombre y/o apellido(s). .	153
8.25	Resultado búsqueda de datos sobre el nickname.	154
8.26	Resultado de la Darknet sobre el nickname.	155
8.27	Dashboard de resultados de IntelX sobre el correo.	156
8.28	Dashboard de resultados de HIBP sobre el correo.	157
8.29	Resultado búsqueda de datos sobre el correo en IntelX. . . .	158

8.30	Resultado búsqueda de datos sobre el correo en HIBP. . . .	159
8.31	Resultado de la Darknet sobre el email.	160
8.32	Dashboard de resultados de IntelX sobre el número de teléfono.	161
8.33	Dashboard de resultados de HIBP sobre el número de teléfono.	162
8.34	Resultado búsqueda de datos sobre el número de teléfono en IntelX.	163
8.35	Resultado búsqueda de datos sobre el número de teléfono en HIBP.	164
8.36	Resultado de la Darknet sobre el número de teléfono. . . .	165

Índice de código

CAPÍTULO 1

Introducción

Hoy en día gracias a Internet tenemos una gran cantidad de información accesible y de cualquier tipo gracias a múltiples servicios disponibles en la red. Esto quiere decir que es posible encontrar información pública y casi siempre gratuita sobre las propias personas, esta información puede ser publicada en Internet tanto de forma voluntaria por la misma persona, como sucede en el caso de las redes sociales o blogs, o la persona ha dado consentimiento a algún tercero para la publicación de información suya, siempre que no se vulnere el derecho al honor o la intimidad del afectado como dicta la ley Orgánica BOE-A-1982-11196, de 5 de mayo [16], o vaya en contra de la protección de datos como dicta la ley Orgánica BOE-A-2018-16673, de 5 de diciembre [17]. Así mismo los datos publicados en documentos oficiales también pueden ser tratados por terceros sin el consentimiento explícito de su titular.

Esto quiere decir que existen fuentes abiertas las cuales se pueden consultar para poder obtener información sobre cualquier persona deseada, este proceso de recopilación de obtención de fuentes abiertas y generación de inteligencia a partir de ellas se llama Open Source Intelligence (OSINT), la cual se explica más adelante con profundidad (ver Capítulo 2).

Por ello, se crea la necesidad de que cualquiera pueda consultar que información personal y de que tipo hay pública en Internet, o el nivel al que está expuesta una persona debido a que se hayan filtrado datos sensibles suyos. Por tanto, a partir de esta necesidad nace la idea de crear una herramienta que pueda unificar y recopilar los datos que existen sobre cierta persona en Internet. Además, que a partir de esta información base se pueda generar otros tipos de información más compleja, es decir, trans-

formarlos en inteligencia. Esta idea será el eje central de *OsintSpector*.

1.1– Contexto y motivación

Como anteriormente se comentó, se crea la necesidad de poder saber que información personal y que nivel de sensibilidad hay expuesta en Internet sobre uno mismo.

En España hay un altísimo porcentaje de gente que usa Internet, pero este porcentaje no es tan satisfactorio cuando estudiamos que tipo de nociones/conocimientos informáticos tienen los habitantes de nuestro país, como refleja este estudio del INE [18] y su informe metodológico [19] que trata sobre las habilidades digitales entre los españoles de 16 a 74 años. En resumen, esto significa que hay mucha gente que publica, transmite o deja información suya, es decir, crean su huella digital en las redes sin conocimiento de causa o peligro del nivel de exposición al que pueden llegar. Además, puede que existan personas con habilidades digitales avanzadas, en este caso según el estudio de 2021 solo un 36,1 % de la población española, las cuales no sepan cómo buscar e investigar con exactitud acerca de su huella digital o de posibles filtraciones de sus datos sensibles por culpa de terceros. Incluso, de poder buscar información suya en redes no superficiales, como por ejemplo, la red profunda de Tor [36].

A su vez, también existe la motivación de la creación de una herramienta desde el lado contrario a esta situación, es decir, a que cualquier persona con cualquier conocimiento en informática pueda consultar la huella digital de cualquier otra. Esto se puede realizar mediante un proceso de generación de inteligencia por parte de este proyecto, el cual, se basa principalmente en seguir pautas y procesos de OSINT y que muestre de una manera clara toda esta información útil generada para un usuario cliente.

1.2– Objetivos

El objetivo de este trabajo fin de grado es desarrollar una aplicación, *OsintSpector*, pública, gratuita y sencilla de manejar por cualquier usuario con la finalidad de poder encontrar toda información pública y sensible que haya expuesta en Internet o redes profundas sobre una persona objetivo. No sólo se debe pensar en el nombre y apellidos cuando se refiere a una persona, sino también a identidades digitales, atributos o formas de especificación e identificación de una; por ejemplo, ubicación, nickname en Internet, correo electrónico, número de teléfono, redes sociales, etc. Por tanto podemos definir los siguientes objetivos de cara al proyecto final:

- Investigación sobre información pública en Internet sobre una persona, la cual consistirá en:
 - Recopilación de datos dada a priori toda la información posible de la persona (nombre, apellidos, ubicación, nickname, correo electrónico, n^o de teléfono, posibilidad de buscar en la red oscura, etc.).
 - Procesamiento, filtrado y análisis de dicha recolección de datos. Para así descartar información no útil y a su vez generar conocimiento relevante de cara al usuario.
 - Visualización clara, sencilla y compacta de todo este conocimiento para que cualquier usuario con cualquier nivel pueda entenderlo.
- Análisis sobre una cuenta de Twitter dado su username, el cual consistirá en:
 - Generación de nubes de palabras (Wordclouds) a través de sus tweets.
 - Análisis de sentimientos sobre cada tweet.
 - Creación de un grafo sobre el top de usuarios que interactúan con el objetivo.
 - Generación de un grafo y estudio de comunidades que forma parte el objetivo.
 - Recopilación de todas las ubicaciones y fechas donde tuiteó el objetivo.
- El proyecto debe ser una aplicación, un sistema o herramienta que cualquier tipo de usuario pueda entender su funcionamiento, quitando posibles pasos que dificulten el acceso a ello.
- Cada búsqueda será lo más anónima posible, sin tener que dar el usuario ninguna información personal. Además, todo proceso que se genere en el sistema no se guardará nada una vez el usuario deje de usarlo.

CAPÍTULO 2

OSINT

En este capítulo hablaremos sobre la metodología de Open Source Intelligence (OSINT), análisis de información en fuentes abiertas en español, para así tener una visión más amplia sobre esta disciplina, la cual es la base y metodología de todo este proyecto. En este capítulo se utilizará de guía el libro llamado *Investigar personas e identidades en Internet* [9] escrito por Carlos Seisdedos y Vicente Aguilera.

2.1— Orígenes

A pesar de que el término OSINT ha cobrado especial relevancia en los últimos tiempos, su origen se asocia al ámbito militar y se remonta a los años de la Segunda Guerra Mundial, en los que el FBIS (Foreign Broadcast Information Service) creado en 1941 por Estados Unidos utilizaba las fuentes abiertas, como seguramente también utilizarían los servicios de otros países, para obtener información mediante la monitorización y traducción de medios de comunicación de forma que le proporcionara una ventaja militar sobre el adversario. Finalmente, debido a recortes presupuestarios el FBIS estuvo a punto de ser clausurado, hasta el último momento donde científicos hicieron una fuerte campaña de no cerrar esta institución, ya que, según ellos decían que se describía al FBIS como “*el dinero mejor invertido por la comunidad de inteligencia estadounidense*”.

En 2009 la CIA desclasificó un interesante documento elaborado en 1969 en el que se presenta la historia del FBIS [34]. El ochenta por ciento de la información utilizada para monitorizar el colapso de la Unión Soviética se ha atribuido a fuentes abiertas.

Los líderes en la comunidad de inteligencia americana reconocieron que los desafíos y la dinámica del siglo XXI provocaría que las técnicas

OSINT serían más necesarias que nunca debido también al uso de ordenadores personales, el almacenamiento digital, los motores de búsqueda y las redes de comunicación de banda ancha, ya que fueron capaces de percibir que todos estos factores conducirían a un crecimiento exponencial en la información.

Actualmente la utilidad del OSINT va mucho más allá del ámbito militar, siendo explotada actualmente por sectores y actores muy diversos de nuestra sociedad. El poder no está en la información, sino en la inteligencia que se puede derivar de ella.

2.2— Definición

No hay una definición única establecida sobre el OSINT, pero se utilizará la definición del Departamento de Defensa de Estados Unidos, ya que como comentamos antes, el origen de esta disciplina viene dada a la inteligencia militar de este departamento. El documento del Departamento de Defensa de EE.UU [30] define al OSINT como *“Open-source intelligence (OSINT) is intelligence that is produced from publicly available information and is collected, exploited, and disseminated in a timely manner to an appropriate audience for the purpose of addressing a specific intelligence requirement”*.

Desarrollando más a fondo cada frase de la anterior definición:

- *Open-source intelligence (OSINT) is intelligence that is produced from publicly available information.* Efectivamente, para ser considerada OSINT, la inteligencia debe ser generada a partir de información disponible de forma pública. Es lo que se conoce como fuentes abiertas, en contraposición a las fuentes cerradas (asociadas a información clasificada o confidencial). Las fuentes abiertas son muy diversas e incluyen entre otras, Internet (motores de búsqueda, redes sociales, foros, darkweb..), medios de comunicación tradicionales, publicaciones especializadas, fotografías, etc. Un error muy común es que creer que toda fuente abierta es gratuita, y aunque sea así en la mayoría de casos también puede que se requiere cierto pago previo, obviamente toda esta información debe ser accesible de forma legal.
- *... and is collected, exploited, and disseminated in a timely manner to an appropriate audience.* Como se detallará más adelante al abordar el ciclo de inteligencia, la información adquirida debe seguir un proceso y ha de ser tratada para que permita al analista su interpretación y la posterior generación de inteligencia procesable. Por ello la inteligencia generada ha de ser suministrada al decisor (el

destinatario, y quien debe hacer uso de dicha información) y se ha de facilitar de manera oportuna. Es decir, hay que tener presente que un producto de inteligencia puede tener gran calidad, pero si es comunicado fuera de la ventana de tiempo de interés, carecerá de utilidad. Por otro lado, se debe conocer el medio de comunicación donde se transmite dicha información, de forma que se garantice la confidencialidad del mismo.

- *...for the purpose of addressing a specific intelligence requirement.* Finalmente, hay que tener claro cuál es el objetivo final del OSINT: dar respuesta a un requerimiento específico de inteligencia. Por tanto, el primer paso será qué requerimientos/intereses hay que cubrir (el decisor será el que dicte dichos requerimientos), entenderlos sin duda alguna, conocer su prioridad y los destinatarios del producto que se genere, de forma que cualquier actuación dentro de la investigación debe ir encaminada exclusivamente a la consecución de este objetivo.

2.3— Retos y oportunidades

Aunque las fuentes abiertas siempre han sido utilizadas en la comunidad de la inteligencia, la evolución de la tecnología ha permitido que puedan ser explotadas para dar respuesta a nuevas cuestiones y, a su vez, ponerlas a disposición de manera global, esto ha supuesto nuevos retos y oportunidades para el OSINT.

2.3.1. Retos

1. **Infoxicación.** Su mayor ventaja a la vez es el mayor defecto. El hecho de disponer de acceso a un volumen de información casi ilimitado provoca que se genere excesivo ruido.
2. **Desinformación.** Uno de los retos de los investigadores consiste en detectar la información posiblemente manipulada por terceros actores y ser capaces de verificar la información antes de ser procesada.
3. **Fiabilidad de las fuentes.** No todas las fuentes poseen el mismo nivel de fiabilidad, por lo que seleccionar las fuentes correctas se convierte en uno de los primeros contratiempos de la investigación.

2.3.2. Oportunidades

De la misma forma que surgen retos para los investigadores OSINT, también aparecen nuevas oportunidades al hacer uso de tantas fuentes abiertas.

1. **Menor riesgo.** Recopilar información disponible públicamente no supone ningún riesgo en comparación con otras disciplinas de inteligencia como HUMINT (HUman INTelligence). En el caso de OSINT, la información está disponible para cualquier usuario y su adquisición no implica levantar sospechas ante un potencial adversario.
2. **Fuentes disponibles.** Es cierto que la sobreinformación pueda llegar a ser un problema, pero es arma de doble filo, ya que si sabemos usar bien estas fuentes podemos llegar a recolectar información muy bien contrastada a través de muchas fuentes fiables.
3. **Accesibilidad y bajo coste.** No sólo es importante el hecho de disponer de múltiples fuentes de información, también es destacable su ubicuidad de dichas fuentes y como se accede remotamente a la información que facilitan. Además la adquisición de dicha información suele tener un coste realmente reducido en comparación con otras disciplinas.
4. **Facilitador en investigaciones.** El principal objetivo de OSINT es ayudar a la toma de decisiones o la obtención de conocimiento, por ello se convierte en un aliado indispensable en muchos procesos de investigación, como por ejemplo:
 - **Investigadores financieros**, mediante la detección de evasores de impuestos. Muchas personas que están involucradas en evasión fiscal han sido detectadas y monitorizadas gracias a sus cuentas en redes sociales analizando sus estilos de vida.
 - **Investigadores judiciales**, mediante la detección del origen de injurias, amenazas, extorsiones, etc.
 - **Luchar contra la venta de falsificaciones en línea.** Se pueden utilizar técnicas OSINT para identificar productos y/o servicios fraudulentos.
 - **Analistas de seguridad informática**, mediante la recopilación exhaustiva de información sobre objetivos de la auditoría.
 - **Inteligencia de amenazas**, mediante la recopilación y análisis de indicadores, así como la monitorización de tendencias.
 - **Recursos humanos**, mediante la investigación y análisis de la reputación online de un candidato. Aunque en este apartado el perfil del candidato debe tener alguna relación con el puesto a desempeñar y que el tratamiento, por tanto, tiene fines profesionales, además hay que avisar previamente a dicho candidato del proceso de investigación que se va a realizar, para así ser jurisperitos con la LOPDGDD [17].

- **Analistas de marketing**, mediante la monitorización de campañas, segmentación de usuarios, tendencias del mercado, etc.
- **Objetivos de investigaciones**, ya sea para llevar a cabo investigaciones corporativas (por ejemplo, en fugas de información confidencial), inteligencia competitiva, apoyo en labores policiales, búsqueda de criminales, identificar páginas fraudulentas o maliciosas, etc.



Figura 2.1: Ejemplo actividades realizadas en Internet en 1 minuto en 2021.

2.4– El ciclo de inteligencia

El ciclo de inteligencia en la técnica de OSINT consta de varias fases que se llevan a cabo de manera secuencial. Estas fases son:

1. **Planificación y dirección.** En esta primera fase, se establecen los objetivos y las metas del proceso de OSINT. Se determina qué infor-

mación se necesita recopilar, qué áreas o temas se deben investigar y se establece un plan estratégico para el proceso de inteligencia.

2. **Adquisición.** En esta fase, se recopila la información de fuentes abiertas disponibles. Esto implica la búsqueda y recolección de datos relevantes de fuentes como sitios web, redes sociales, bases de datos públicas, documentos públicos, entre otros. También puede incluir la interacción con personas o comunidades en línea para obtener información relevante.
3. **Procesamiento.** Una vez recopilada la información, se procede a procesarla y organizarla de manera estructurada. Esto puede implicar la limpieza y filtrado de los datos, la extracción de información relevante, la traducción de estos mismos y la transformación de los datos en un formato utilizable.
4. **Análisis y producción.** En esta fase, se realiza un análisis detallado de la información recopilada y procesada. Se examinan los datos en busca de patrones, tendencias, relaciones y cualquier otra información relevante que pueda ayudar a responder las preguntas planteadas en la fase de planificación. A partir de este análisis, se generan productos de inteligencia, como informes, perfiles, evaluaciones de riesgos, entre otros.
5. **Difusión y feedback.** En esta última fase, los productos de inteligencia generados se comunican y comparten con las partes interesadas relevantes. Esto puede incluir la presentación de informes, la entrega de resultados a los tomadores de decisiones o la colaboración con otros analistas. También se recopila y se recibe feedback, lo que ayuda a mejorar los procesos y productos de inteligencia en futuros ciclos.

Es importante destacar que el ciclo de inteligencia en OSINT es un proceso iterativo, lo que significa que las fases se repiten y se retroalimentan entre sí. A medida que se obtiene más información y se recibe feedback, se pueden ajustar y refinar las etapas posteriores del ciclo para mejorar la efectividad del proceso de inteligencia.

2.5— Legalidad

La legitimidad sobre el OSINT puede ser diferente en cada país y localidad. El OSINT en sí mismo no es ilegal en España porque implica la recopilación y análisis de información de fuentes abiertas que están disponibles para todos. Sin embargo, es importante tener en cuenta las formas específicas de obtener información dentro del marco legal.

En cuanto a las formas de obtener información en motores de búsqueda, a continuación se mencionan algunas técnicas comunes, incluyendo el raspado de datos (scraping) y la consideración de su legalidad en España:

- **Motores de búsqueda.** Obtener información a partir de motores como Google o Bing es una forma común de obtener información OSINT. Esta práctica es totalmente legal, ya que estos motores están diseñados para indexar y mostrar información disponible de forma pública en la web.
- **Acceso directo a sitios web.** Visitar y acceder a sitios web públicos sin violar términos de servicio o políticas específicas es una forma legítima de obtener información en OSINT. Al respetar los términos y condiciones de los sitios web, se puede recopilar información legalmente.
- **Acceso a bases de datos públicas.** Algunas entidades o instituciones pueden tener bases de datos públicas disponibles para consulta. Estas bases de datos suelen ser legales de acceder y utilizar, siempre y cuando se cumplan los requisitos y condiciones establecidos por las entidades que las gestionan.
- **Raspado de datos (scraping).** El scraping implica la extracción automatizada de datos de sitios web. En España, el scraping puede ser legal o ilegal dependiendo de varios factores, como los términos de servicio y las políticas del sitio web objetivo. Es importante revisar las condiciones de cada sitio web antes de realizar scraping. Si un sitio web prohíbe explícitamente el scraping o limita el acceso y uso de sus datos, realizar scraping en ese sitio puede ser considerado una infracción contractual. Por ello es siempre recomendable contrastar si el servicio que queremos consultar tiene algún tipo de API o forma de comunicarse con su base de datos de forma que no viole sus políticas.

Es esencial tener en cuenta que las leyes y regulaciones pueden cambiar con el tiempo, por lo que es recomendable consultar a un profesional legal o a fuentes actualizadas para obtener asesoramiento específico sobre la legalidad del OSINT y las prácticas de obtención de información en España.

CAPÍTULO 3

Planificación y costes

El tipo de planificación por el que se ha regido la creación de `OsintSpector` corresponde al uso de metodologías ágiles, concretamente el modelo SCRUM [31]. Esto se debe a que estas metodologías propician la producción de software útil de manera rápida, a la vez que proporciona un enfoque iterativo e incremental, de manera que se optimiza el control de riesgos y la productividad.

Debido a que el desarrollo es realizado por un único integrante, éste tomará todos los roles posibles que pueda haber, como por ejemplo analista, desarrollador, dirección, etc. El tutor tomará el rol de usuario, para dar las opiniones y validaciones necesarias para usar la metodología elegida.

3.1— Análisis temporal

A continuación, en la Tabla 3.1 se expone el análisis temporal de `OsintSpector`. En esta tabla se muestran las fases y tareas a realizar para completar el proyecto junto a su tiempo estimado. Finalmente se muestra el tiempo real dedicado junto a su desviación con el tiempo estimado que se tenía a priori.

Como puede comprobarse en la Tabla 3.1, se obtiene una desviación final de 12 horas superiores al tiempo estimado. Se aprecia que las horas han sido mal planificadas en el desarrollo del código, sobre todo en la parte de implementación en el Back-End, la cual se estimaban 26 horas menos de las reales. Esto ha sido debido a los múltiples problemas que han habido en el camino del desarrollo de la parte del back-end y que no se esperaba tener en un principio, al igual que la parte del front-end, la cual se esperaba terminar 8 horas más tarde de las reales debido a la experiencia del desarrollador en este ámbito. En el resto de tareas hay

variedad en las desviaciones, pero nunca superando las 5 horas arriba o abajo, incluso existe una tarea que se estimó correctamente.

Tarea	Tiempo estimado	Tiempo real	Desviación
Elección del proyecto	2h	2h	0h
Investigación/Documentación tecnologías	12h	11h	-1h
Entorno de desarrollo	13h	9h	-4h
Aprendizaje Tecnologías	18h	21h	+3h
Elicitación de requisitos	13h	11h	-2h
Análisis y diseño	20h	18h	-2h
Implementación Back-End	90h	116h	+26h
Implementación Front-End	90h	82h	-8h
Pruebas	12h	16h	+4h
Memoria del proyecto	41h	37h	-4h
Total	311h	323h	+12h

Tabla 3.1: Estimación temporal del proyecto.

3.2– Diagrama de Gantt

El diagrama de Gantt, generado con la herramienta Clickup [12], se expone en la siguiente Figura 3.1 y muestra el desarrollo del proyecto completo. El diagrama ha sido definido con las siguientes cinco fases:

- Planificación, Figura 3.2..
- Análisis y Diseño, Figura 3.3.
- Ejecución, Figura 3.4.
- Seguimiento y Control, Figura 3.5.
- Cierre, Figura 3.6.

En la primera Figura 3.1 se puede apreciar el Diagrama de Gantt donde hay momentos en los que se realizan dos fases concurrentemente, esto es debido a la naturaleza de ambas, en las que se pueden realizar el mismo tiempo como es el caso de la implementación del back-end y la del front-end, en la que se necesitan una a otra para comprobar la funcionalidad y conexiones entre ellas y por ende se realizan al mismo tiempo.

A continuación, del diagrama de Gantt se pueden visualizar distintas figuras que indican cada fase con sus fechas exactas de inicio y finalización en cada una de sus tareas, para así mostrar de manera detallada cada bloque.

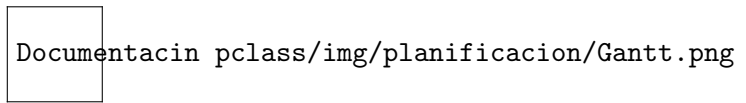


Figura 3.1: Diagrama de Gantt completo.



Figura 3.2: Vista paquete planificación.



Figura 3.3: Vista paquete análisis y diseño.

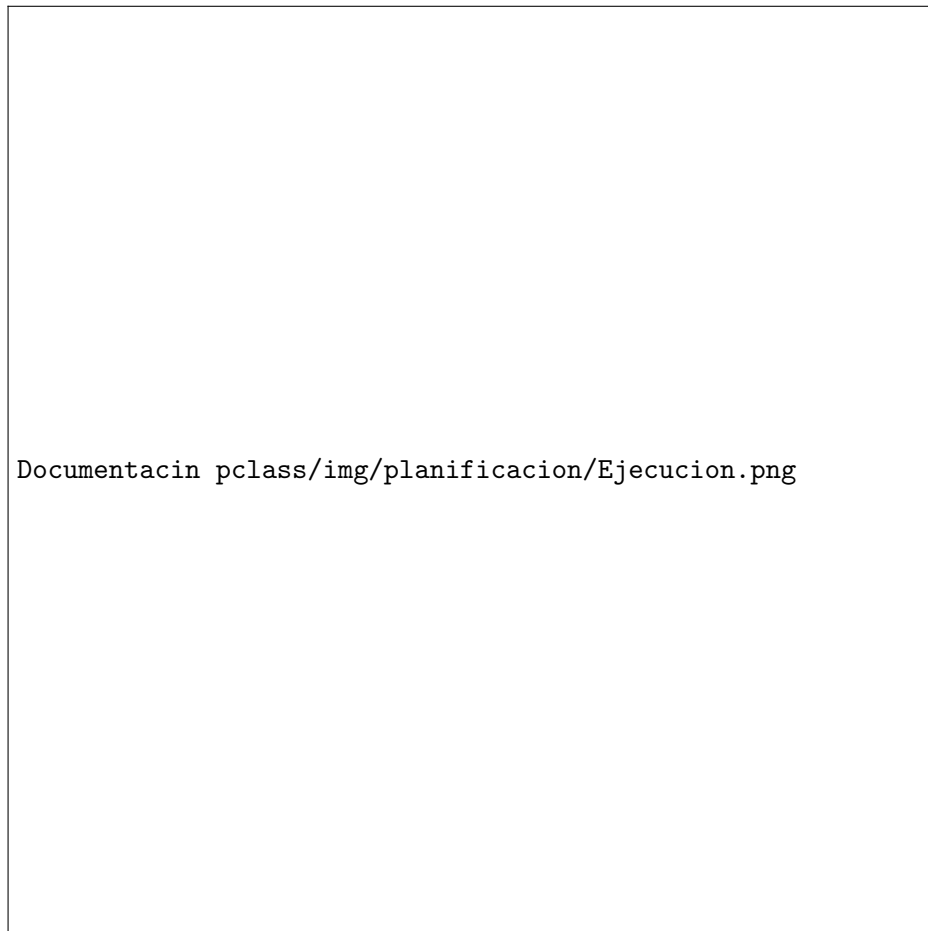


Figura 3.4: Vista paquete ejecución.

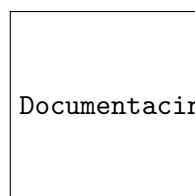


Figura 3.5: Vista paquete seguimiento y control.

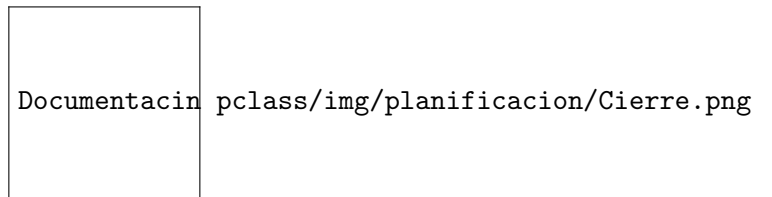


Figura 3.6: Vista paquete cierre.

3.3– Análisis de Costes

En la sección de costes se va a descomponer el coste total del desarrollo del proyecto, desglosándolo; por tanto, en costes directos e indirectos. Además se realizará una comparativa entre los costes estimados y los costes en los que incurre finalmente, mediante los valores recogidos en la Tabla 3.1.

3.3.1. Costes directos

Usando el portal de empleo de GlassDoor [25] se ha podido estimar la media de sueldos para los distintos puestos que se necesitarán en nuestro proyecto. El equipo se compondría de un Full Stack Developer, OSINT Analyst. Procedemos a realizar una media aritmética de los 3 sueldos. Debido a las posibles incoherencias para obtener el salario neto por la situación del trabajador, la comunidad autónoma y distintas casuísticas que influyen en los impuestos, los valores del salario son siempre indicados en bruto.

Puesto de Trabajo	Salario Medio anual	Salario Mensual (12 pagas)
OSINT Analyst	35.772,96€	2981,08€
Full Stack Developer Junior	21.000,00€	1750,00€
Jefe de proyecto	40.000,00€	4044,20€
Media	32.257,65€	2688,14€

Tabla 3.2: Salarios de las posiciones en el proyecto.

A partir de la media obtenida anteriormente se puede obtener el salario bruto del único desarrollador del proyecto que llevará a cabo todos estos roles, siendo 17€por hora. La Tabla 3.3.1 expone los costes estimados del proyecto

Horas trabajadas	Sueldo
311h	5.287€

Tabla 3.3: Costes directos estimados.

3.3.2. Costes reales

A continuación, se desglosará el coste directo real según el análisis temporal del proyecto usando los tiempos reales de desarrollo.

Tarea	Horas trabajadas	Sueldo
Elección del proyecto	2h	34€
Investigación/Documentación tecnologías	11h	187€
Entorno de desarrollo	9h	153€
Aprendizaje tecnologías	21h	357€
Elicitación de requisitos	11h	187€
Análisis y diseño	18h	306€
Implementación Back-End	116h	1.972€
Implementación Front-End	82h	1.394€
Pruebas	16h	176€
Memoria del proyecto	37h	407€
Total	323h	5.491€

Tabla 3.4: Costes reales por tarea.

Se puede comprobar que con respecto al tiempo estimado de 311h el coste del salario real se ve incrementado en 204 euros aproximadamente. Esto quiere decir que se supone un incremento menor del 5 % del valor estimado. En conclusión es un porcentaje de margen de error no muy elevado debido al tamaño del proyecto, aunque podría haber sido más preciso.

3.3.3. Costes indirectos

Se va a tomar como gastos indirectos la amortización del hardware y software que se ha utilizado para llevar a cabo del proyecto. Con respecto al hardware, se dispone de un equipo valorado en 850€, que se amortizará en aproximadamente cuatro años, es decir, un 25 % anual. Además, se incluye todo periférico utilizado, los cuales se amortizarán también alrededor de cuatro años. Estos periféricos son:

- Sobremesa por piezas 850,00€.
- Monitor BenQ GW2283 110,00€.
- Monitor Samsung LS32AG320N 201,98€.
- Teclado Logitech 20,00€.
- Ratón MSI Clutch GM10 20,00€.
- WebCam HD 10,00€.

La duración total del desarrollo del proyecto completo es de 6 meses, por lo tanto, el coste total, se trata del coste mensual multiplicado por los 6 meses del proyecto.

Hardware	Coste mensual	Total
Sobremesa por piezas	17,70€	106,25€
Monitor BenQ GW2283	2,29€	13,75€
Monitor Samsung LS32AG320N	4,21€	25,25€
Teclado Logitech	0,42€	2,5€
Ratón MSI Clutch GM10	0,42€	2,5€
WebCam HD	0,21€	1,25€
Total	25,25€	151,5€

Tabla 3.5: Costes del hardware utilizado en el proyecto.

Los programas software utilizados en el desarrollo, apenas han supuesto un incremento en el coste, debido a que en la mayoría se hacen uso de licencias de estudiante o aplicaciones con licencia gratuita. Estos programas son indicados en la tabla 3.3.3.

Software	Coste Mensual	Coste Total
GitHub Pro (Estudiante)	0,00€	0,00€
Editor de texto L ^A T _E XOverleaf	0,00€	0,00€
Visual Studio Code 1.78.2	0,00€	0,00€
Moqups	0,00€	0,00€
Lucid.app	0,00€	0,00€
IntelX Student API	0,00€	0,00€
Have I Been Pwned API	3,32€	19,92€
Scale SERP API	0,00€	0,00€
Google Geocoding API	0,00€	0,00€
Microsoft Teams	0,00€	0,00€
Total	3,32€	19,92€

Tabla 3.6: Costes del software utilizado en el proyecto.

3.3.4. Costes totales

Una vez se ha obtenido los costes indirectos y directos de nuestro proyecto, llegamos a la conclusión que el desarrollo tendría un coste de **5.662,42 €**. Este coste contempla toda la etapa de desarrollo del proyecto, no se ha tenido en cuenta ninguna etapa posterior como pudiera ser la de despliegue o la de mantenimiento del sistema.

CAPÍTULO 4

Elicitación del problema

En este capítulo se procederá a la elicitación de los requisitos del sistema, lo cual consistirá en una tarea de abstracción para conocer los requisitos y necesidades del mismo. Siguiendo con las directrices del estándar IEEE 830-1998 [27] especificaremos los requisitos mediante las siguientes secciones. La sección 4.1 contemplará el alcance del sistema y la definición de los actores del sistema. La sección 4.2 especificará los requisitos del sistema final, requisitos funcionales 4.2.1, de información 4.2.2, requisitos de negocio 4.2.3 y finalmente los requisitos no funcionales 4.2.4.

Una buena definición de requisitos es crucial de cara a un futuro evitar cambios o modificaciones que afecten a nuestro proyecto, tanto en diseño, como implementación, tiempo dedicado, costos, etc.

4.1— Introducción

4.1.1. Alcance

El objetivo de la herramienta es crear una plataforma que permita recopilar y analizar información pública online sobre una persona, usando información previa dada por un usuario. Este proceso de recopilación y análisis vendrá dado mediante la metodología OSINT 2.

4.1.2. Actores del sistema

Los diferentes actores que podrán aparecer en el sistema propuesto se exponen a continuación. De esta manera, indicaremos en la Sección 4.2.1 los requisitos y funciones que puede realizar cada actor.

Actor 1: Usuario	
Descripción	Este actor representa a la persona que ha accedido al sistema.

Tabla 4.1: Usuario

Actor 2: Sistema	
Descripción	Este actor representa el sistema que estará dotado con las capacidades necesarias para realizar su servicio correctamente.

Tabla 4.2: Sistema

4.2– Requisitos Específicos

4.2.1. Requisitos funcionales

En la primera sección de requisitos se expondrán los funcionales, son aquellos comportamientos que debe tener un sistema para satisfacer las necesidades de los actores. Se acompañará cada requisito con el actor que interactúa para que ocurran dichas acciones, así también para acotarlo en los casos de uso que se mostrará más adelante en el Capítulo 5.

RF-01	Información previa a las búsquedas
Actor	Usuario.
Dependencias	Ninguna.
Descripción	El sistema ofrece la capacidad de informarse sobre el proyecto, para así tener conocimiento de causa sobre la herramienta y sus búsquedas. Además también ofrece la capacidad de consultar quiénes fueron los desarrolladores e información sobre ellos.

Tabla 4.3: Información previa a las búsquedas.

RF-02	Petición Twitter
Actor	Usuario.
Dependencias	RN-02.
Descripción	El sistema ofrece la capacidad de solicitar el análisis de una cuenta de Twitter del objetivo deseado por el Usuario.

Tabla 4.4: Petición Twitter.

RF-03	Análisis de Twitter
Actor	Sistema.
Dependencias	RF-02, RF-05, RF-07, RF-09, RF-11, RF-13
Descripción	El sistema ofrece la capacidad de recopilar toda la información posible del objetivo solicitado. Una vez se recopile dicha información el sistema será capaz de filtrar, evaluar y clasificar estos datos. Finalmente el sistema ofrece la capacidad de visualizar al usuario la información generada, convertida en inteligencia. Tanto en forma de texto como en gráficas.

Tabla 4.5: Análisis de Twitter.

RF-04	Extracción de datos para la nube de palabras
Actor	Sistema.
Dependencias	RF-02, RI-06.
Descripción	El sistema ofrece la capacidad de extracción de la información en crudo necesaria para formar una nube de palabras en base a las palabras más repetidas en los tweets del objetivo.

Tabla 4.6: Extracción de datos para la nube de palabras.

RF-05	Elaboración y visualización de la nube de palabras
Actor	Sistema.
Dependencias	RF-04, RI-07.
Descripción	El sistema ofrece la capacidad de filtrar y evaluar las palabras importantes más repetidas por el objetivo. A su vez el sistema ofrece la capacidad de visualización al cliente de dicha nube de palabras.

Tabla 4.7: Elaboración y visualización de la nube de palabras.

RF-06	Extracción de datos para el análisis de sentimientos
Actor	Sistema.
Dependencias	RF-02, RI-06.
Descripción	El sistema ofrece la capacidad de extracción de la información en crudo necesaria para realizar una clasificación en distintos tipos de sentimientos (alegría, tristeza, enfado, miedo) de cada tweet del objetivo.

Tabla 4.8: Extracción de datos para el análisis de sentimientos.

RF-07	Elaboración y visualización del análisis de sentimientos
Actor	Sistema.
Dependencias	RF-06, RI-08.
Descripción	El sistema ofrece la capacidad de filtrar, evaluar y clasificar cada tweet por el tipo de sentimiento más destacable en dicho tweet. A su vez el sistema ofrece la capacidad de visualización al cliente de dicha clasificación.

Tabla 4.9: Elaboración y visualización de la nube de palabras.

RF-08	Extracción de datos para el grafo de interacciones
Actor	Sistema.
Dependencias	RF-02, RI-06.
Descripción	El sistema ofrece la capacidad de extracción de la información en crudo necesaria para dibujar un grafo. Dicho grafo muestra los usuarios con los que más interactúa la persona buscada en Twitter, aparte de información extra sobre cada uno de éstos.

Tabla 4.10: Extracción de datos para el grafo de interacciones.

RF-09	Elaboración y visualización del grafo de interacciones
Actor	Sistema.
Dependencias	RF-08, RI-09.
Descripción	El sistema ofrece la capacidad de filtrar, evaluar, clasificar, analizar y crear el grafo de interacciones con los usuarios a los que más contacta. A su vez el sistema ofrece la capacidad de dibujo y visualización de dicho grafo junto a información extra de cada partícipe.

Tabla 4.11: Elaboración y visualización del grafo de interacciones.

RF-10	Extracción de datos para el grafo de comunidades
Actor	Sistema.
Dependencias	RF-02, RI-06.
Descripción	El sistema ofrece la capacidad de extracción de la información en crudo necesaria para dibujar el grafo de comunidades. Dicho grafo muestra las relaciones entre los distintos usuarios con los que se comunica la persona formando así pequeñas comunidades.

Tabla 4.12: Extracción de datos para el grafo de comunidades.

RF-11	Elaboración y visualización del grafo de comunidades
Actor	Sistema.
Dependencias	RF-10, RI-10.
Descripción	El sistema ofrece la capacidad de filtrar, evaluar, clasificar, analizar y crear el grafo de comunidades donde se reflejará cada comunidad junto a sus partícipes. A su vez el sistema ofrece la capacidad de dibujo y visualización de dicho grafo junto a datos matemáticos sobre cada grafo para un estudio exhaustivo de estos.

Tabla 4.13: Elaboración y visualización del grafo de comunidades.

RF-12	Extracción de datos para el estudio de localizaciones
Actor	Sistema.
Dependencias	RF-02, RI-06.
Descripción	El sistema ofrece la capacidad de extracción de la información en crudo necesaria para poder hacer un estudio de las distintas localizaciones y fechas desde donde tuiteó el objetivo.

Tabla 4.14: Extracción de datos para el estudio de localizaciones.

RF-13	Elaboración y visualización del estudio de localizaciones
Actor	Sistema.
Dependencias	RF-12, RI-11.
Descripción	El sistema ofrece la capacidad de filtrar, evaluar, clasificar y recopilar las distintas localizaciones y fechas desde donde tuiteó la persona. A su vez el sistema ofrece la capacidad de visualización sobre dichos datos tanto en forma de texto como añadidas en un mapa.

Tabla 4.15: Elaboración y visualización del estudio de localizaciones.

RF-14	Descarga de la información generada en el análisis de Twitter.
Actor	Usuario.
Dependencias	RF-03, RI-08, RI-09, RI-10, RI-11 .
Descripción	El sistema ofrece la capacidad de descargar la información generada en forma de texto para uso personal exceptuando la nube de palabras.

Tabla 4.16: Descarga de la información generada en el análisis de Twitter.

RF-15	Petición búsqueda persona
Actor	Usuario.
Dependencias	RN-01, RN-03.
Descripción	El sistema ofrece la capacidad de solicitar la búsqueda de cierta persona objetivo con todos los datos posibles que sepa el usuario.

Tabla 4.17: Petición búsqueda persona.

RF-16	Búsqueda de persona
Actor	Sistema.
Dependencias	RF-15, RF-18, RF-20, RF-22, RF-24, RF-26, RF-28.
Descripción	El sistema ofrece la capacidad de recopilar toda la información posible del objetivo. Una vez se reco-pile dicha información el sistema será capaz de fil-trar, evaluar y clasificar estos datos. Finalmente el sistema ofrece la capacidad de visualizar al usuario la información generada, convertida en inteligencia. Tanto en forma de texto como en gráficas.

Tabla 4.18: Búsqueda de persona.

RF-17	Extracción de datos del INE [20] sobre nom-bre y/o apellido(s)
Actor	Sistema.
Dependencias	RF-15, RI-01, RI-02.
Descripción	El sistema ofrece la capacidad de extracción y reco-pilación de la información en crudo del INE sobre nombre y/o apellido(s) del objetivo necesaria para su estudio a posteriori.

Tabla 4.19: Extracción de datos del INE sobre nombre y/o apellido(s).

RF-18	Elaboración y visualización de los datos del INE sobre nombre y/o apellido(s)
Actor	Sistema.
Dependencias	RF-17, RI-13.
Descripción	El sistema ofrece la capacidad de filtrar y evaluar la información proporcionada por el INE acerca del nombre y/o apellidos. A su vez el sistema ofrece la capacidad de visualización al actor de dichos datos ya ordenados.

Tabla 4.20: Elaboración y visualización de los datos del INE sobre nombre y/o apellido(s).

RF-19	Extracción de datos sobre el nombre y/o apellido(s) y ciudad en motores de búsqueda
Actor	Usuario.
Dependencias	RF-15, RI-04.
Descripción	El sistema ofrece la capacidad de extracción y recopilación de la información en crudo ofrecida por los motores de búsqueda sobre el nombre y/o apellido(s) y ciudad del objetivo necesaria para su estudio a posteriori.

Tabla 4.21: Extracción de datos sobre el nombre y/o apellido(s) y ciudad en motores de búsqueda.

RF-20	Elaboración y visualización de los datos sobre el nombre y/o apellido(s) y ciudad en motores de búsqueda.
Actor	Sistema.
Dependencias	RF-19, RI-13.
Descripción	El sistema ofrece la capacidad de filtrar, evaluar, clasificar y analizar la información proporcionada por los motores de búsqueda acerca del nombre y/o apellidos y ciudad. A su vez el sistema ofrece la capacidad de visualización al actor de dichos datos ya ordenados tanto en forma de texto como en gráficas.

Tabla 4.22: Elaboración y visualización de los datos sobre el nombre y/o apellido(s) y ciudad en motores de búsqueda.

RF-21	Extracción de datos sobre el nickname en motores de búsqueda
Actor	Sistema.
Dependencias	RF-15, RI-02, RI-03.
Descripción	El sistema ofrece la capacidad de extracción y recopilación de la información en crudo ofrecida por los motores de búsqueda sobre el nickname del objetivo necesaria para su estudio a posteriori.

Tabla 4.23: Extracción de datos sobre el nickname en motores de búsqueda.

RF-22	Elaboración y visualización de los datos sobre el nickname en motores de búsqueda.
Actor	Sistema.
Dependencias	RF-21, RI-15.
Descripción	El sistema ofrece la capacidad de filtrar, evaluar, clasificar y analizar la información proporcionada por los motores de búsqueda acerca del nickname. A su vez el sistema ofrece la capacidad de visualización al actor de dichos datos ya ordenados tanto en forma de texto como en gráficas.

Tabla 4.24: Elaboración y visualización de los datos sobre el nickname en motores de búsqueda.

RF-23	Extracción de datos sobre el correo electrónico en motores de búsqueda
Actor	Sistema.
Dependencias	RF-15, RI-04.
Descripción	El sistema ofrece la capacidad de extracción y recopilación de la información en crudo ofrecida por los motores de búsqueda sobre el correo electrónico del objetivo necesaria para su estudio a posteriori.

Tabla 4.25: Extracción de datos sobre el correo electrónico en motores de búsqueda.

RF-24	Elaboración y visualización de los datos sobre el correo electrónico en motores de búsqueda.
Actor	Sistema.
Dependencias	RF-23, RI-17.
Descripción	El sistema ofrece la capacidad de filtrar, evaluar, clasificar y analizar la información proporcionada por los motores de búsqueda acerca del correo electrónico. A su vez el sistema ofrece la capacidad de visualización al actor de dichos datos ya ordenados tanto en forma de texto como en gráficas.

Tabla 4.26: Elaboración y visualización de los datos sobre el correo electrónico en motores de búsqueda.

RF-25	Extracción de datos sobre el número de teléfono en motores de búsqueda
Actor	Sistema.
Dependencias	RF-15, RI-04.
Descripción	El sistema ofrece la capacidad de extracción y recopilación de la información en crudo ofrecida por los motores de búsqueda sobre el número de teléfono del objetivo necesaria para su estudio a posteriori.

Tabla 4.27: Extracción de datos sobre el número de teléfono en motores de búsqueda.

RF-26	Elaboración y visualización de los datos sobre el número de teléfono en motores de búsqueda.
Actor	Sistema.
Dependencias	RF-25, RI-19.
Descripción	El sistema ofrece la capacidad de filtrar, evaluar, clasificar y analizar la información proporcionada por los motores de búsqueda acerca del número de teléfono. A su vez el sistema ofrece la capacidad de visualización al actor de dichos datos ya ordenados tanto en forma de texto como en gráficas.

Tabla 4.28: Elaboración y visualización de los datos sobre el número de teléfono en motores de búsqueda.

RF-27	Extracción de datos en la Darknet sobre los atributos rellenados
Actor	Sistema.
Dependencias	RF-15, RI-01, RI-02
Descripción	El sistema ofrece la capacidad de extracción y recopilación de la información en crudo indexada en la Darknet donde se mencione los atributos rellenados previamente, pueden ser: nombre y/o apellido(s), nickname, correo electrónico y/o número de teléfono.

Tabla 4.29: Extracción de datos en la Darknet sobre los atributos rellenados.

RF-28	Elaboración y visualización de los datos de la Darknet sobre los atributos rellenados.
Actor	Sistema.
Dependencias	RF-27, RI-21.
Descripción	El sistema ofrece la capacidad de filtrar y clasificar la información proporcionada por la Darknet acerca de los atributos rellenados previamente. A su vez el sistema ofrece la capacidad de visualización al usuario de dichos datos ya ordenados.

Tabla 4.30: Elaboración y visualización de los datos de la Darknet sobre los atributos rellenados.

RF-29	Descarga de la información generada en la búsqueda de la persona.
Actor	Usuario.
Dependencias	RF-16, RI-14, RI-16, RI-18, RI-20, RI-21.
Descripción	El sistema ofrece la capacidad de descargar la información generada en el apartado de búsqueda de persona en forma de texto para uso personal.

Tabla 4.31: Descarga de la información generada en la búsqueda de la persona.

4.2.2. Requisitos de información

En nuestro sistema se guardan bastantes tipos diferentes de información originados en cada búsqueda/análisis para cumplir con los requisitos funcionales expuestos anteriormente en la Sección 4.2.1. Igualmente para poder buscar y generar esta información también se necesitarán otros datos usados como base.

RI-01	Proxies
Dependencias	Ninguna.
Descripción	El sistema ofrece la capacidad de almacenar servidores proxies [29] públicos actualizados constantemente para utilizarlos en distintos servicios externos.
Datos almacenados por proxy	<ul style="list-style-type: none"> • Dirección IP y puerto.

Tabla 4.32: Proxies.

RI-02	User-Agents
Dependencias	Ninguna.
Descripción	El sistema ofrece la capacidad de almacenar múltiples <i>User-Agents</i> [23] para formar distintas cabeceras de peticiones a servicios externos.
Datos almacenados	<ul style="list-style-type: none"> • Conjunto de muchos User-Agents distintos entre sí.

Tabla 4.33: User-Agents.

RI-03	Motores Nickname
Dependencias	Ninguna.
Descripción	El sistema debe almacenar múltiples motores de búsquedas con sus atributos respectivos para la búsqueda de nicknames.
Datos almacenados por motor de búsqueda	<ul style="list-style-type: none">• Nombre.• URI.• Código de estado de respuesta satisfactorio.• Código de estado de respuesta fallido.• Frase de respuesta del motor si existe la cuenta.• Frase de respuesta del motor si no existe la cuenta.• Cuentas que si existen en dicho motor de búsqueda.• Categoría.• Valor binario si está funcional el servicio de dicho motor.

Tabla 4.34: Motores Nickname.

RI-04	API keys
Dependencias	Ninguna.
Descripción	El sistema ofrece la capacidad de almacenar variables de entorno que contengan las claves de cada API que se utilice.
Datos almacenados por cada clave	<ul style="list-style-type: none">• Nombre.• Clave de la API.

Tabla 4.35: API keys.

RI-05	Username Twitter
Dependencias	Ninguna.
Descripción	El sistema ofrece la capacidad de almacenar el nombre de usuario aportado por el actor en la parte de análisis de Twitter.
Datos almacenados de la información aportada	<ul style="list-style-type: none">• Nombre de usuario.

Tabla 4.36: Username Twitter.

RI-06	Recopilación Twitter
Dependencias	RI-04, RI-05.
Descripción	El sistema ofrece la capacidad de almacenar la información cruda del username indicado.
Datos almacenados por cada tweet	<ul style="list-style-type: none">• Fecha de publicación.• ID.• Texto en crudo.• URL.• Localización.

Tabla 4.37: Recopilación Twitter.

RI-07	Wordcloud Twitter
Dependencias	RI-06.
Descripción	El sistema ofrece la capacidad de almacenar la información ordenada y procesada sobre el usuario para el apartado de la nube de palabras.
Datos almacenados para la nube de palabras	<ul style="list-style-type: none">• Nube de palabras.

Tabla 4.38: Wordcloud Twitter.

RI-08	Sentimientos
Dependencias	RI-06.
Descripción	El sistema ofrece la capacidad de almacenar la información ordenada y procesada sobre el usuario para el apartado de análisis de sentimientos.
Datos almacenados para el análisis de sentimientos	<ul style="list-style-type: none"> • Cada tweet con su respectivo sentimiento calificado. • El tweet que más destacable por cada sentimiento.

Tabla 4.39: Sentimientos Twitter.

RI-09	Interacciones
Dependencias	RI-06.
Descripción	El sistema ofrece la capacidad de almacenar la información ordenada y procesada sobre el usuario para el grafo de interacciones.
Datos almacenados por cada usuario del grafo	<ul style="list-style-type: none"> • Nombre de usuario. • N° de interacciones con nuestro usuario objetivo.

Tabla 4.40: Interacciones Twitter.

RI-10	Comunidades
Dependencias	RI-06.
Descripción	El sistema ofrece la capacidad de almacenar la información ordenada y procesada sobre el usuario para el grafo de comunidades.
Datos almacenados por cada comunidad	<ul style="list-style-type: none"> • Conjunto de usuarios de la comunidad. • Conjunto de conexiones entre dichos usuarios. • Valores matemáticos basados en teoría de grafos.

Tabla 4.41: Comunidades Twitter.

RI-11	Localizaciones
Dependencias	RI-04, RI-06.
Descripción	El sistema ofrece la capacidad de almacenar la información ordenada y procesada sobre el usuario para el apartado de localizaciones.
Datos almacenados por cada tweet con localización	<ul style="list-style-type: none"> • Localizaciones agrupadas por lugar donde tuiteó el usuario. <ul style="list-style-type: none"> ◦ Enlace de Google Maps sobre el lugar. ◦ Enlace(s) del propio tweet. ◦ Fecha(s) de cuando tuiteó el usuario.

Tabla 4.42: Información ordenada e inteligencia sobre el usuario de Twitter para el apartado de localizaciones.

RI-12	Persona
Dependencias	Ninguna.
Descripción	El sistema ofrece la capacidad de almacenar distintos atributos aportados por el actor en la parte de búsqueda de la persona.
Datos almacenados de la información aportada	<ul style="list-style-type: none"> • Nombre. • Apellido(s). • Ciudad. • Nickname. • Correo electrónico. • Número de teléfono. • Opción de búsqueda en la Darknet.

Tabla 4.43: Persona.

RI-13	Recopilación Nombre-Apellido(s)-Ciudad
Dependencias	RI-04, RI-12.
Descripción	El sistema ofrece la capacidad de almacenar la información cruda sobre la búsqueda de nombre y/o apellido(s) y ciudad de la persona que se va a analizar a posteriori.
Datos almacenados de nombre y/o apellido(s) y ciudad	<ul style="list-style-type: none"> • Datos aportados por el INE. • Datos aportados por los motores de búsqueda.

Tabla 4.44: Recopilación Nombre-Apellido(s)-Ciudad.

RI-14	Inteligencia Nombre-Apellido(s)-Ciudad
Dependencias	RI-13.
Descripción	El sistema ofrece la capacidad de almacenar la información ordenada y procesada sobre el nombre y/o apellido(s) y ciudad del objetivo.
Datos almacenados de nombre y/o apellido(s) y ciudad	<ul style="list-style-type: none">• Información ordenada sobre los datos aportados por el INE.• Información ordenada sobre los datos aportados por los motores de búsqueda.• Gráficas basadas en dicha información dada por los motores de búsqueda.

Tabla 4.45: Inteligencia Nombre-Apellido(s)-Ciudad.

RI-15	Recopilación Nickname
Dependencias	RI-02, RI-03, RI-12.
Descripción	El sistema ofrece la capacidad de almacenar la información cruda sobre la búsqueda del nickname de la persona que se va a analizar a posteriori.
Datos almacenados del nickname	<ul style="list-style-type: none">• Motores de búsqueda donde está indexado el nickname.

Tabla 4.46: Recopilación Nickname.

RI-16	Inteligencia Nickname
Dependencias	RI-15.
Descripción	El sistema ofrece la capacidad de almacenar la información ordenada y procesada sobre el nickname.
Datos almacenados del nickname	<ul style="list-style-type: none"> • Motores de búsqueda donde está indexado el nickname. • Categorías de cada motor de búsqueda. • Gráficas basadas en dicha información dada por los motores de búsqueda.

Tabla 4.47: Inteligencia Nickname.

RI-17	Recopilación e-mail
Dependencias	RI-04, RI-12.
Descripción	El sistema ofrece la capacidad de almacenar la información cruda sobre la búsqueda del correo electrónico de la persona que se va a analizar a posteriori.
Datos almacenados del correo electrónico	<ul style="list-style-type: none"> • Conjunto de las distintas brechas de seguridad donde se filtró el correo electrónico.

Tabla 4.48: Recopilación e-mail.

RI-18	Inteligencia e-mail
Dependencias	RI-17.
Descripción	El sistema ofrece la capacidad de almacenar la información ordenada y procesada sobre el correo electrónico.
Datos almacenados del correo electrónico	<ul style="list-style-type: none">• Conjunto de las distintas brechas de seguridad donde se filtró el correo electrónico.• Gráficas basadas en dicha información de las brechas.

Tabla 4.49: Inteligencia e-mail

RI-19	Recopilación Teléfono
Dependencias	RI-04, RI-12.
Descripción	El sistema ofrece la capacidad de almacenar la información cruda sobre la búsqueda del número de teléfono de la persona que se va a analizar a posteriori.
Datos almacenados del número de teléfono	<ul style="list-style-type: none">• Conjunto de las distintas brechas de seguridad donde se filtró el número de teléfono.

Tabla 4.50: Recopilación Teléfono.

RI-20	Inteligencia Teléfono
Dependencias	RI-19.
Descripción	El sistema ofrece la capacidad de almacenar la información ordenada y procesada sobre el número de teléfono del objetivo.
Datos almacenados del número de teléfono	<ul style="list-style-type: none"> • Conjunto de las distintas brechas de seguridad donde se filtró el número de teléfono. • Gráficas basadas en dicha información de las brechas.

Tabla 4.51: Inteligencia Teléfono.

RI-21	Recopilación Darknet.
Dependencias	RI-01, RI-02, RI-12.
Descripción	El sistema ofrece la capacidad de almacenar la información recopilada de la búsqueda en la Darknet sobre distintos atributos anteriormente rellenados.
Datos almacenados de la Darknet	<ul style="list-style-type: none"> • Lugares y fecha de actualización donde fue indexado el nombre y/o apellido(s). • Lugares y fecha de actualización donde fue indexado el nickname. • Lugares y fecha de actualización donde fue indexado el correo electrónico. • Lugares y fecha de actualización donde fue indexado el número de teléfono.

Tabla 4.52: Recopilación Darknet.

4.2.3. Requisitos de negocio

Para acotar de una manera más precisa la funcionalidad de búsqueda y evitar búsquedas banales o vanas procedemos a exponer ciertas restricciones.

RN-01	Búsqueda trivial de personas
Actor	Usuario.
Condiciones	Rellenar solo la casilla <i>Ciudad</i> y/o elegir la opción de buscar en la Darknet.
Excepciones	Mensaje de error en el cliente antes de realizar cualquier búsqueda.
Descripción	Para poder realizar una búsqueda de cierta persona no podremos rellenar solo la casilla <i>Ciudad</i> , ya que no nos dice nada sobre ella. Al igual que solo elegir la opción de buscar en la Darknet, ya que tampoco nos indica nada sobre la persona en sí.

Tabla 4.53: Búsqueda trivial de personas.

RN-02	Búsqueda usuarios no válidos en Twitter
Actor	Usuario.
Condiciones	Enviar petición de análisis de Twitter de un nombre de usuario no válido en dicha plataforma [13].
Excepciones	Mensaje de error en el cliente antes de realizar cualquier búsqueda.
Descripción	Bloquear la posibilidad de rellenar y enviar la petición de análisis de un usuario no válido para la plataforma de Twitter.

Tabla 4.54: Búsqueda usuarios no válidos en Twitter.

RN-03	Atributos de búsqueda de personas opcionales
Actor	Usuario.
Condiciones	Rellenar los datos para la búsqueda de personas.
Excepciones	RN-01.
Descripción	Todo atributo o dato que necesite aportar el usuario cliente para realizar la búsqueda de personas es totalmente opcional, exceptuando ciertos casos dados en la RN-01 (tabla 4.53).

Tabla 4.55: Atributos de búsqueda de personas opcionales.

4.2.4. Requisitos no funcionales

Los requisitos no funcionales, garantizan el funcionamiento óptimo del sistema. Si los requisitos funcionales especifican lo que debe hacer un sistema para satisfacer las necesidades del usuario, los requisitos no funcionales describen cómo se hará.

RNF-01	Diseño responsive y simple
Descripción	Deberá ser posible usar la web con cualquier tipo de navegador y dispositivo. Con una interfaz simple, concisa y <i>responsive</i> con las dimensiones del dispositivo.

Tabla 4.56: Diseño responsive y simple.

RNF-02	Recopilación de proxies rotatorios
Descripción	Por cada vez que se use el raspado de datos en las búsquedas de personas necesitaremos servidores proxies. Esto es debido a que no queremos ser bloqueados de dichas páginas webs con nuestra IP original. Por ello es más cómodo usar proxies que se actualizan cada cierto tiempo y nos hagan de puente para conectarnos a dichas páginas deseadas.

Tabla 4.57: Recopilación de proxies rotatorios.

RNF-03	Ayuda en el anonimato del usuario y sus búsquedas
Descripción	Apoyar el anonimato del usuario cliente que realiza las búsquedas, evitando guardar información sobre ellos o las búsquedas/análisis finales que haya realizado. Una vez el usuario salga de la pantalla de visualización de resultados todos estos serán eliminados sin dejar rastro, tanto en el apartado de análisis de Twitter como en el de búsqueda de personas.

Tabla 4.58: Ayuda en el anonimato del usuario.

CAPÍTULO 5

Análisis y diseño

En este capítulo se va a profundizar en las fases de análisis y diseño de la aplicación web. Estas fases ayudan a convertir los requisitos vistos en el capítulo anterior en un modelo que de vista a un futuro se intente implementar.

En la sección 5.1 se muestra el modelo completo de casos de uso, junto a las tablas descriptivas de cada caso de uso disponible en la solución. El capítulo continúa con la sección 5.2 exponiendo el diagrama de secuencia que se produce al interactuar con la web. La sección 5.3 cuenta con el modelo de datos describiendo los atributos y relaciones. Finalizamos con la sección 5.4 que expone la arquitectura que se ha seguido para la implementación.

5.1– Modelo de casos de uso

A continuación, se muestra un diagrama que indica los casos de uso disponibles, junto al actor Usuario de nuestro proyecto.

Los bloques contienen las funcionalidades relacionadas con su título, y el actor, apunta a los bloques de acciones que puede realizar.



Documentacin pclass/img/diagramas/ModeloDeCasosDeUsos.png

Figura 5.1: Diagrama casos de uso.

Ahora se describirá de manera mas detallada y concreta los casos de usos mostrados anteriormente en el diagrama 5.1. Siguiendo con las directrices y definiciones impuestas por la Guía para la redacción de casos de uso [15] propuesta por el Marco de Desarrollo de la Junta de Andalucía se busca describir cómo actuará el sistema al realizar una acción por parte

de nuestro actor el usuario.

En este primer bloque se describirán los casos de uso asociados a la documentación previa a las funciones principales.

CU-01	Información previa a las búsquedas	
Dependencias	RF-01.	
Descripción	El sistema se comportará como se indica en el siguiente caso de uso cuando un usuario quiera documentarse previamente a realizar cualquier tipo de búsqueda.	
Precondición	Ninguna.	
Secuencia normal	Paso	Acción
	1	El usuario accede al sistema.
	2	El sistema muestra la página principal.
	3	El usuario pulsa sobre el apartado de información previa.
	4	El sistema muestra el apartado donde se documenta sobre dichas búsquedas.
Postcondición	Ninguna.	
Excepciones	Ninguna.	

Tabla 5.1: Información previa a las búsquedas.

CU-02	Información sobre el desarrollador	
Dependencias	RF-01.	
Descripción	El sistema se comportará como se indica en el siguiente caso de uso cuando un usuario quiera consultar quiénes desarrollaron la herramienta y saber su contacto para cualquier duda/petición.	
Precondición	Ninguna.	
Secuencia normal	Paso	Acción
	1	El usuario accede al sistema.
	2	El sistema muestra la página principal.
	3	El usuario solicita ir al apartado de información sobre el desarrollador.
	4	El sistema muestra el apartado de información sobre el desarrollador.
Postcondición	Ninguna.	
Excepciones	Ninguna.	

Tabla 5.2: Información sobre el desarrollador.

En este segundo bloque se describirán los casos de uso asociados a la sección de una de las funciones principales del sistema, el cual se trata del análisis sobre una cuenta de Twitter objetivo dada por el usuario. El inicio y excepción de todas es el mismo, ya que, como se ha comentado pertenecen a la sección de Twitter y por ello tienen algunas similitudes. En la Figura 5.1 se puede observar visualmente el bloque de análisis de Twitter y dentro los siguientes casos de uso:

CU-03	Visualización de la nube de palabras	
Dependencias	RF-05, RI-07, RN-02.	
Descripción	El sistema se comportará como se indica en el siguiente caso de uso cuando un usuario rellene el formulario de análisis de Twitter para visualizar la nube de palabras sobre el usuario de Twitter objetivo.	
Precondición	RF-02.	
Secuencia normal	Paso	Acción
	1	El usuario accede al sistema.
	2	El sistema muestra la página principal.
	3	El usuario solicita ir al apartado de análisis de Twitter.
	4	El sistema muestra el apartado de análisis de Twitter.
	5	El usuario rellena el formulario de análisis de Twitter y envía la petición.
	6	El sistema obtendrá la información cruda.
	7	El sistema filtrará, evaluará y analizará dichos datos para crear la nube de palabras.
	8	El usuario visualizará la nube de palabras.
Postcondición	Ninguna.	
Excepciones	Si en el formulario escribe un nombre de usuario no válido para Twitter [13] se mostrará un mensaje de error indicando el porqué el análisis que solicita se denegó.	

Tabla 5.3: Visualización de la nube de palabras.

CU-04	Visualización sobre el análisis de sentimientos	
Dependencias	RF-07, RI-08, RN-02.	
Descripción	El sistema se comportará como se indica en el siguiente caso de uso cuando un usuario rellene el formulario de análisis de Twitter para visualizar el apartado de análisis de sentimientos sobre el usuario de Twitter objetivo.	
Precondición	RF-02.	
Secuencia normal	Paso	Acción
	1	El usuario accede al sistema.
	2	El sistema muestra la página principal.
	3	El usuario solicita ir al apartado de análisis de Twitter.
	4	El sistema muestra el apartado de análisis de Twitter.
	5	El usuario rellena el formulario de análisis de Twitter y envía la petición.
	6	El sistema obtendrá la información cruda.
	7	El sistema filtrará, evaluará y analizará dichos datos para crear el análisis de sentimientos.
	8	El usuario visualizará los resultados de dicho análisis.
Postcondición	Ninguna.	
Excepciones	Si en el formulario escribe un nombre de usuario no válido para Twitter [13] se mostrará un mensaje de error indicando el porqué el análisis que solicita se denegó.	

Tabla 5.4: Visualización sobre el análisis de sentimientos.

CU-05	Visualización del grafo de interacciones	
Dependencias	RF-09, RI-09, RN-02	
Descripción	El sistema se comportará como se indica en el siguiente caso de uso cuando un usuario rellene el formulario de análisis de Twitter para visualizar el grafo de interacciones sobre el usuario de Twitter objetivo	
Precondición	RF-02.	
Secuencia normal	Paso	Acción
	1	El usuario accede al sistema.
	2	El sistema muestra la página principal.
	3	El usuario solicita ir al apartado de análisis de Twitter.
	4	El sistema muestra el apartado de análisis de Twitter.
	5	El usuario rellena el formulario de análisis de Twitter y envía la petición.
	6	El sistema obtendrá la información cruda.
	7	El sistema filtrará, evaluará y analizará dichos datos para crear el grafo de interacciones.
	8	El usuario visualizará el grafo de interacciones.
Postcondición	Ninguna.	
Excepciones	Si en el formulario escribe un nombre de usuario no válido para Twitter [13] se mostrará un mensaje de error indicando el porqué el análisis que solicita se denegó.	

Tabla 5.5: Visualización del grafo de interacciones.

CU-06	Visualización del grafo de comunidades	
Dependencias	RF-11, RI-10, RN-02.	
Descripción	El sistema se comportará como se indica en el siguiente caso de uso cuando un usuario rellene el formulario de análisis de Twitter para visualizar el grafo de comunidades sobre el usuario de Twitter objetivo	
Precondición	RF-02.	
Secuencia normal	Paso	Acción
	1	El usuario accede al sistema.
	2	El sistema muestra la página principal.
	3	El usuario solicita ir al apartado de análisis de Twitter.
	4	El sistema muestra el apartado de análisis de Twitter.
	5	El usuario rellena el formulario de análisis de Twitter y envía la petición.
	6	El sistema obtendrá la información cruda.
	7	El sistema filtrará, evaluará y analizará dichos datos para crear el grafo de comunidades.
	8	El usuario visualizará el grafo de comunidades.
Postcondición	Ninguna.	
Excepciones	Si en el formulario escribe un nombre de usuario no válido para Twitter [13] se mostrará un mensaje de error indicando el porqué el análisis que solicita se denegó.	

Tabla 5.6: Visualización del grafo de comunidades.

CU-07	Visualización de la recopilación de localizaciones	
Dependencias	RF-13, RI-11, RN-02.	
Descripción	El sistema se comportará como se indica en el siguiente caso de uso cuando un usuario rellene el formulario de análisis de Twitter para visualizar la recopilación de localizaciones desde donde tuiteó el usuario de Twitter objetivo	
Precondición	RF-02.	
Secuencia normal	Paso	Acción
	1	El usuario accede al sistema.
	2	El sistema muestra la página principal.
	3	El usuario solicita ir al apartado de análisis de Twitter.
	4	El sistema muestra el apartado de análisis de Twitter.
	5	El usuario rellena el formulario de análisis de Twitter y envía la petición.
	6	El sistema obtendrá la información cruda.
	7	El sistema filtrará, evaluará y analizará dichos datos para crear la recopilación de localizaciones.
	8	El usuario visualizará la recopilación de localizaciones.
Postcondición	Ninguna.	
Excepciones	Si en el formulario escribe un nombre de usuario no válido para Twitter [13] se mostrará un mensaje de error indicando el porqué el análisis que solicita se denegó.	

Tabla 5.7: Visualización de la recopilación de localizaciones.

CU-08	Descarga de la información generada	
Dependencias	RF-03.	
Descripción	El sistema se comportará como se indica en el siguiente caso de uso cuando un usuario rellene el formulario de análisis de Twitter para a posteriori poder descargar dichos datos para su uso personal.	
Precondición	RF-02.	
Secuencia normal	Paso	Acción
	1	El usuario accede al sistema.
	2	El sistema muestra la página principal.
	3	El usuario solicita ir al apartado de análisis de Twitter.
	4	El sistema muestra el apartado de análisis de Twitter.
	5	El usuario rellena el formulario de análisis de Twitter y envía la petición.
	6	El sistema obtendrá la información cruda.
	7	El sistema filtrará, evaluará y analizará dichos datos para simplificarlos y ordenarlos.
	8	El usuario visualizará toda la información junto a una opción de descarga que deberá clicar.
	9	El sistema generará una recopilación de todos estos datos y los enviará al usuario.
	10	El usuario recibirá dichos datos y aceptará la descarga en su dispositivo.
Postcondición	Ninguna.	
Excepciones	Si en el formulario escribe un nombre de usuario no válido para Twitter [13] se mostrará un mensaje de error indicando el porqué el análisis que solicita se denegó.	

Tabla 5.8: Descarga de la información generada.

En este tercer y último bloque se describirán los casos de uso asociados a la sección de una de las funciones principales del sistema, el cual se trata de la búsqueda de información sobre una persona objetivo dada por el usuario. El inicio de todas es el mismo, ya que, como se ha comentado pertenecen a la sección de búsqueda de personas y por ello tienen algunas similitudes. En la Figura 5.1 se puede observar visualmente el bloque de búsqueda de información de la persona y dentro los siguientes casos de

uso:

CU-09	Visualización de los datos sobre el nombre y/o apellido(s) del INE	
Dependencias	RF-18, RI-14, RN-03.	
Descripción	El sistema se comportará como se indica en el siguiente caso de uso cuando un usuario rellene el apartado de nombre y/o apellido(s) en el formulario de búsqueda de personas, para obtener información pública del INE sobre dicha persona.	
Precondición	Ninguna.	
Secuencia normal	Paso	Acción
	1	El usuario accede al sistema.
	2	El sistema muestra la página principal.
	3	El usuario solicita ir al apartado de búsqueda de personas.
	4	El sistema muestra el apartado de búsqueda de personas.
	5	El usuario rellena el apartado de nombre y/o apellido(s) del formulario de búsqueda de personas y envía la petición.
	6	El sistema obtendrá la información cruda en base a lo solicitado en el formulario.
	7	El sistema filtrará, evaluará y analizará dichos datos para simplificarlos y ordenarlos.
	8	El usuario visualizará toda la información aportada por el INE para dichos datos.
Postcondición	Ninguna.	
Excepciones	Ninguna.	

Tabla 5.9: Visualización de los datos sobre el nombre y/o apellido(s) del INE.

CU-10	Visualización de los datos sobre el nombre y/o apellido(s) y ciudad en motores de búsqueda	
Dependencias	RF-20, RI-14, RN-01, RN-03.	
Descripción	El sistema se comportará como se indica en el siguiente caso de uso cuando un usuario rellene el apartado de nombre y/o apellido(s) y ciudad en el formulario de búsqueda de personas, para obtener información pública en motores de búsqueda sobre dicha persona.	
Precondición	RF-15.	
Secuencia normal	Paso	Acción
	1	El usuario accede al sistema.
	2	El sistema muestra la página principal.
	3	El usuario solicita ir al apartado de búsqueda de personas.
	4	El sistema muestra el apartado de búsqueda de personas.
	5	El usuario rellena el apartado de nombre y/o apellido(s) y ciudad del formulario de búsqueda de personas y envía la petición.
	6	El sistema obtendrá la información cruda en base a lo solicitado en el formulario.
	7	El sistema filtrará, evaluará y analizará dichos datos para simplificarlos y ordenarlos.
	8	El usuario visualizará toda la información aportada por los motores de búsqueda para dichos datos.
Postcondición	Ninguna.	
Excepciones	Si en el formulario solo se rellena el apartado de ciudad se mostrará un mensaje de error indicando el porqué no se acepto dicha solicitud de búsqueda.	

Tabla 5.10: Visualización de los datos sobre el nombre y/o apellido(s) en motores de búsqueda.

CU-11	Visualización de los datos sobre el nickname en motores de búsqueda	
Dependencias	RF-22, RI-16, RN-03.	
Descripción	El sistema se comportará como se indica en el siguiente caso de uso cuando un usuario rellene el apartado de nombre y/o apellido(s) y ciudad en el formulario de búsqueda de personas, para obtener información pública en motores de búsqueda sobre dicha persona.	
Precondición	RF-15.	
Secuencia normal	Paso	Acción
	1	El usuario accede al sistema.
	2	El sistema muestra la página principal.
	3	El usuario solicita ir al apartado de búsqueda de personas.
	4	El sistema muestra el apartado de búsqueda de personas.
	5	El usuario rellena el apartado de nickname del formulario de búsqueda de personas y envía la petición.
	6	El sistema obtendrá la información cruda en base a lo solicitado en el formulario.
	7	El sistema filtrará, evaluará y analizará dichos datos para simplificarlos y ordenarlos.
	8	El usuario visualizará toda la información aportada por los motores de búsqueda para dichos datos.
Postcondición	Ninguna.	
Excepciones	Ninguna.	

Tabla 5.11: Visualización de los datos sobre el nickname en motores de búsqueda.

CU-12	Visualización de los datos sobre el correo electrónico en motores de búsqueda	
Dependencias	RF-24, RI-18, RN-03.	
Descripción	El sistema se comportará como se indica en el siguiente caso de uso cuando un usuario rellene el apartado de correo electrónico en el formulario de búsqueda de personas, para obtener información pública en motores de búsqueda sobre dicha persona.	
Precondición	RF-15.	
Secuencia normal	Paso	Acción
	1	El usuario accede al sistema.
	2	El sistema muestra la página principal.
	3	El usuario solicita ir al apartado de búsqueda de personas.
	4	El sistema muestra el apartado de búsqueda de personas.
	5	El usuario rellena el apartado de correo electrónico del formulario de búsqueda de personas y envía la petición.
	6	El sistema obtendrá la información cruda en base a lo solicitado en el formulario.
	7	El sistema filtrará, evaluará y analizará dichos datos para simplificarlos y ordenarlos.
	8	El usuario visualizará toda la información aportada por los motores de búsqueda para dichos datos.
Postcondición	Ninguna.	
Excepciones	Ninguna.	

Tabla 5.12: Visualización de los datos sobre el correo electrónico en motores de búsqueda.

CU-13	Visualización de los datos sobre el número de teléfono en motores de búsqueda	
Dependencias	RF-26, RI-20, RN-03.	
Descripción	El sistema se comportará como se indica en el siguiente caso de uso cuando un usuario rellene el apartado de número de teléfono en el formulario de búsqueda de personas, para obtener información pública en motores de búsqueda sobre dicha persona.	
Precondición	RF-15.	
Secuencia normal	Paso	Acción
	1	El usuario accede al sistema.
	2	El sistema muestra la página principal.
	3	El usuario solicita ir al apartado de búsqueda de personas.
	4	El sistema muestra el apartado de búsqueda de personas.
	5	El usuario rellena el apartado de número de teléfono del formulario de búsqueda de personas y envía la petición.
	6	El sistema obtendrá la información cruda en base a lo solicitado en el formulario.
	7	El sistema filtrará, evaluará y analizará dichos datos para simplificarlos y ordenarlos.
	8	El usuario visualizará toda la información aportada por los motores de búsqueda para dichos datos.
Postcondición	Ninguna.	
Excepciones	Ninguna.	

Tabla 5.13: Visualización de los datos sobre el número de teléfono en motores de búsqueda.

CU-14	Visualización de los datos indexados en la Darknet sobre los atributos rellenados.	
Dependencias	RF-28, RI-21, RN-01, RN-03.	
Descripción	El sistema se comportará como se indica en el siguiente caso de uso cuando un usuario rellene el apartado de nombre, apellido(s), nickname, correo electrónico, número de teléfono o una combinación entre ellos y agregue la opción de búsqueda en la Darknet. Para así obtener información indexada en la Darknet donde aparezcan dichos atributos, cada uno de ellos se visualizarán en distintos apartados para llevar un orden.	
Precondición	RF-15.	
Secuencia normal	Paso	Acción
	1	El usuario accede al sistema.
	2	El sistema muestra la página principal.
	3	El usuario solicita ir al apartado de búsqueda de personas.
	4	El sistema muestra el apartado de búsqueda de personas.
	5	El usuario rellena algún(os) apartado(s) del formulario de búsqueda de personas.
	6	El usuario a su vez también agrega en el formulario la opción de búsqueda en la Darknet.
	7	El sistema obtendrá la información cruda en base a lo solicitado en el formulario.
	8	El sistema filtrará, evaluará y analizará dichos datos para simplificarlos y ordenarlos.
	9	El usuario visualizará toda la información aportada por los motores de búsqueda para dichos datos, incluyendo contenido indexado de la Darknet.
Postcondición	Ninguna.	
Excepciones	Si en el formulario solo se rellena el apartado de ciudad y/o simplemente selecciona si desea o no buscar información en la Darknet se mostrará un mensaje de error indicando el porqué no se aceptó dicha solicitud de búsqueda.	

Tabla 5.14: Visualización de los datos indexados en la Darknet sobre los atributos rellenados.

CU-15	Descarga de la información resultante de la persona	
Dependencias	RF-16.	
Descripción	El sistema se comportará como se indica en el siguiente caso de uso cuando un usuario rellene el formulario de búsqueda de la persona descrita para a posteriori poder descargar la información generada por el sistema para su uso personal.	
Precondición	RF-15.	
Secuencia normal	Paso	Acción
	1	El usuario accede al sistema.
	2	El sistema muestra la página principal.
	3	El usuario solicita ir al apartado de búsqueda de personas.
	4	El sistema muestra el apartado de búsqueda de personas.
	5	El usuario rellena el formulario de búsqueda de personas y envía la petición.
	6	El sistema obtendrá la información cruda.
	7	El sistema filtrará, evaluará y analizará dichos datos para simplificarlos y ordenarlos.
	8	El usuario visualizará toda la información junto a una opción de descarga que deberá clicar.
	9	El sistema generará una recopilación de todos estos datos y los enviará al usuario.
	10	El usuario recibirá dichos datos y aceptará la descarga en su dispositivo.
Postcondición	Ninguna.	
Excepciones	Si en el formulario solo se rellena el apartado de ciudad y/o simplemente selecciona si desea o no buscar información en la Darknet se mostrará un mensaje de error indicando el porqué no se acepto dicha solicitud de búsqueda.	

Tabla 5.15: Descarga de la información resultante de la persona.

5.2– Descripción de diagramas de secuencia

El proyecto de `OsintSpector` se divide en dos partes principales:

- Análisis de Twitter de una cuenta objetivo.
- Búsqueda de la persona objetivo dado datos a priori por parte del usuario cliente(nombre, apellidos, nickname, correo electrónico, ciudad, teléfono, opción de búsqueda en la Darknet).

Por tanto, para declarar los distintos procesos y sus diagramas de secuencia se podría diferenciar desde el principio estas dos partes. Comencemos primero enumerando los procesos de la parte del **análisis de Twitter** para la creación de inteligencia a partir de sus tweets:

1. Envío por parte del usuario de la petición de análisis de cierta cuenta de Twitter.
2. Recolección de información de la cuenta objetivo y sus tweets.
3. Evaluación y filtrado de datos en crudo.
4. Procesamiento y análisis de los datos para generar información ordenada e inteligencia.
5. Difusión y visualización de dichos análisis para el usuario cliente.

Como una imagen vale más que mil palabras se procede a realizar un pequeño esquema en la Figura 5.2 para visualizar con más comodidad dicho diagrama de secuencias que hemos definido.

Para finalizar se definirá la otra parte principal, la cual consiste en la **búsqueda y recopilación de información de una persona objetivo**. El proceso para una búsqueda con todas las opciones disponibles mencionadas (nombre, apellidos, nickname, etc.) sería:

1. Envío por parte del usuario de la petición de búsqueda de información de cierta persona.
2. Recolección de información de la tríada nombre-apellidos-ciudad, usando motores de búsqueda e información del INE.
3. Recopilación de motores de búsqueda donde esté registrado el nickname.
4. Recopilación de información sensible filtrada al público sobre el correo y el teléfono.
5. Sitios de la Darknet donde el nombre-apellidos, nickname, correo y teléfonos hayan sido indexados.



Figura 5.2: Diagrama de secuencia para el análisis de Twitter.

6. Análisis de la información para crear información ordenada e inteligencia
7. Puesta en escena y visualización de la información, simple, ordenada y visualmente agradable para el usuario con gráficos incluidos.

Como se dijo anteriormente es mucho más fácil ver este diagrama de secuencias con un pequeño esquema (ver Figura 5.3) que resume todos estos procesos de la parte de la búsqueda de persona.



Documentacin pclass/img/diagramas/DiagramaDeSecuenciaPersonaCropped.png

Figura 5.3: Diagrama de secuencia para la búsqueda y recopilación de información de una persona objetivo.

5.3— Modelo de datos

El modelo de datos de nuestra solución expone las relaciones entre las entidades disponibles, y los atributos que contienen en cada caso. En la Figura 5.4 se muestra el modelo de datos completo. Para un visionado más sencillo, procedemos a separar el Modelo de Datos en distintas Tablas que explicaremos a continuación.

La Figura 5.5 muestra la relación de los datos en el apartado de análisis de Twitter. Donde la información recopilada es compuesta por muchos tweets con sus respectivos atributos. A partir de esta información recolectada se crean otras entidades, las cuales son el wordcloud, comunidades, interacciones, localizaciones y sentimientos.

La Figura 5.6 muestra la relación de los datos de la sección de búsqueda de personas, en general se divide en cinco grandes grupos, los cuales son las de NAC(nombre-apellido(s)-ciudad), nickname, email, número de teléfono y la recopilación de la darlmet de estos atributos. Existen otras entidades que sirven de entidades auxiliares, como el de User-Agents, Proxies, Api-Keys y Motores Nickname. La base de toda esta relación de datos es la entidad de Datos personales, es desde donde se ramifican los cinco grandes grupos antes mencionados.

Documentación

Figura 5.4: Modelo de datos.

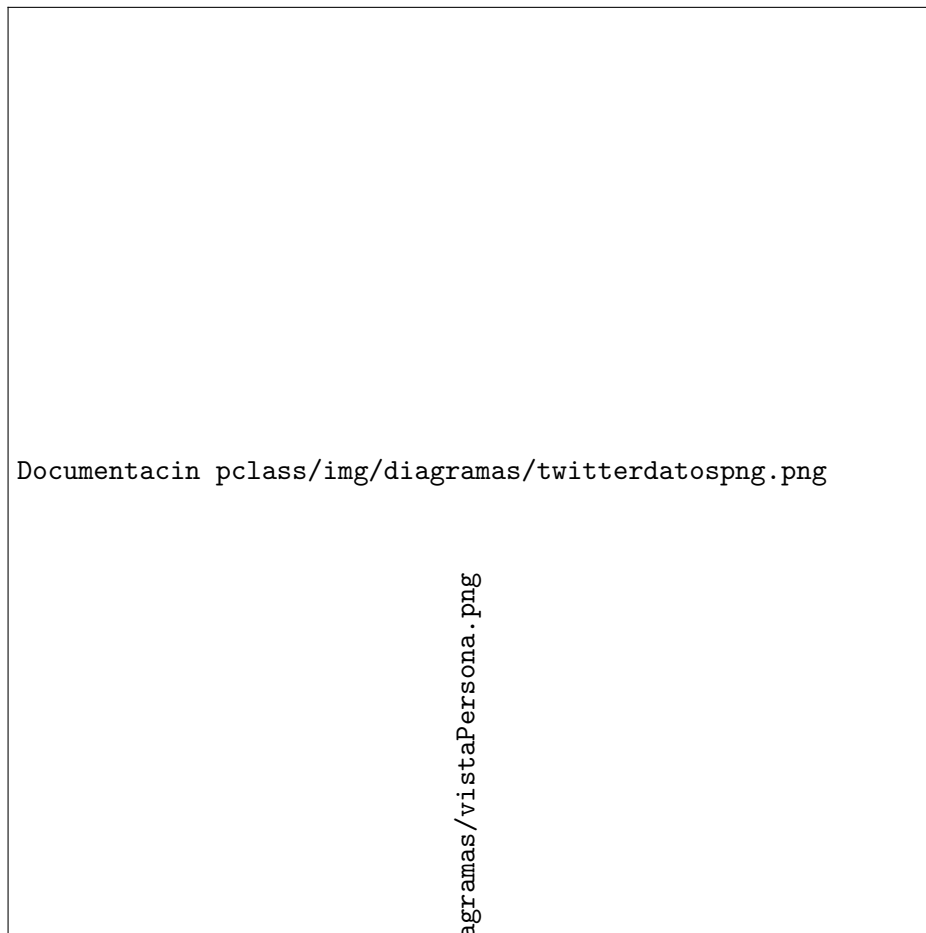


Figura 5.5: Vista análisis de Twitter.



Figura 5.6: Vista búsqueda de Persona.

5.4– Arquitectura del sistema

En esta última sección del capítulo se expondrá la arquitectura del sistema. El sistema cuenta con un modelo de cliente/servidor. Por ello se puede dividir dicho sistema en dos partes principales, el front-end y el back-end.

El front-end, también conocido como parte del cliente o interfaz de usuario, se refiere a la capa del sistema que se encuentra del lado del usuario. Es la parte visible y tangible con la que interactúan los usuarios finales, es decir, donde se muestra la información de una manera clara y sencilla de entender para el susodicho usuario final.

En nuestro caso el front-end será un apartado sencillo donde el usuario pueda navegar cómodamente, pueda visualizar las vistas creadas desde la gestión del frontal y pueda realizar por tanto las funciones expuestas en apartados anteriores.

Al otro lado, se encuentra el back-end, también conocido como parte del servidor, esta capa del sistema se dedica al recibimiento de instrucciones por parte del front-end, al almacenaje/extracción de la información para la gestión de dichas instrucciones dadas, al procesamiento y finalmente a la transmisión de la respuesta al front-end. Por ende, esta parte implica funciones o partes vitales tales como el controlador de vistas/-frontal, los propios servicios internos (que procesan y ejecutan la mayoría de funcionalidades disponibles en el sistema) o el comunicador con los servicios externos que usaremos (usando un intermediario en dicha comunicación).

El back-end en nuestro caso poseerá distintos apartados. Sobre todo se diferencian en dos grandes grupos, la parte de análisis de Twitter y la parte de búsquedas de personas, ambos servicios internos requerirán de funciones u otros servicios auxiliares implementados dentro del propio back-end. Como por ejemplo en el apartado de búsquedas de personas se utiliza un sistema de gestión de proxies necesarios para la búsqueda de información en motores de búsqueda. Por otro lado, en el apartado de análisis de Twitter, necesitaremos desplegar un modelo de IA basado en análisis de sentimientos para clasificar los tweets en los distintos tipos de sentimientos descritos anteriormente. Igualmente, ambos apartados requerirán de múltiples fuentes y servicios externos de Internet, como bases de datos, motores de búsquedas, gestores... Todo ello usando en las comunicaciones un intermediario. En la Figura 5.7 se puede ver a grosso modo las distintas partes lógicas y su comunicación entre ellas.



Figura 5.7: Arquitectura del sistema.

En cuanto a la ubicación de la arquitectura del back-end, toda ella estará ubicada en el mismo lugar, con el fin de minimizar los riesgos o problemas que puede suponer una comunicación a través de la red entre los servicios internos y auxiliares que existen. La única salida a Internet será siempre usando un mediador como se ha comentado anteriormente.

CAPÍTULO 6

Implementación y pruebas

En este capítulo de implementación se realizará a cabo el desarrollo a partir de lo indicado en el análisis y diseño que se ha planteado anteriormente. Además de dicha implementación se ejecutarán pruebas para comprobar y corroborar el correcto funcionamiento del sistema.

6.1– Arquitectura tecnológica

Como se propuso anteriormente nuestra arquitectura será un modelo cliente/servidor. Por ello, se describirá las tecnologías implementadas en ambas partes así como su conexión entre ellas, empezando por la parte del cliente:

6.1.1. Tecnologías usadas en el front-end

Primero se describirán los lenguajes implementados:

- **HTML5** es la última versión del lenguaje de marcado utilizado para estructurar y presentar contenido en la web, que ofrece mejoras significativas en términos de semántica, multimedia y accesibilidad..
- **CSS3** es el lenguaje de estilos que permite personalizar y embellecer las páginas web con efectos visuales y diseños modernos.
- **Javascript** como lenguaje de programación utilizado para proporcionar funcionalidades interactivas y dinámicas al sistema en el lado del cliente.
- **Jinja2** es un motor de plantillas en Flask, del cual se hablará más adelante, que permite generar contenido dinámico en la parte frontal de las aplicaciones web. Permite combinar datos con plantillas

HTML utilizando una sintaxis sencilla, lo que facilita la creación de páginas web interactivas y reutilizables. Además permite la recepción de información de datos generados en el back-end dependiendo de la solicitud del cliente.

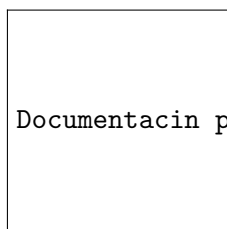


Figura 6.1: HTML5.

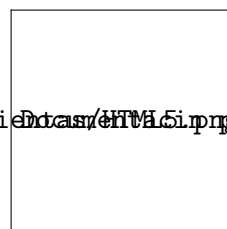


Figura 6.2: CSS3.

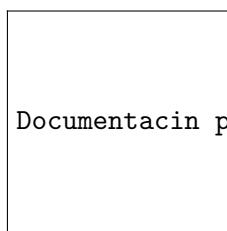


Figura 6.3: Javascript.

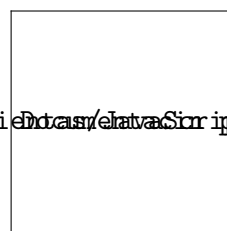


Figura 6.4: Jinja2.

En OsintSpector no se usará ningún framework para la parte del cliente, además, la comunicación entre cliente-servidor será gracias al framework usado en el back-end que se explicará en la siguiente subsección de este capítulo 6.1.2.

Ahora se describirá todas las librerías usadas para mejorar y facilitar el desarrollo del código del front-end:

- **Bootstrap** es una biblioteca multiplataforma o conjunto de herramientas de código abierto para CSS y HTML para diseño de sitios y aplicaciones web, facilitando así tener un diseño estructurado y cómodo a la hora de programar.
- **JQuery** es una biblioteca multiplataforma de JavaScript, la cual permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos...
- **Apexcharts** es una biblioteca de Javascript que se usará para generar gráficas agradables visualmente. Además se utilizará también

en la parte de la visualización de la información en el resultado de búsqueda de personas para generar dashboards [7], de esta forma la información visual será compacta y cómoda para el estudio por parte del usuario.

- **Chart.js** es otra biblioteca de Javascript que se implementará para generar gráficas agradables visualmente.
- **Vis-network.js** es un apartado de la biblioteca de **Vis.js** de Javascript de visualización de grafos. Ésta puede mostrar redes compuestas por nodos y conexiones, es fácil de usar y admite formas personalizadas, estilos, colores, tamaños, imágenes, entre otros. Además de permitir que el usuario interactúe con ella, se usará para los grafos de interacciones y de comunidades.
- **Luxon** es una biblioteca para facilitar el manejo de fechas y horas en Javascript. Se aplicará para el apartado de localizaciones.
- **Leaflet** es una biblioteca JavaScript de código abierto que se utiliza para crear aplicaciones de mapas web, se empleará para desplegar un mapa del mundo interactivo con las localizaciones donde tuiteó el usuario objetivo.

Documentacin pclass/img/herramientas/Bootstrap.png

Figura 6.5: Bootstrap.

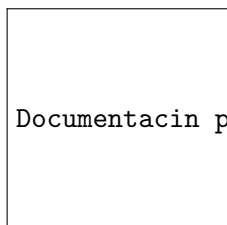


Figura 6.6: JQuery.

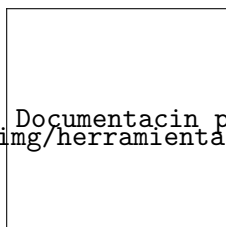


Figura
ApexCharts.

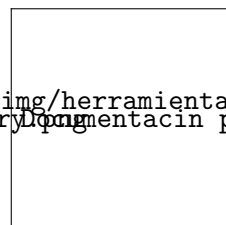


Figura 6.8: ChartJS.

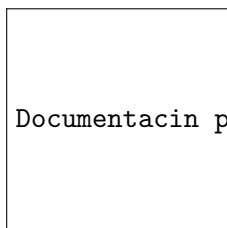


Figura 6.9: Vis-
NetworkJS.

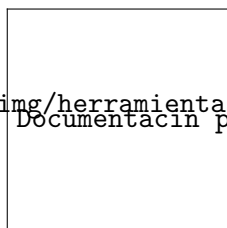


Figura 6.10: Luxon.

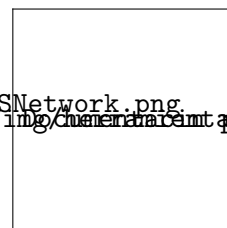


Figura 6.11: Leaflet.

6.1.2. Tecnologías usadas en el back-end

Primero se describirá el framework utilizado para el desarrollo del back-end y sus diferencias con otros parecidos a él. Más tarde se mostrarán los lenguajes y librerías implementados en el back-end.

Como se mencionó en el otro apartado, el framework de back-end utilizado será Flask, en concreto la versión 2.1 [6].

¿Qué es Flask?

Flask es un microframework web escrito en Python que se centra en la simplicidad y la extensibilidad. Utiliza el protocolo HTTP para manejar solicitudes y respuestas, y proporciona un enrutamiento eficiente y flexible para gestionar las diferentes URL de una aplicación web. Además, ofrece un potente sistema de plantillas llamado Jinja2, explicado anteriormente, para generar contenido dinámico basado en HTML. Con su arquitectura modular y su comunidad activa, Flask es una opción popular para el desarrollo de aplicaciones web, especialmente para proyectos más pequeños y ágiles. Por todo ello, Flask es el microservicio perfecto para implementar en OsintSpector.

A continuación se pasará a explicar el lenguaje usado en dicho back-end:

- **Python** es el lenguaje de programación de alto nivel utilizado en nuestro desarrollo web. Flask, está construido con Python, debido a que Python en Flask ofrece una amplia variedad de bibliotecas y módulos que facilitan tareas comunes en el desarrollo web, como el manejo de solicitudes y respuestas HTTP, el enrutamiento de URLs y la generación de contenido dinámico. Además en nuestro caso nos facilita la tarea de scraping [10] de la cuál se hablará más detenidamente en el siguiente apartado.

Finalmente las librerías y los paquetes o módulos más importantes utilizados, clasificados por su función, son:

□Documentación pclass/img/herramientas/Python.png

Figura 6.12: Python.

- **Gestor de Paquetes**
 - Pip.
- **Web Scraping y obtención de información**
 - requests, fake headers, SnScrape, BeautifulSoup, playwright, playwright stealth.
- **Tratamiento de los datos y la información**
 - Pandas, nltk, tempfile, wordcloud, PyTorch, networkx.
- **Control de subprocesos, corutinas, threads y procesos asíncronos**
 - subprocess, threading, multiprocessing, asyncio, time.

6.2– Herramientas adicionales

Para facilitar el diseño, desarrollo, tratamiento y despliegue del código, se usan las siguientes herramientas:

- **Github**, plataforma de desarrollo colaborativo que proporciona un servicio basado en Git, usada para el control de versionado del código en un repositorio público y que proporciona muchas herramientas útiles para el control del código. Se cuenta con la versión PRO, de manera gratuita, al ser estudiante universitario.
- **Visual Studio Code**, editor de código fuente para cualquier sistema operativo con extensiones que facilitan desarrollar y depurar el código. Cuenta con Licencia MIT, haciendo su uso gratuito. Usado también para la conexión con el controlador de versionado de Github.
- **Lucidapp**, herramienta en línea que permite crear y colaborar en la creación de diagramas de forma visual y sencilla.
- **Moqups**, aplicación web gratuita que permite realizar diferentes tipos de diagramas.

6.3– Esquema tecnológico

El siguiente esquema muestra la relación entre todas las herramientas expuestas anteriormente.



Figura 6.13: Esquema de tecnologías usadas.

6.4– Detalles de la implementación

En la sección de detalles de implementación se definirá la estructura de paquetes, el contenido de sus elementos más importantes, los elementos más reseñables del desarrollo back-end y front-end y finalmente un apartado donde se describen las vistas que tiene la aplicación (en el manual de usuario 8.2 también se pueden ver, pero en esta sección se describirán las vistas centrándose un poco más en el apartado de la implementación). Una vista global del proyecto antes de entrar en detalles sería la siguiente:

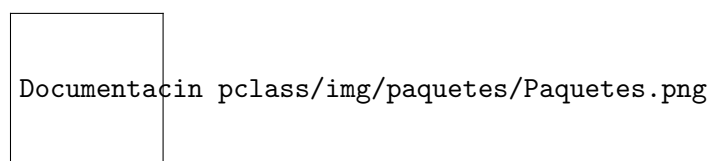


Figura 6.14: Vista global de los paquetes.

6.4.1. Estructura de paquetes

Los paquetes de la aplicación han sido distribuidos en cuatro carpetas principales, el archivo App.py. Las dos primeras carpetas que se describirán forman parte del back-end, mientras que las dos últimas forman parte del front-end.

Paquete searchScripts

El primer paquete contiene la mayoría de servicios internos y casi toda la comunicación con los servicios externos, de ahí el nombre de *SearchScripts*. Esto es debido a que se encarga de las búsquedas de información en servicios externos, una vez obtenida dicha información también se encargará de la parte de servicios internos dedicada al análisis, procesamiento y clasificación de estos datos obtenidos.

Por ello se puede ver que dentro de esta carpeta 6.15 se dividen en otras dos subcarpetas llamadas *buscarPersona* y *twitter*, la primera se encarga del apartado de búsquedas de personas; a su vez esta carpeta se subdivide por cada atributo del formulario de búsquedas de personas ya mencionado. Esa división se ha hecho así para llevar un control más modular y cómodo a la hora de desarrollar y sobre todo del mantenimiento del código. Finalmente en la carpeta de *twitter* se haya el módulo de *busquedaTwitter.py*, dicho módulo contiene una única clase que se entrará en detalles en el siguiente apartado, pero en general es la encargada de todo

el apartado del servicio interno de análisis de Twitter, tanto recopilación, como procesamiento de datos, como generación de inteligencia y finalmente envió de los resultados al controlador de vistas.

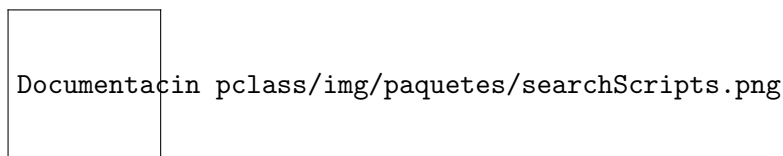


Figura 6.15: Paquete searchScripts.

Paquete utils

Este segundo paquete contiene el resto de servicios internos y lo que quedaba de la parte de conexión con servicios externos. Esto es así porque este paquete se dedica a la creación de servicios auxiliares o pequeñas funciones que se utilizan en la mayoría de los servicios internos explicados arriba. El servicio auxiliar más importante es el gestor de servidores Proxies que se actualiza cada 30 minutos y que se usará para entrar en todo servicio externo donde debamos de *scrapear* [10]. En el siguiente apartado 6.4.2 del capítulo se explicará brevemente como se consiguen estos servidores que actúan de intermediarios.

Finalmente se puede ver el módulo *commonFunctions.py* que almacena toda función auxiliar como comentamos anteriormente y también un archivo llamado *userAgentsList.txt* que almacena un total de mil userAgents, como se comentó anteriormente se utiliza para crear cabeceras distintas por cada request que se realice a un servicio externo.

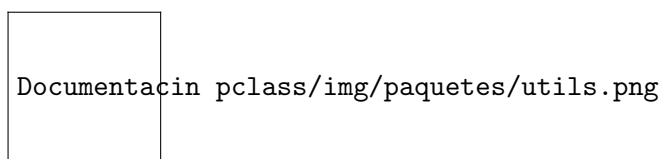


Figura 6.16: Paquete utils.

Paquete templates

Este tercer paquete es una carpeta que contiene archivos HTML utilizados para construir la interfaz de usuario de la aplicación web. Existen varios archivos base que definen la estructura común de todas las páginas. Otros archivos HTML más complejos se basan en el archivo base y añaden funcionalidades adicionales, como las vista de resultados.

En algunos de estos archivos HTMLs más complejos, se importan datos del backend de la aplicación. Esto se logra utilizando gracias al frontal usando Jinja2, un motor de plantillas que permite la comunicación entre el front-end y el back-end. Jinja2 permite insertar código Python en los archivos HTML, lo que facilita la transferencia de datos desde el back hacia el front. Al utilizar marcadores especiales de Jinja2, los datos del back-end pueden ser mostrados dinámicamente en el front-end. Esto permite que las páginas web respondan y se actualicen según los datos proporcionados por los servicios internos del back.

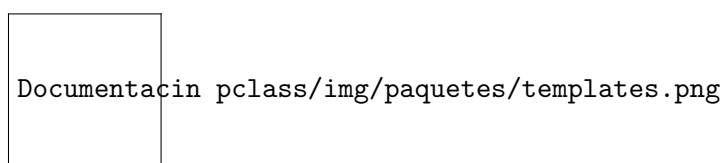


Figura 6.17: Paquete templates.

Paquete static

En este último paquete llamado *static* de Flask se almacenan archivos estáticos como CSS, imágenes y archivos JavaScript.

Estos archivos no requieren procesamiento o generación dinámica y se sirven directamente al cliente sin intervención del servidor.

Algunas librerías del front en lugar de usar servidores CDN se almacenan localmente, como el caso de leaflet o tabulator, por tanto se deben almacenar en esta carpeta.

A su vez también podemos ver que existen distintos archivos JavaScript para cada vista compleja, como la de análisis de Twitter o las de búsqueda de persona.

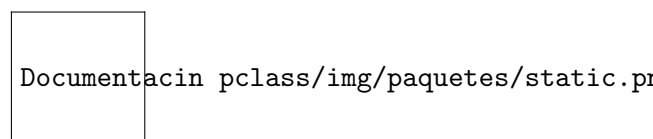


Figura 6.18: Paquete static.

App.py

Es el archivo de entrada principal que actúa como el punto de partida de la aplicación web. En este archivo, se definen las rutas y se configura el enrutamiento de solicitudes HTTP. También se especifican los controladores (llamados *views* en Flask) que manejan las solicitudes y generan las respuestas correspondientes. Además, se pueden definir configuraciones adicionales de la aplicación, como la conexión con otros módulos y paquetes. En resumen, App.py es el corazón de OsintSpector, donde se define su comportamiento y se inicia su ejecución.

6.4.2. Código desarrollado

A continuación, se va a mencionar algunos elementos destacables del código desarrollado y finalmente un repaso a los problemas encontrados y su solución.

Empezando por el back-end hay ciertos servicios interesantes que mencionar:

- Para el análisis de Twitter, se tiene una clase padre que contiene seis clases hijas, una por cada funcionalidad implementada en este apartado más la propia de scrapear los tweets en bruto, lo más relevante a comentar de este apartado es:

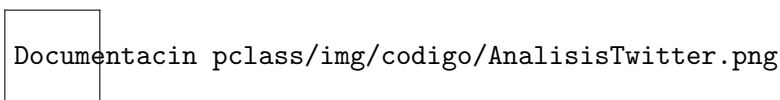


Figura 6.19: Clase Análisis de Twitter.

- En el apartado de análisis de sentimientos se ha desplegado un modelo de IA llamado xlm-emo-t [8] que se ejecuta junto a nuestro proyecto como uno de nuestros servicios internos, el tipo de conexión usada para comunicarse entre el modelo y el servicio interno que tiene los datos en bruto a analizar es mediante el paquete de transformers [24], el cual proporciona herramientas y APIs fáciles de controlar de cara a la entrada y salida de información, en la Figura 6.20 se puede comprobar dicha inicialización del modelo. Apenas requiere una cantidad significativa de recursos de computación y memoria así que no hay problema en desplegarlo localmente, más aún si se posee una gráfica, ya que en el código está implementado la preferencia a usar una GPU dedicada, sino existiera, se usaría la CPU.

La clasificación de estos tweets se hacen en paralelo mediante el uso de hilos, pero lo importante a exponer es el proceso de clasificación de cada tweet, como se puede apreciar en la Figura 6.21, el cual es:

1. **Tokenización:** El tweet se procesa utilizando un *tokenizer*, que divide el texto en unidades más pequeñas llamadas tokens. Se utiliza el tokenizer definido en *self.tokenizer* y se pasa el tweet como entrada. Los tokens resultantes se envían al dispositivo disponible.
2. **Clasificación:** Los tokens del tweet se pasan al modelo de clasificación definido en *self.model*. Los resultados de la clasificación se almacenan en *outputs*.
3. **Obtención de *logits*:** Los *logits* son los vectores con las predicciones en crudo generados por el modelo de clasificación. Se extraen los *logits* de *outputs*.
4. **Cálculo de probabilidades:** Se aplica la función softmax [37] a los *logits* para obtener las probabilidades de cada clase. Las probabilidades se calculan utilizando la función de la librería Torch *torch.softmax* y se convierten a una lista.
5. **Etiquetas de emociones:** Se obtienen las etiquetas de las emociones del atributo *id2label* del modelo.
6. **Resultado de la clasificación:** Se crea una lista de diccionarios llamada *sentiment_result*, donde cada diccionario contiene la etiqueta de la emoción y el puntaje correspondiente.
7. **Emoción con mayor probabilidad:** Se selecciona la emoción con el puntaje más alto utilizando la función *max* y se almacena en *top_sentiment*.
8. **Actualización de los tweets con la emoción más alta:** Se compara la emoción seleccionada, *top_sentiment* con las emociones predefinidas en *self.emotions*. Si la emoción seleccionada es una de las emociones predefinidas y tiene un puntaje mayor que el puntaje almacenado previamente en *self.tweets_with_top_emotion*, se actualiza el puntaje y se guarda el tweet y el enlace asociados con esa emoción.
9. **Retorno de resultados:** Finalmente, se retorna la etiqueta de la emoción con mayor probabilidad, *top_sentiment['label']*.

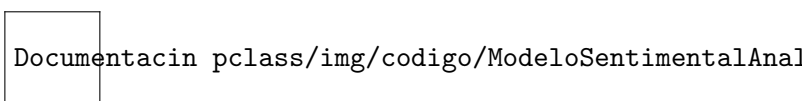


Figura 6.20: Inicialización clase Sentimental Analysis.

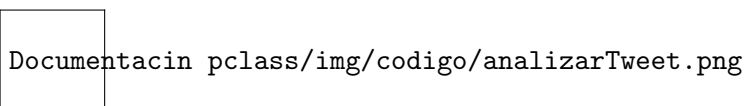
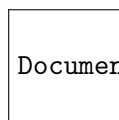


Figura 6.21: Proceso de análisis de sentimientos de cada tweet.

- o En el apartado de grafo de comunidades, para crear dicho grafo se tiene que estudiar todas las conexiones entre los usuarios (nodos) junto a sus aristas (cuando se mencionan a distintos usuarios en un mismo tweet del username objetivo). Por ello para la creación de estas divisiones en comunidades se usa el algoritmo de *Clauset-Newman-Moore greedy modularity maximization* [11] del paquete de networkx [21]. Los principales pasos que realiza resumidamente son los siguientes:
 1. **Inicialización:** Se crea un objeto GrafoComunidad y se guarda el DataFrame que contiene los datos de interacciones entre usuarios.
 2. **Conteo de interacciones:** Se cuenta el número de interacciones entre los usuarios en el DataFrame y se almacenan en un diccionario.
 3. **Creación del grafo:** Se crea un grafo utilizando las interacciones contadas. Cada usuario es un nodo y las interacciones entre usuarios son los enlaces (edges) del grafo.
 4. **Obtención de comunidades:** Se utilizan algoritmos de detección de comunidades en el grafo para identificar grupos de usuarios que interactúan más entre sí que con usuarios externos. Las comunidades encontradas se almacenan en una lista.
 5. **Análisis del grafo:** Se realizan diversos cálculos y análisis en cada comunidad, incluyendo densidad, diámetro, excentricidad, centro, grado medio, clustering medio, cohesión y betweenness centrality.
 6. **Conversión a formato JSON para el front-end:** Se convierte el grafo y las comunidades en un formato JSON compatible con la visualización de redes utilizando la biblioteca Vis-NetworkJS. Cada nodo y enlace se representa

con atributos específicos, como por ejemplo el color, id nodo, identificador del nodo origen y del nodo destino para los enlaces, etc.

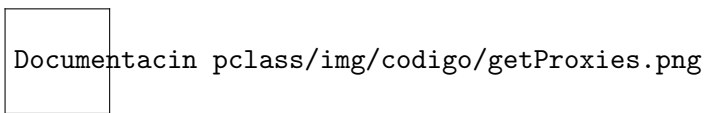


pclass/img/codigo/GrafoComunidades.png

Figura 6.22: Clase Grafo comunidades.

- Para la búsqueda de personas:
 - El servicio interno de gestor de servidores proxies es un subproceso basado en multiprocessing para no ralentizar el completo funcionamiento del rendimiento de la aplicación. Este gestor funciona gracias al uso de scraping de la página Proxylist [22] cada 30 minutos. Una vez obtenido todos los proxies se comprueba con varios filtros si son funcionales en cada una de las páginas donde usamos técnicas de scraping, para así usarlos como intermediarios cuando se realicen las búsquedas y así la propia IP del servidor no se ve comprometida a ser bloqueada en aquellos lugares donde se scrapea. Los métodos principales de la clase donde se implementa este servicio de gestor de proxies son:
 1. **getRawProxies**: Realiza una solicitud a un sitio web que proporciona proxies gratuitos actualizados y extrae la lista de proxies disponibles de tipo elite [33].
 2. **extractINE y extractAhmia**: Verifican la viabilidad de los proxies al realizar solicitudes a sitios web específicos (en el caso de OsintSpector al INE y a Ahmia). Si la respuesta indica que el proxy es válido, se agrega a la lista de proxies funcionales respectivamente.
 3. **extractINEFiltrado**: Utiliza la biblioteca Playwright para realizar pruebas más exhaustivas de los proxies con la web del INE donde se extrae la información del nombre y/o apellido(s). Crea un navegador y realiza solicitudes utilizando cada proxy para determinar si funcionan correctamente.
 4. **obtenerProxiesFiltrados**: Coordina las operaciones anteriores para obtener y filtrar los proxies. Realiza filtrados iniciales utilizando los métodos *extractINE* y *extractAhmia*, y luego realiza un filtrado adicional utilizando Playwright con el método *extractINEFiltrado*. Los proxies filtrados se

guardan en archivos de texto separados por plataforma (INE y Ahmia) para su uso posterior.



Documentación pclass/img/codigo/getProxies.png

Figura 6.23: Clase del gestor de proxies.

- o En la búsqueda de información sobre el correo electrónico y el número de teléfono se consultarán las bases de datos de Intelx [38] y Have I Been Pwned (HIBP) [26], ambas usando sus APIs correspondientes.
- o Para la recopilación de información indexada en la Darknet de todo atributo rellenado en el formulario se usará Ahmia [28]. Ahmia.fi es un motor de búsqueda en el Internet superficial sobre la web oscura que se basa en el proyecto de código abierto Ahmia, este proyecto se especializa en indexar y buscar contenido específicamente de la red Tor, que es una de las redes más conocidas de la web oscura. Por tanto la recopilación de información se usará haciendo scraping con este motor de búsqueda. Algunas características principales de éste módulo son:
 1. **Selección aleatoria de User-Agent:** La función `randomUserAgent(filename)` elige de forma aleatoria un User-Agent de un archivo de texto que contiene una lista de User-Agents. Esto se utiliza para simular diferentes navegadores o dispositivos al hacer solicitudes web, lo que puede ayudar a evitar bloqueos o restricciones.
 2. **Selección aleatoria de servidor proxy:** La función `randomProxyServer(filename)` elige de forma aleatoria un servidor proxy de un archivo de texto que contiene una lista de servidores proxy. Esto se utiliza para enrutar las solicitudes web a través de un proxy, lo que permite ocultar la dirección IP real del cliente y proporcionar cierto nivel de anonimato.
 3. **Clase *AhmiaScraping*:** La clase *AhmiaScraping* contiene métodos para realizar solicitudes web, obtener el contenido HTML de una página web y extraer datos específicos de interés. El método *make_request* realiza una solicitud GET a una URL especificada utilizando un servidor proxy y un User-Agent seleccionados aleatoriamente. El método *getAhmiaHtml* obtiene el contenido HTML de una URL utilizando *make_request* y devuelve el objeto

BeautifulSoup resultante. El método *parseHTML* analiza el contenido HTML obtenido de una URL utilizando *getAhmiaHtml* y extrae información relevante de los resultados de búsqueda en un formato específico.

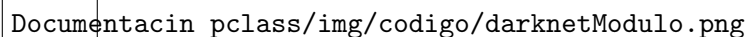
A small rectangular box containing the text "Documentacin pclass/img/codigo/darknetModulo.png". This is likely a placeholder for a screenshot of the code for the Darknet module.

Figura 6.24: Módulo para la recopilación de información en la Darknet.

Para el front-end hay ciertos aspectos en el código interesantes que mencionar:

- Para la muestra gráfica sobre los resultados de búsqueda del correo electrónico y el número de teléfono se ha implementado los dashboards de la librería de Apexcharts. Estos dashboards son muy vistosos y organizan muy bien los gráficos para la comodidad del usuario, pero además son cómodos de manipular por parte del desarrollador. Un ejemplo sería la Figura 6.25:

A small rectangular box containing the text "Documentacin pclass/img/codigo/EjemploDashBoard.png". This is likely a placeholder for a screenshot of the code for the Dashboard example.

Figura 6.25: Módulo para la recopilación de información en la Darknet.

- Todos los grafos que se muestra al usuario final son totalmente manipulables e interactivables. Esto es gracias a que se generan los objetos en el back-end usando la librería de networkx, el frontal envía estos datos de forma entendible para el front-end, finalmente este último puede generar los objetos finales usando la librería de NetWorkJS, Figura 6.9, con todas sus funcionalidades de click, arrastre, zoom, etc.
- El uso de las plantillas de Jinja2 nos permiten crear una base HTML donde registrarse y basarse para formar un esqueleto sólido y compartido entre todas las distintas vistas, que obviamente cada una es personalizada, pero gracias a tener una base es muy fácil de modificar o mantener de cara a un futuro, ya que sabemos las partes principales de dichas vistas. En el caso de OsintSpector existen dos tipos distintos de base, la primera es la parte de documentación previa y navegación por la página principal y formularios, la segunda es la de resultados búsqueda de personas y resultado de análisis de la cuenta de Twitter.

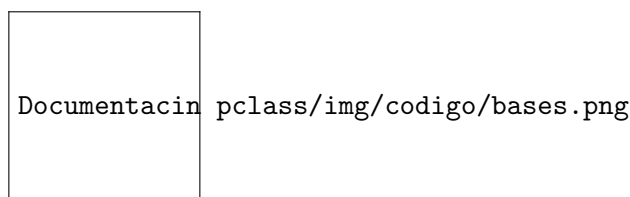


Figura 6.26: Distintas plantillas base de OsintSpector.

Problemas y cambios en el camino

1. Google no permite scrapear Google Results, por ello se pretendría usar la API de google searchs, al implementarla se comprobó que los resultados de la API de búsqueda y de Google.com no son idénticos [5]. Por tanto se hizo un pequeño estudio sobre que se podría hacer, el resultado fue usar una API de terceros con pruebas mensuales gratuitas. En concreto se eligió la API de ScaleSerp [35] por la estructura de los resultados que da esta API, perfecta para anidar y manipular.
2. El gestor de intermediarios a veces daba falsos positivos, servidores proxies que deberían de funcionar en ciertos servicios externos realmente en producción no funcionaban, por ello se tuvo que remodelar el filtrado de ellos para poder asegurar que no existan estos falsos positivos.
3. Al principio los resultados dados por HIBP para el correo y el teléfono eran recopilados gracias a hacer scraping sobre la web, pero, a partir de enero de 2023 se mejoró la detección de bots y apenas ningún proxy funcionaba de intermediario. Además que aunque se usaran User-agents aleatorios, cabeceras cambiadas y aleatorias, modo *stealth* de Playwright y usando todas las pautas posibles para no ser detectados solía dar fallo. Finalmente la solución fue comprar una API key de pago mensual para dicho servicio y reestructurar todo el código para obtener la información desde ahí. Cuando ocurría dichos errores, estas eran las respuestas de estado, algunas veces si dejaba entrar en la propia web pero en el momento de solicitar información de un correo o teléfono ya prohibía dicha acción, como se puede ver en la imagen 6.27. Otras veces HIBP detectaba el bot en el primer momento de la solicitud como se puede ver en la imagen 6.28:

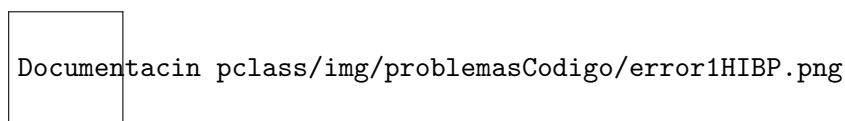


Figura 6.27: Error Scraping solicitud de búsqueda HIBP.

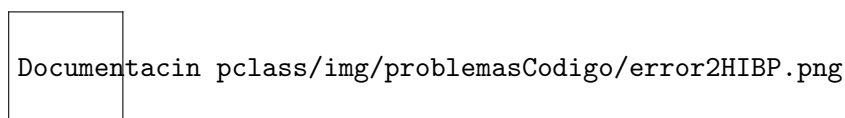


Figura 6.28: Error Scraping 403 HIBP.

4. Para la obtención de tweets dado un username se decidió de primeras usar la API gratuita de Twitter, la empresa proporcionó una API key con acceso elevado y a esperas aún de la respuesta de Twitter para conseguir el acceso académico, la diferencia principal es que en el acceso académico podemos consultar todo el historial de tweets del usuario, pero con el elevado solo los de la última semana. Igualmente el 31 de marzo cambió la política de esta API [1] y pasó a ser de pago mensual por un valor de 100 dólares, eliminando toda posibilidad de tener una cuenta académica [2]. Finalmente se decidió que para la recopilación de tweets se scrapearán sin usar la API, sino la librería de SnScrape [4], con la cual se podía obtener todos los tweets que quisiéramos de un usuario junto a información sensible de cada uno de ellos.
5. El 31 de abril la forma desde la que se scrapeaban los tweets usando Snsrape, mediante el buscador de Twitter sin iniciar sesión con ninguna cuenta, se cerró por decisión de Twitter [3], seguramente porque sabían que el tráfico mayoritariamente eran bots. Por ello se usó otro tipo de obtención de tweets que también proporciona la librería, pero, en este caso el máximo de tweets que se pueden conseguir oscila entre dos mil y tres mil doscientos, es decir, hay una reducción considerable en cantidad de información recopilada, pero es un sacrificio que se tuvo que hacer a la fuerza.

6.4.3. Vistas

Este último apartado será más visual y se mostrarán las distintas vistas que existen en OsintSpector, para más detalles acerca de las vistas se puede encontrar en el manual de usuario 8.2.

Página principal y documentación previa

La página principal es la que conecta con todas las distintas funcionalidades que existen en el sistema, las cuales son la parte de documentación previa, análisis de una cuenta de Twitter y la última parte que se trata de la búsqueda estándar de personas. Además se podrá ver la navbar superior y un footer, como se puede ver en la Figura 8.5.

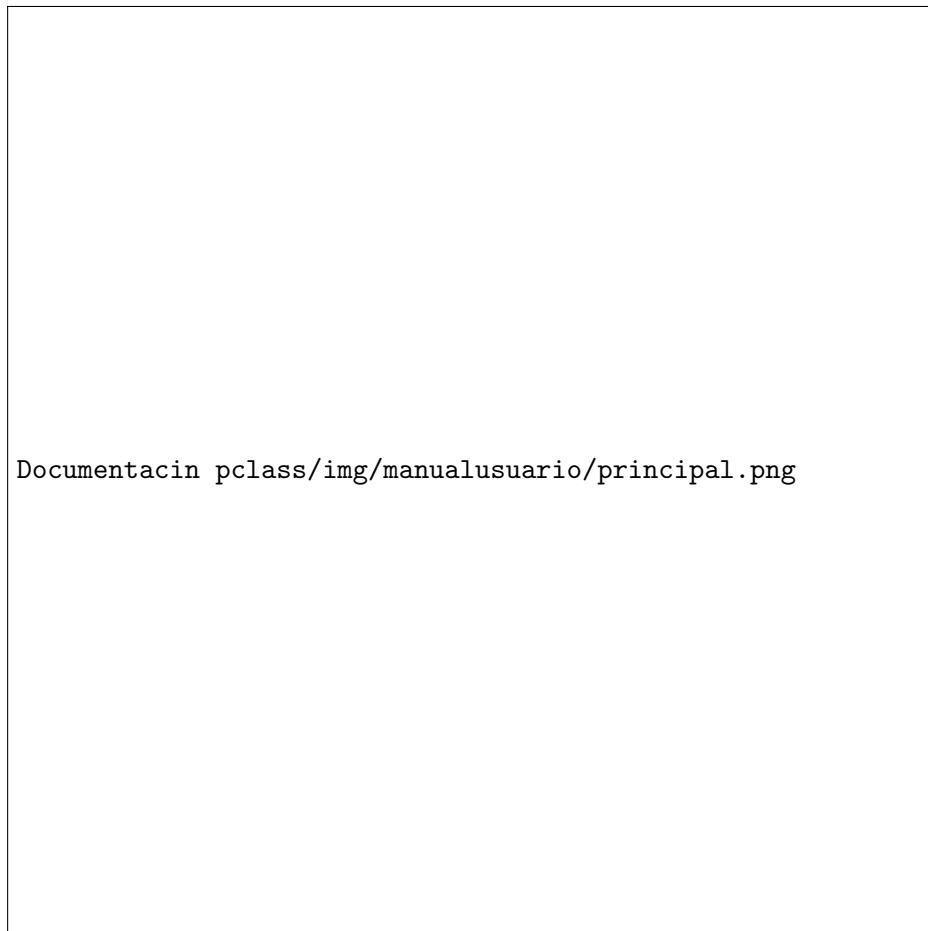


Figura 6.29: Página principal.

En la navbar superior se puede ver que en la parte izquierda existe un apartado que se llama *Osintspectator* junto al logo de nuestro proyecto, al clicar en el siempre se redireccionará al menú principal, a su vez en la parte derecha se encuentran los dos apartados de la documentación previa los cuales se tratan de la sección *Sobre nosotros* y *De qué se trata este proyecto*.

Al clicar en *Sobre nosotros* se explica un poco quienes fueron los desarrolladores del proyecto y una breve explicación de su motivación, como se puede apreciar en la Figura 6.31.



Figura 6.30: Página de sobre nosotros.

Finalmente la parte de *De qué se trata este proyecto* explica un poco como funcionan los dos distintos tipos de búsquedas y detalles sobre ellas.

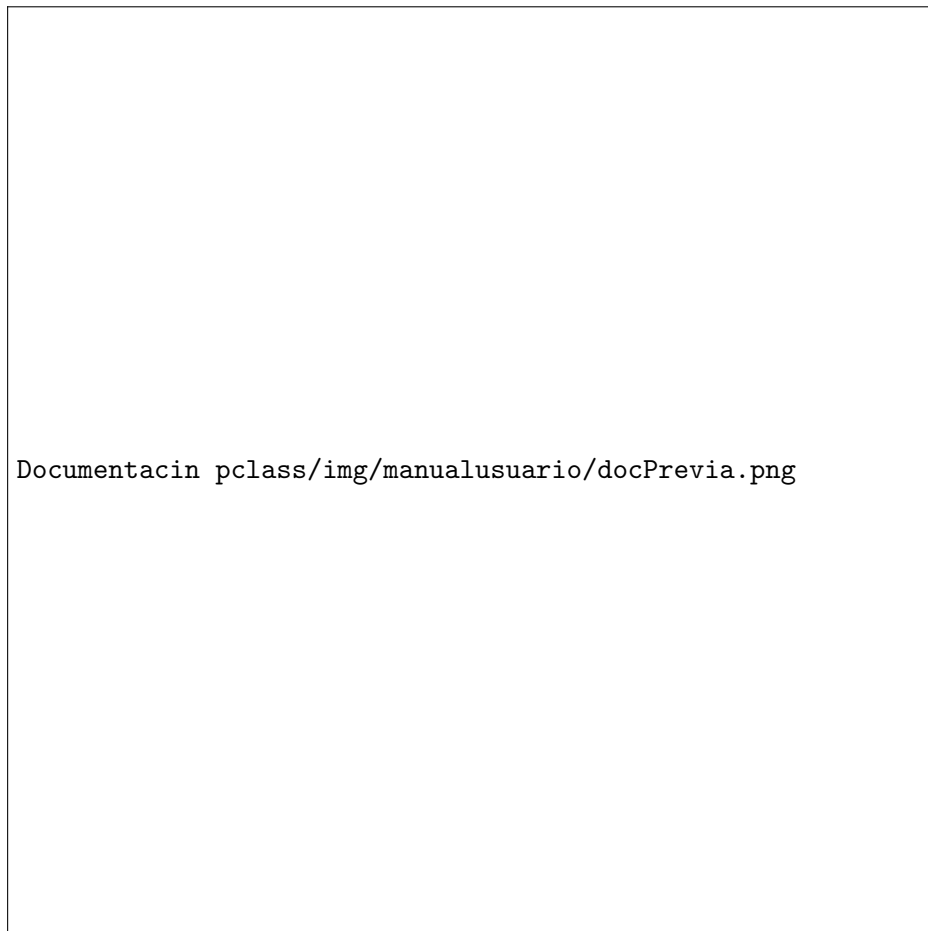


Figura 6.31: Página de documentación previa a las búsquedas.

En ninguna de estas secciones se usan librerías aparte de la de Bootstrap, ya que son estáticas y apenas tiene interacción con el usuario, ya que es simple texto para documentar o navegación entre páginas.

Análisis de Twitter

Esta sección se tratará de rellenar el formulario con el username que queremos analizar de Twitter, después de terminar dicho análisis se redirigirá a la parte de resultados del propio análisis con los distintos apartados, los cuales se irán describiendo uno por uno en orden de arriba a abajo de la página.



Figura 6.32: Páginas del formulario análisis de Twitter.

La primera parte será la opción para descargar en formato JSON toda la información generada, al clickar en el botón de *Descargar información generada*. Esta recapitulación de la información se genera desde el propio frontend, recopilando toda la información que se ha transmitido a él en los distintos apartados, un ejemplo de la estructura del JSON descargado sería la Figura 6.34.

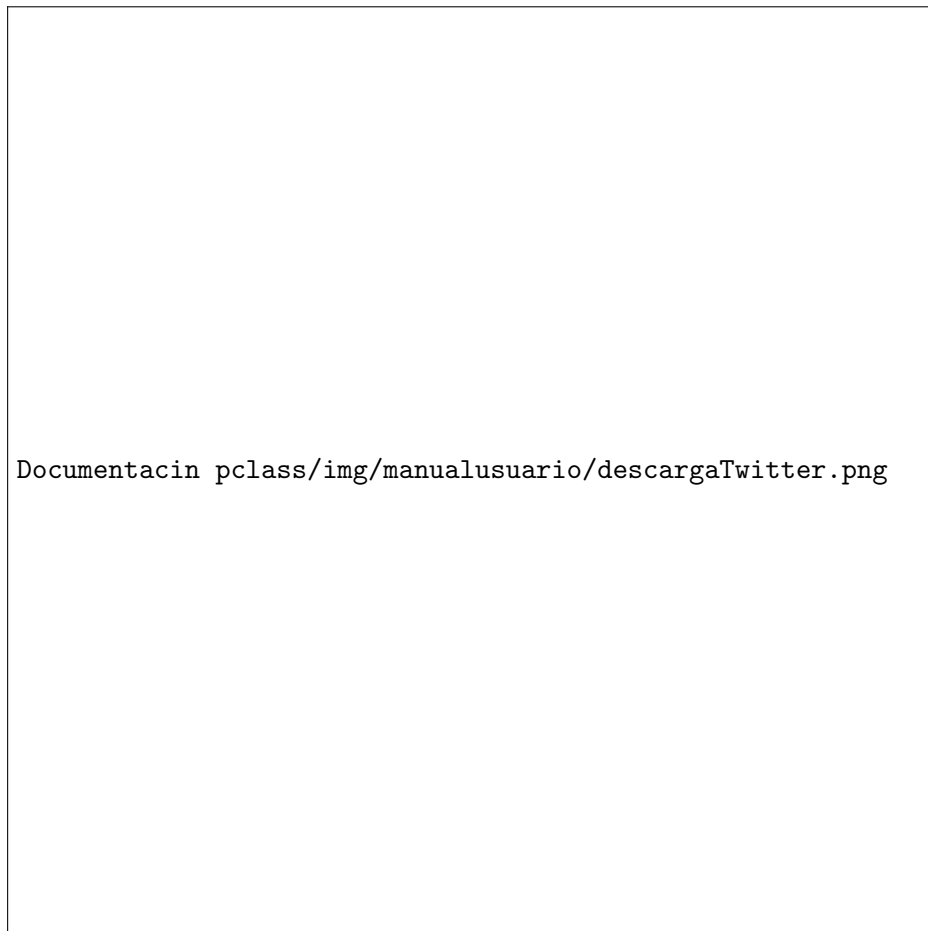


Figura 6.33: Opción descarga información generada Twitter.

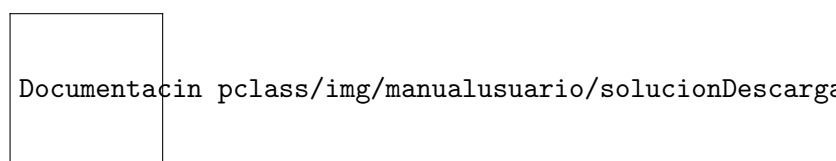


Figura 6.34: JSON generado.

El siguiente apartado se trata de la nube de palabras, para este apartado se uso la librería de wordcloud para la generación del mismo, aparte se crearon muchos filtros para eliminar palabras no sensibles o importantes a reflejar en él usando de apoyo la librería *nltk* para el procesamiento del lenguaje natural y la de *re* para las expresiones regulares. En este y los

siguientes apartados se mostrarán los resultados para la cuenta de Twitter del actual Presidente del Gobierno Pedro Sánchez Castejón.

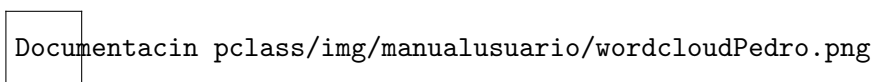


Figura 6.35: Wordcloud del Twitter de Pedro Sánchez.

A continuación se tratará el apartado de análisis de sentimientos, donde se podrá ver la tabla con los tweets que más puntuación tienen sobre cada sentimiento y una gráfica interactiva con el número total de tweets que están clasificados en los distintos sentimientos posibles. Para este apartado se usó varias librerías en el apartado del backend, como *networkx*, *torch*, *transformers*, *concurrent* para el procesamiento de clasificación en paralelo de los tweets, etc; para el front-end se usó la librería de *Chart.js* para la creación del gráfico circular con el porcentaje total de tweets clasificados en cada sentimiento.

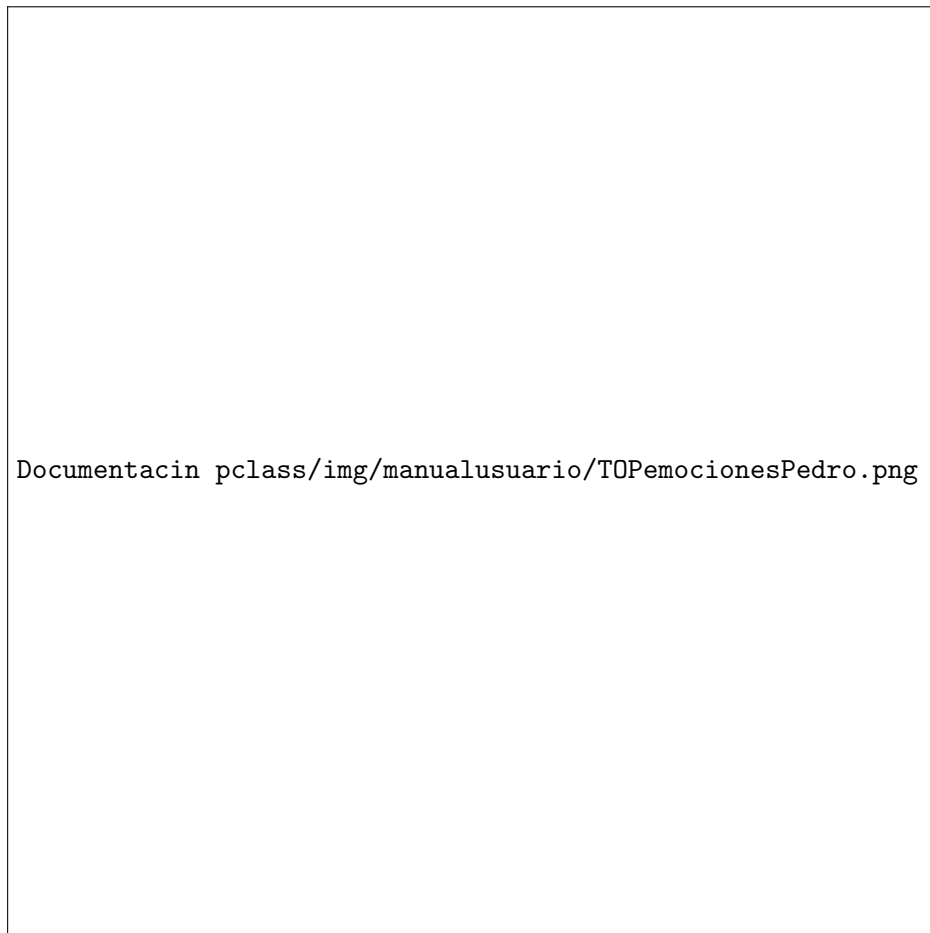


Figura 6.36: Resultado análisis de sentimientos del Twitter de Pedro Sánchez

El siguiente apartado se tratará del grafo TOP de interacciones, el cual podemos interactuar moviendo los nodos, ampliando el marco, clickando en cada nodo para hacer más detalles, etc gracias a la librería del front-end de NetworkJS.



Figura 6.37: Resultado grafo TOP interacciones del Twitter de Pedro Sánchez

El apartado siguiente se trata del apartado de Comunidades, al principio se puede ver un frame con el grafo en sí de comunidades, cada grafo conexo (comunidad) de un color y cada nodo con su nombre correspondiente, en él también se puede interactuar. Más abajo aparecerá una tabla con datos importantes sobre cada comunidad, y aún más abajo existirá un botón, el cual al clickar despliega como una leyenda explicando matemáticamente que significa cada valor de la tabla mostrado sobre los grafos basado en teoría de grafos.

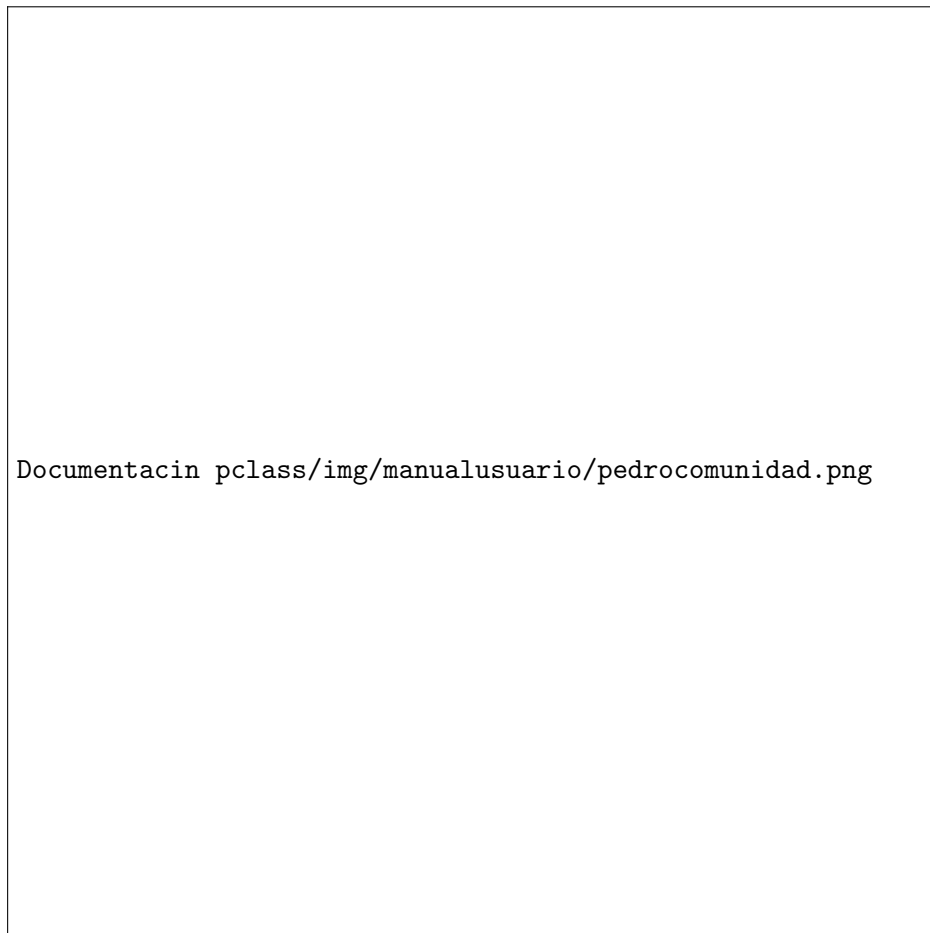


Figura 6.38: Resultado grafo de comunidades del Twitter de Pedro Sánchez



Figura 6.39: Resultado grafo de comunidades del Twitter de Pedro Sánchez

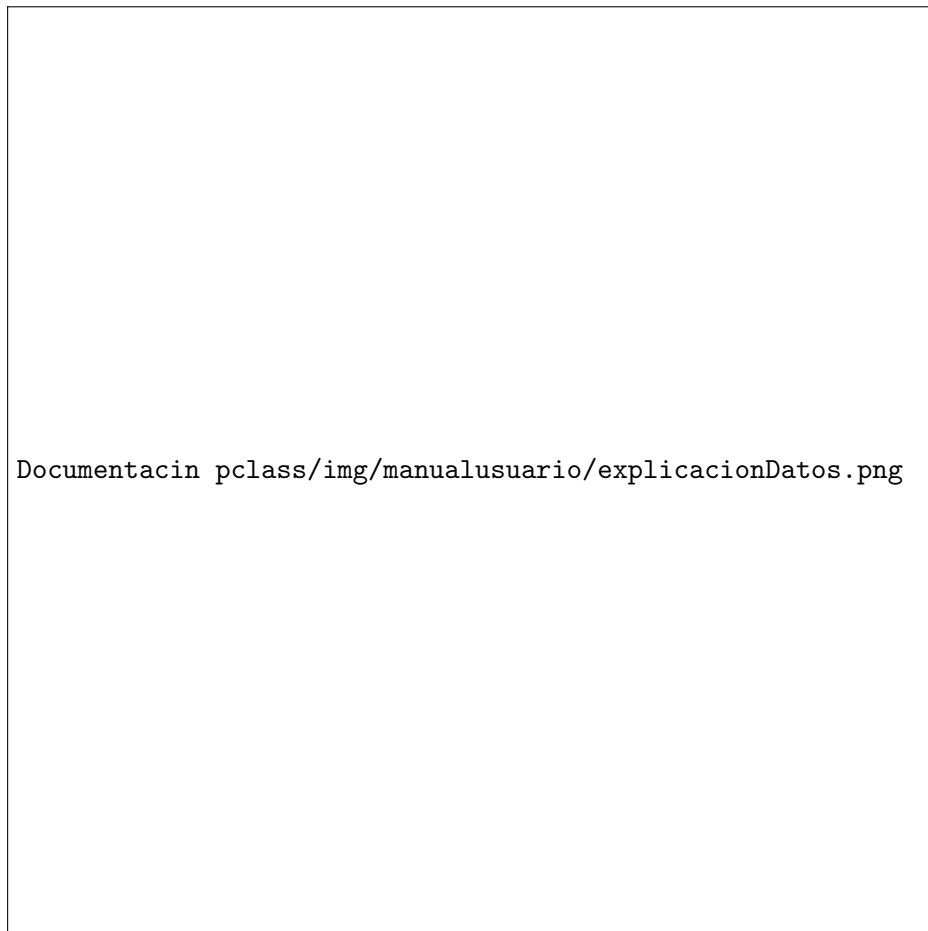


Figura 6.40: Leyenda explicativa sobre los valores de la tabla de comunidades

Finalmente el último apartado, donde se muestra una tabla con todas las localizaciones y fechas donde ha tuiteado el usuario, agrupadas por lugar y con una opción de filtro sobre las fechas, tanto en año, como mes, como día, como hora, minutos y segundos. Al final de este apartado existe un pequeño mapamundi que reúne todas las localizaciones visitadas por el usuario, para un vistazo más en general. En el front-end la librería usada para la generación de la tabla con sus filtros y clasificaciones es *Tabulator* y para el mapa interactivo se usó *Leaflet*.



Documentacin pclass/img/manualusuario/LocalizacionesResultado.png

Figura 6.41: Tabla de recopilación de todos las localizaciones y fechas.

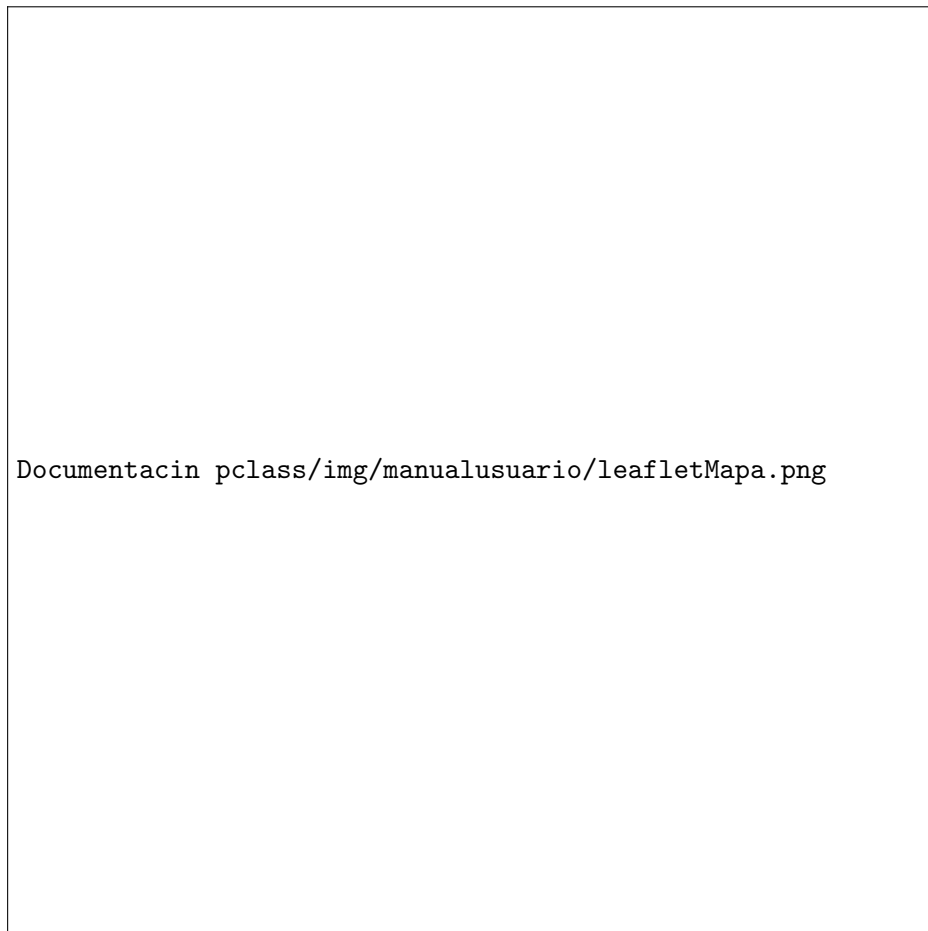


Figura 6.42: Mapamundi con todas las localizaciones.

Búsqueda de personas

Este apartado la longitud de la página dependerá de cuantos atributos hayamos escrito en el formulario de búsquedas de personas como se aprecia en la Figura 6.43 y de la cantidad de datos indexados que se hayan recopilado. Para una mejor muestra de la página en su totalidad se harán capturas seguidas de la página y finalmente se describirá algún detalle sobre esta página al final, para más detalles sobre cada apartado se puede apreciar en el apartado del manual de usuario referente a la búsqueda de personas 8.2.3.



Documentacin pclass/img/manualusuario/formularioPersonas.png

Figura 6.43: Formulario búsqueda de personas.



Documentacin pclass/img/manualusuario/descargaPersona.png

Figura 6.44: Descarga resultados de la búsqueda del objetivo.



Figura 6.45: Resultados búsqueda de datos del INE sobre el nombre y/o apellido(s).



Figura 6.46: Resultado búsqueda de datos sobre el nombre y/o apellido(s) y ciudad.



Figura 6.47: Gráficas sobre datos recopilados del nombre y/o apellido(s) y ciudad.



Figura 6.48: Resultado de la Darknet sobre el nombre y/o apellido(s).



Figura 6.49: Resultado búsqueda de datos sobre el nickname.

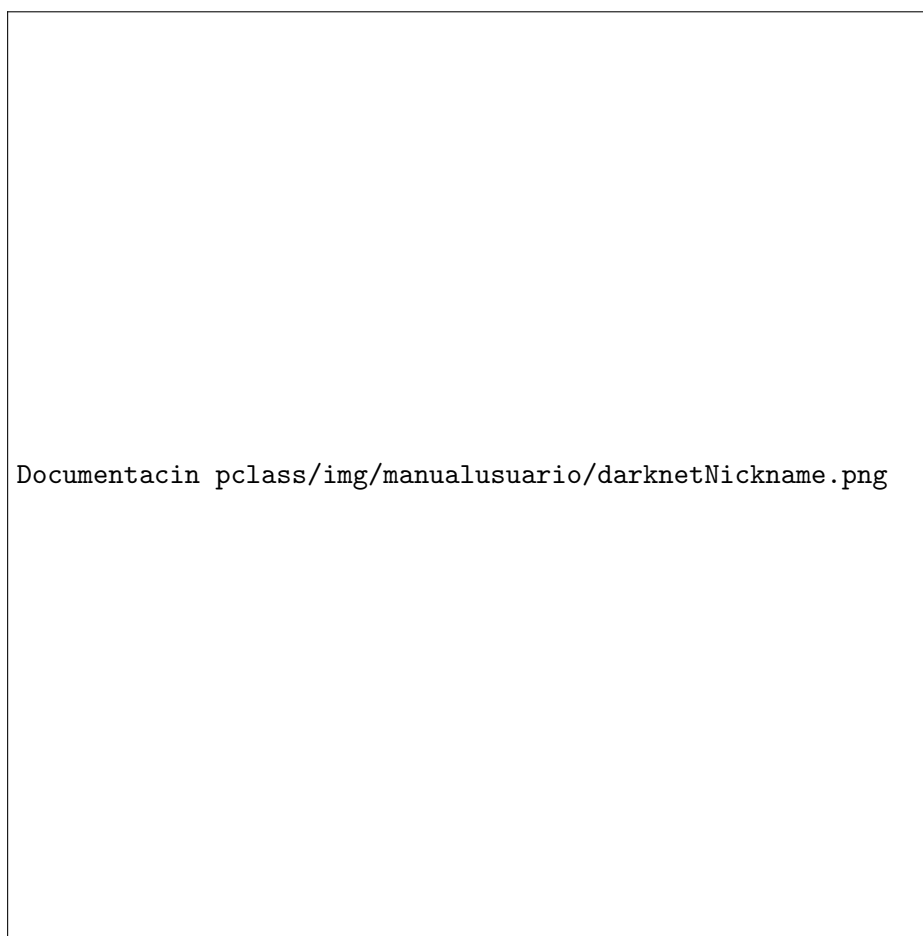


Figura 6.50: Resultado de la Darknet sobre el nickname.



Figura 6.51: Dashboard de resultados de IntelX sobre el correo.



Documentacin pclass/img/manualusuario/HIBPGrafica.png

Figura 6.52: Dashboard de resultados de HIBP sobre el correo.



Figura 6.53: Resultado búsqueda de datos sobre el correo en IntelX.



Figura 6.54: Resultado búsqueda de datos sobre el correo en HIBP.



Documentacin pclass/img/manualusuario/darknetEmail.png

Figura 6.55: Resultado de la Darknet sobre el email.



Figura 6.56: Dashboard de resultados de IntelX sobre el número de teléfono.



Figura 6.57: Dashboard de resultados de HIBP sobre el número de teléfono.



Documentacin pclass/img/manualusuario/resultadosphoneintelx.png

Figura 6.58: Resultado búsqueda de datos sobre el número de teléfono en IntelX.



Figura 6.59: Resultado búsqueda de datos sobre el número de teléfono en HIBP.



Figura 6.60: Resultado de la Darknet sobre el número de teléfono.

Como se puede apreciar las vistas están divididas por secciones, para así modularizarlas por atributo, en general el orden de cada apartado será primero unas gráficas o dashboards para un impacto más visual de la información, a continuación se verán divididos en cartas de Bootstrap cada resultado unitario que se ha encontrado y finalmente, si se puso como opción encontrar resultados en la Darknet existirá una sección al final de cada apartado donde se describa en forma de carta los resultados encontrados en la Darknet de dicho atributo.

6.5— Pruebas

En esta sección se van a detallar las pruebas que se han ido realizando cuando se han ido completando las distintas funcionalidades de OsintS-

pector y finalmente un apartado exponiendo los ataques más comunes que se deben de evitar, protegiendo al sistema de ellos.

6.5.1. Pruebas de aceptación

Se realizaron pruebas de aceptación a los casos de uso desarrollados en la Sección 5.1, estas se han realizado de forma manual, usando malintencionadamente las funcionalidades. Se procede a realizar una explicación de cada test realizado y de su resultado, así como las excepciones, es decir, lo que ocurre al realizar una acción no permitida.

Pruebas sobre el primer bloque asociado a la documentación previa a las funciones principales:

CU-1: Información previa a las búsquedas

Test < #001 >	
Descripción	Comprobar que se muestra la pantalla de información previa a las búsquedas.
Resultado	Se muestra la pantalla de información previa a las búsquedas en la aplicación, la cual permite documentar en que consisten dichas búsquedas y sus características.

Tabla 6.1: Prueba sobre información previa a las búsquedas.

CU-2: Información sobre el desarrollador

Test < #002 >	
Descripción	Comprobar que se muestra la pantalla de información sobre el desarrollador.
Resultado	Se muestra la pantalla de información sobre el desarrollador en la aplicación, la cual permite conocer a los desarrolladores y como contactar con ellos.

Tabla 6.2: Prueba sobre información sobre el desarrollador.

Pruebas sobre el segundo bloque asociado al apartado de análisis de Twitter, una de las funciones principales:

CU-3: Visualización de la nube de palabras

Test < #003 >	
Descripción	Comprobar que se pueda visualizar la nube de palabras generada.
Resultado	Cuando un usuario envía la petición de análisis de Twitter de cierto username se pueda ver en la información generada el apartado de la nube de palabras.
Excepciones	El sistema muestra un aviso de error cuando: <ul style="list-style-type: none">• Se envía la petición de análisis de una cuenta no válida para Twitter [13].

Tabla 6.3: Prueba sobre la nube de palabras.

CU-4: Visualización sobre el análisis de sentimientos

Test < #004 >	
Descripción	Comprobar que se pueda visualizar el apartado de análisis de sentimientos.
Resultado	Cuando un usuario envía la petición de análisis de Twitter de cierto username se pueda ver en la información generada el apartado de análisis de sentimientos.
Excepciones	El sistema muestra un aviso de error cuando: <ul style="list-style-type: none">• Se envía la petición de análisis de una cuenta no válida para Twitter [13].

Tabla 6.4: Prueba sobre el análisis de sentimientos.

CU-5: Visualización del grafo de interacciones

Test < #005 >	
Descripción	Comprobar que se pueda visualizar el apartado de grafo de interacciones.
Resultado	Cuando un usuario envía la petición de análisis de Twitter de cierto username se pueda ver en la información generada el apartado de grafo de interacciones.
Excepciones	El sistema muestra un aviso de error cuando: <ul style="list-style-type: none">• Se envía la petición de análisis de una cuenta no válida para Twitter [13].

Tabla 6.5: Prueba sobre el grafo de interacciones.

CU-6: Visualización del grafo de comunidades

Test < #006 >	
Descripción	Comprobar que se pueda visualizar el apartado de grafo de comunidades.
Resultado	Cuando un usuario envía la petición de análisis de Twitter de cierto username se pueda ver en la información generada el apartado de grafo de comunidades.
Excepciones	El sistema muestra un aviso de error cuando: <ul style="list-style-type: none">• Se envía la petición de análisis de una cuenta no válida para Twitter [13].

Tabla 6.6: Prueba sobre el grafo de comunidades.

CU-7: Visualización de la recopilación de localizaciones

Test < #007 >	
Descripción	Comprobar que se pueda visualizar el apartado de la recopilación de localizaciones.
Resultado	Cuando un usuario envía la petición de análisis de Twitter de cierto username se pueda ver en la información generada el apartado de recopilación de localizaciones.
Excepciones	El sistema muestra un aviso de error cuando: <ul style="list-style-type: none"> • Se envía la petición de análisis de una cuenta no válida para Twitter [13].

Tabla 6.7: Prueba sobre el grafo de la recopilación de localizaciones.

CU-8: Descarga de la información generada de Twitter

Test < #008 >	
Descripción	Comprobar que se pueda descargar toda la información generada, excepto la nube de palabras, del apartado de análisis de Twitter en un archivo con formato JSON.
Resultado	Cuando un usuario envía la petición de análisis de Twitter de cierto username se pueda ver en la página con la información generada un botón, donde al presionar en éste se pueda descargar toda la información generada en formato JSON, excepto la parte de nube de palabras.

Tabla 6.8: Prueba sobre la descarga de información generada de Twitter.

Pruebas sobre el tercer y último bloque asociado a la búsqueda de información de personas, una de las funciones principales:

CU-9: Visualización sobre el nombre y/o apellido(s) del INE

Test < #009 >	
Descripción	Comprobar que se pueda visualizar el apartado de datos recogidos del INE a partir del nombre y/o apellido(s).
Resultado	Cuando un usuario envía la petición de búsqueda de personas y añade el nombre y/o apellido(s) del objetivo se puede ver en la información generada el apartado de datos recopilados del INE.

Tabla 6.9: Prueba sobre los datos recogidos del INE a partir del nombre y/o apellido(s).

CU-10: Visualización sobre el N.A.C en motores de búsqueda

Test < #010 >	
Descripción	Comprobar que se pueda visualizar el apartado de datos recogidos sobre el nombre y/o apellido(s) y ciudad del objetivo en motores de búsqueda.
Resultado	Cuando un usuario envía la petición de búsqueda de personas y añade el nombre y/o apellido(s) y ciudad del objetivo se puede ver en la información generada el apartado de datos recopilados a partir de motores de búsqueda.
Excepciones	El sistema muestra un aviso de error cuando: <ul style="list-style-type: none">• Si el usuario solo rellena el apartado de ciudad.

Tabla 6.10: Prueba sobre los datos recogidos del nombre y/o apellido(s) y ciudad en motores de búsqueda.

CU-11: Visualización sobre el nickname en motores de búsqueda

Test < #011 >	
Descripción	Comprobar que se pueda visualizar el apartado de datos recogidos sobre el nickname del objetivo en motores de búsqueda.
Resultado	Cuando un usuario envía la petición de búsqueda de personas y añade el nickname del objetivo se puede ver en la información generada el apartado de datos recopilados a partir de motores de búsqueda.

Tabla 6.11: Prueba sobre los datos recogidos del nickname en motores de búsqueda.

CU-12: Visualización sobre el e-mail en motores de búsqueda

Test < #012 >	
Descripción	Comprobar que se pueda visualizar el apartado de datos recogidos sobre el correo electrónico del objetivo en motores de búsqueda.
Resultado	Cuando un usuario envía la petición de búsqueda de personas y añade el correo electrónico del objetivo se puede ver en la información generada el apartado de datos recopilados a partir de motores de búsqueda.

Tabla 6.12: Prueba sobre los datos recogidos del correo electrónico en motores de búsqueda.

CU-13: Visualización sobre el teléfono en motores de búsqueda

Test < #013 >	
Descripción	Comprobar que se pueda visualizar el apartado de datos recogidos sobre el número teléfono del objetivo en motores de búsqueda.
Resultado	Cuando un usuario envía la petición de búsqueda de personas y añade el número teléfono del objetivo se puede ver en la información generada el apartado de datos recopilados a partir de motores de búsqueda.

Tabla 6.13: Prueba sobre los datos recogidos del número teléfono en motores de búsqueda.

CU-14: Visualización sobre datos indexados en la Darknet

Test < #014 >	
Descripción	Comprobar que se pueda visualizar el apartado de datos indexados en la Darknet sobre los datos rellenados del objetivo (nombre y/o apellido(s), nickname, correo electrónico y/o número de teléfono).
Resultado	Cuando un usuario envía la petición de búsqueda de personas y añade algún atributo del objetivo y a su vez selecciona la opción de búsqueda en la Darknet de los atributos rellenados, a continuación se pueda visualizar la información recopilada en la Darknet de dichos atributos.
Excepciones	El sistema muestra un aviso de error cuando: <ul style="list-style-type: none"> • Si el usuario solo rellena el apartado de ciudad. • Si el usuario solo selecciona si desea o no buscar información de la Darknet sobre los atributos rellenados.

Tabla 6.14: Prueba sobre los datos indexados en la Darknet sobre los atributos rellenados.

CU-15: Descarga de la información resultante de la persona

Test < #015 >	
Descripción	Comprobar que se pueda descargar la información resultante generada por el sistema sobre la búsqueda de una persona objetivo.
Resultado	El usuario al visualizar la página con la información recopilada de la persona pueda clickar en un botón para descargarse en formato JSON dicha información generada.

Tabla 6.15: Prueba sobre la descarga de la información resultante de la persona.

6.5.2. Intrusión

En la documentación de Flask se especifica lo siguiente sobre los ataques de intrusión más comunes [32] a evitar junto con algunos consejos

para abordar dichos problemas:

- **Cross-Site Scripting (XSS)** es la inyección de código HTML y JavaScript en un sitio web, Flask utiliza Jinja2 para escapar automáticamente los valores y prevenir problemas de XSS en las plantillas. Sin embargo, se deben tener precauciones adicionales al generar HTML sin Jinja2, al manipular datos enviados por usuarios, al enviar HTML desde archivos cargados y al tratar atributos no citados. Es importante citar los atributos correctamente para evitar la inyección de código malicioso. Además, Jinja2 no protege contra la ejecución de JavaScript en el atributo href de la etiqueta `.a`, por lo que se recomienda establecer una política de seguridad de contenido (Content Security Policy, CSP) para mitigar esta vulnerabilidad.
- **Cross-Site Request Forgery (CSRF)** es un problema grave en el que terceros pueden usar la información almacenada en las cookies para enviar solicitudes falsas en nombre del usuario real. Esto puede hacer que los usuarios hagan cosas no deseadas sin su conocimiento. Por ejemplo, un atacante puede engañar a los usuarios para que carguen una página que envía una solicitud POST para eliminar su perfil. Para prevenir esto, se recomienda utilizar tokens únicos en cada solicitud que modifica el contenido del servidor. Flask no proporciona un marco de validación de formularios, por lo que la implementación de esta protección debe hacerse manualmente. Para prevenir esto, se recomienda utilizar tokens únicos en cada solicitud que modifica el contenido del servidor. Flask no proporciona un marco de validación de formularios, por lo que la implementación de esta protección debe hacerse manualmente.
- **Seguridad en las cabeceras**, hay que tener en cuenta que en las cabeceras existen distintos encabezados que ayudan a evitar posibles ataques. Entre ellos, se destacan el HTTP Strict Transport Security (HSTS) que garantiza la comunicación segura a través de HTTPS, el Content Security Policy (CSP) que controla qué recursos se pueden cargar en la página, el X-Content-Type-Options que previene ataques de tipo cross-site scripting (XSS), el X-Frame-Options que evita el clickjacking, y las opciones de Set-Cookie que mejoran la seguridad de las cookies. Además, el HTTP Public Key Pinning (HPKP) autentica el servidor utilizando una clave de certificado específica. Estos encabezados son fundamentales para fortalecer la seguridad en aplicaciones web.

CAPÍTULO 7

Conclusiones

Este capítulo de conclusiones comenzará con la sección 7.1, donde se procederá a estudiar sobre los aciertos y fallos del desarrollo, para tener una retrospectiva general y crear una base de conocimiento de cara a próximos proyectos. La sección 7.2, lecciones aprendidas, describirá los conocimientos adquiridos con el proyecto. Se concluirá este capítulo con la sección 7.3, donde se expondrán las posibles mejoras para el sistema.

7.1— Retrospectiva

7.1.1. Aspectos a repetir

A continuación, se expondrá algunos de los puntos más importantes a repetir que se han logrado en el desarrollo de OsintSpector de cara a futuros proyectos:

- Planificación correcta. El coste de tiempo real de desarrollo ha sido solo de un margen menor al 5 % respecto a la estimación a priori que se tenía, creando un sistema totalmente funcional.
- Elicitación de requisitos correcta y un esquema lógico de la arquitectura adecuado a nuestras necesidades.
- Buena elección sobre las tecnologías usadas en cada apartado de la arquitectura.
- Comunicación con el tutor.

7.1.2. Aspectos a mejorar o evitar

Uno de los problemas más importantes de cara al desarrollo de esta aplicación ha sido la parte de extracción de información en servicios exter-

nos, en concreto aquellos en los que se usaba scraping, ya que, la mayoría de servicios no aceptan este tipo de extracción de datos en sus términos de uso, por ello se ha tenido que ocultar de varias maneras la acción de scrapear y en otros casos, como en el de HIBP, se ha tenido que reestructurar desde cero y pagando la API, debido al problema que se tuvo. Quizás la solución perfecta hubiera sido contratar servicios externos que nos proporcionen estos datos sin violar sus términos, pero este aspecto fue elegido a priori así debido a que el desarrollador quería mejorar sus habilidades en esa materia.

Otro punto también a mejorar sería en la parte de Twitter, ya que desde la compra de Twitter por parte de Elon Musk ha cambiado muchísimo todos los aspectos de la plataforma, tanto las condiciones de las APIs, como los tipos de restricción de búsquedas, cerrar distintas funcionalidades que antes si se tenía de cara a extraer datos, etc. Quizás hubiera sido mejor haber elegido otra red social como Instagram, Facebook, LinkedIn, etc; pero es algo que en los primeros meses no hubo tanto movimiento de cambios en Twitter y ya se desarrollaron las bases del proyecto en él.

7.2— Lecciones aprendidas

Algunas lecciones aprendidas de cara a tener experiencia para futuros proyectos ha sido:

- Una buena planificación es la base de todo, incluso de ésta depende el éxito o fallo e incluso abandono del proyecto.
- Tener claro a su vez los objetivos principales, intentando cambiarlos lo menos posible desde el principio, para así tener unos términos donde apoyarse y no improvisar en el proceso de desarrollo.
- La comunicación constante con el cliente (el tutor en este caso) es fundamental para establecer las necesidades del usuario a medida que se va desarrollando las distintas partes del proyecto.
- Destacar también que gracias a OsintSpector se ha aprendido mucho de varios temas, por ejemplo en términos de programación se ha aprendido mucho sobre Python y módulos importantes de él como *Pandas*, *BeautifulSoup* o *Playwright*; a su vez también se ha aprendido bastante sobre distintas librerías de Javascript como *Tabulator*, *Apexcharts* o *Leaflet*. En conocimientos de frameworks igualmente se ha aprendido bastante, ya que desde un principio el desarrollador no tenía conocimiento alguno sobre Flask o algún framework parecido. Finalmente también se ha aprendido mucho sobre el proceso de planificación, diseño, seguimiento y cierre de un proyecto consistente y serio.

7.3– Posibles mejoras del sistema

El proyecto, aunque sea funcional, no tiene realizado ningún despliegue, hubiera sido interesante desplegar dicha web y tener una parte de integración e implementación continua utilizando alguna tecnología como Jenkins.

En el apartado de búsqueda de personas hubiera estado interesante utilizar distintos motores de búsqueda, como DuckDuckGo, Bing, Yahoo, etc; y no solamente el de Google, aunque sea el más potente, para así tener una diversidad de resultados; por desgracia se tuvo que recortar esta parte por el tiempo extra de dedicación que se hubiera tenido que realizar.

CAPÍTULO 8

Manuales

8.1– Manual de instalación y despliegue

El siguiente manual expone los pasos a realizar para instalar y ejecutar el sistema de Osintpector en Windows 10.

8.1.1. Tecnologías a descargar previa a la instalación

Para este proyecto se necesita descargar el lenguaje de Python, a ser posible la versión 3.10 o superior, aunque realmente funciona con versiones anteriores, pero para asegurar su correcto funcionamiento se recomienda la versión 3.10.

Para la creación del servicio interno de análisis de sentimientos, si el usuario posee una gráfica de marca NVIDIA o una gráfica que soporte CUDA [14] es más que recomendable la instalación de ésta, ya que, el servicio de análisis de sentimientos está configurado para usar antes la GPU que la CPU por tema de rendimiento y tiempos de finalización.

Para instalar CUDA en la máquina, se debe seguir los siguientes pasos:

1. Descargar la versión adecuada de CUDA desde la página oficial de NVIDIA: <https://developer.nvidia.com/cuda-downloads>
2. Ejecutar el instalador descargado.
3. Seguir las instrucciones del instalador para seleccionar las opciones de instalación que se desea.
4. Asegurarse de agregar la ruta a la carpeta "bin" de CUDA a la variable de entorno PATH del sistema personal.

5. Verificar que la instalación haya sido exitosa ejecutando `nvcc -version` en el terminal. Si se muestra la versión de CUDA, entonces la instalación fue exitosa.

Hay que tener en cuenta que la instalación de CUDA puede ser un proceso complicado y puede haber problemas de compatibilidad con el hardware de la máquina. Si existen otros problemas para instalar CUDA, es recomendable buscar ayuda en los foros de NVIDIA o en la comunidad de programación de Python.

8.1.2. Instalación

El primer paso es clonar el repositorio de OsintSpector, el cual se aloja en Github. A continuación crearemos un entorno virtual y lo activaremos para finalmente instalar las dependencias necesarias para el funcionamiento del sistema. El proceso sería:

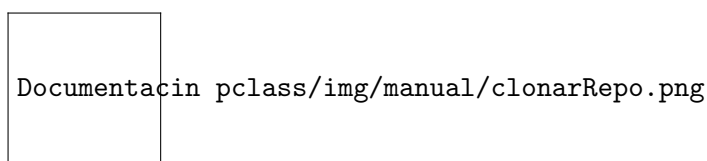


Figura 8.1: Clonación del repositorio.

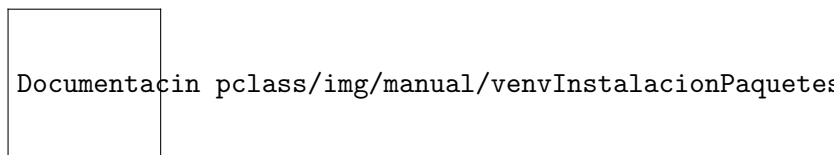


Figura 8.2: Creación entorno virtual y descarga de paquetes.

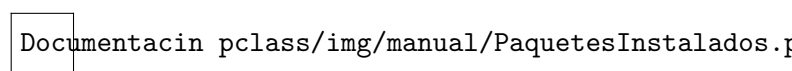


Figura 8.3: Todos los paquetes instalados finalmente.

Luego tendríamos que crear una carpeta dentro de OsintSpector llamada `.env` para guardar todas las variables de entorno que necesitamos, en el caso del proyecto serían un total de seis claves distintas, cinco de ellas son para conectarnos con las distintas APIs necesarias y la otra es la clave de autenticación principal del proyecto de Flask.

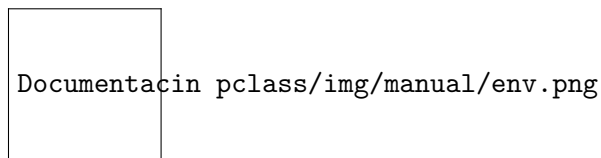


Figura 8.4: Variables de entorno.

Una vez ejecutados estos últimos comandos y creada la carpeta de variables de entorno junto a las claves, el servidor quedará preparado para su uso, solo tendremos que ejecutar el módulo principal del proyecto que es *App.py*. De esta forma, en el navegador se podrá abrir una ventana a la dirección, la cual normalmente es `http://127.0.0.1:5000` a menos que exista otro servicio ocupando dicha IP. Una vez entrando en la dirección correcta se encontrará la página principal, donde ya podremos navegar sin ningún problema.

8.2– Manual de usuario

Se finalizará el capítulo con una visión general de la herramienta, dotando a cualquier usuario de la información necesaria para su uso. Es necesario haber realizado los pasos descritos en la Sección 8.1.

El sistema se abrirá automáticamente en la dirección localhost:5000 mostrando la siguiente página principal, correspondiente a la Figura 8.5



Figura 8.5: Página principal.

8.2.1. Documentación previa a las funciones principales

Antes de realizar cualquier búsqueda es recomendable dirigirse al apartado de más información que aparece arriba del todo en grande junto a la imagen, o en la navbar superior a la derecha también aparece el mismo

sitio web, solo que se llama *De qué se trata este proyecto*. Una vez ubicados en esta página se explica un poco en general sobre el proyecto y sus tipos de búsquedas, como se puede apreciar en la Figura 8.6.



Figura 8.6: Página de documentación previa a las búsquedas.

También en la parte de la barra de navegación superior existe otro apartado llamado *Sobre nosotros*, el cual explica un poco quienes fueron los desarrolladores del proyecto y una breve explicación de su motivación, como se puede apreciar en la Figura 8.7.

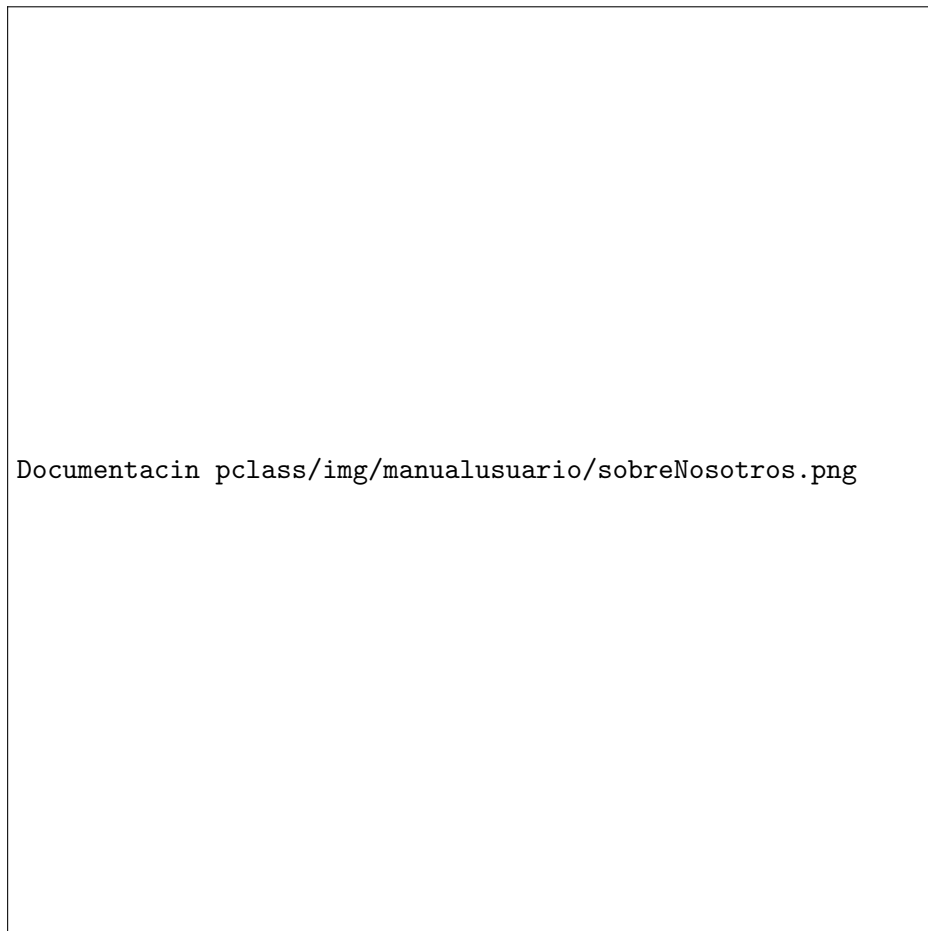


Figura 8.7: Página de sobre nosotros.

Una vez visitados estos sitios ya es hora de usar sus funciones principales, se podrá entrar en estos sitios desde la página principal en la parte de abajo donde existen dos cartitas distintas, una para entrar en el formulario de análisis de Twitter y la otra para la búsqueda de personas.



Figura 8.8: Navegación a las funciones principales.

8.2.2. Análisis de Twitter

Empezaremos con el apartado de análisis de Twitter, al clickar en dicho botón se navegará hasta el formulario de análisis de la cuenta de Twitter del objetivo que querramos buscar, es tan simple como rellenar el atributo de *username* y darle al botón de enter del teclado o al propio botón de analizar que se encuentra en azulito abajo del formulario, como se puede comprobar en la Figura 8.9.

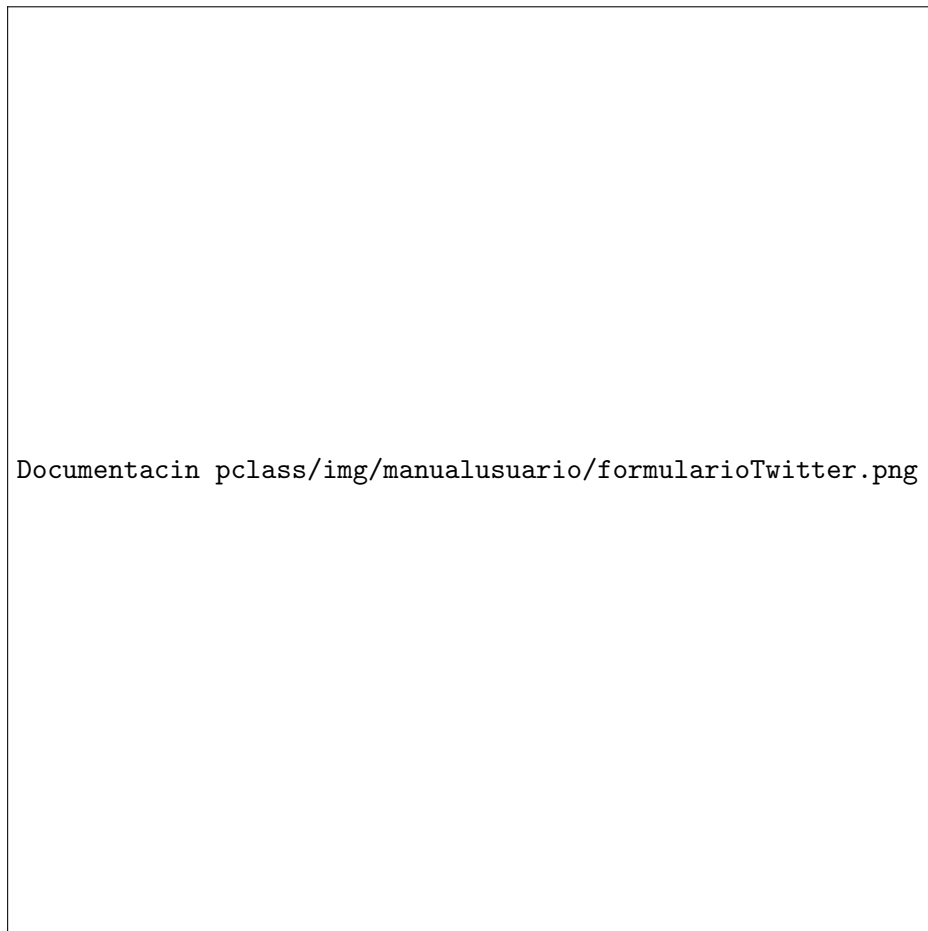


Figura 8.9: Página del formulario análisis de Twitter.

Una vez hayamos añadido un username y se haya cargado la página con los análisis hechos de cierta cuenta, veremos apartado por apartado cada uno de los resultados. En este manual se ha creado un ejemplo analizando la cuenta del tutor Ángel.

En la Figura 8.10 se puede ver la opción para descargar en formato JSON toda la información generada, al clickar en el botón de *Descargar información generada*.



Figura 8.10: Opción descarga información generada Twitter.

El siguiente apartado se tratará de la generación del wordcloud a través de los tweets recopilados de su cuenta, la imagen formará la silueta del logo de Twitter para que sea más vistosa, como se puede comprobar en la Figura 8.11.

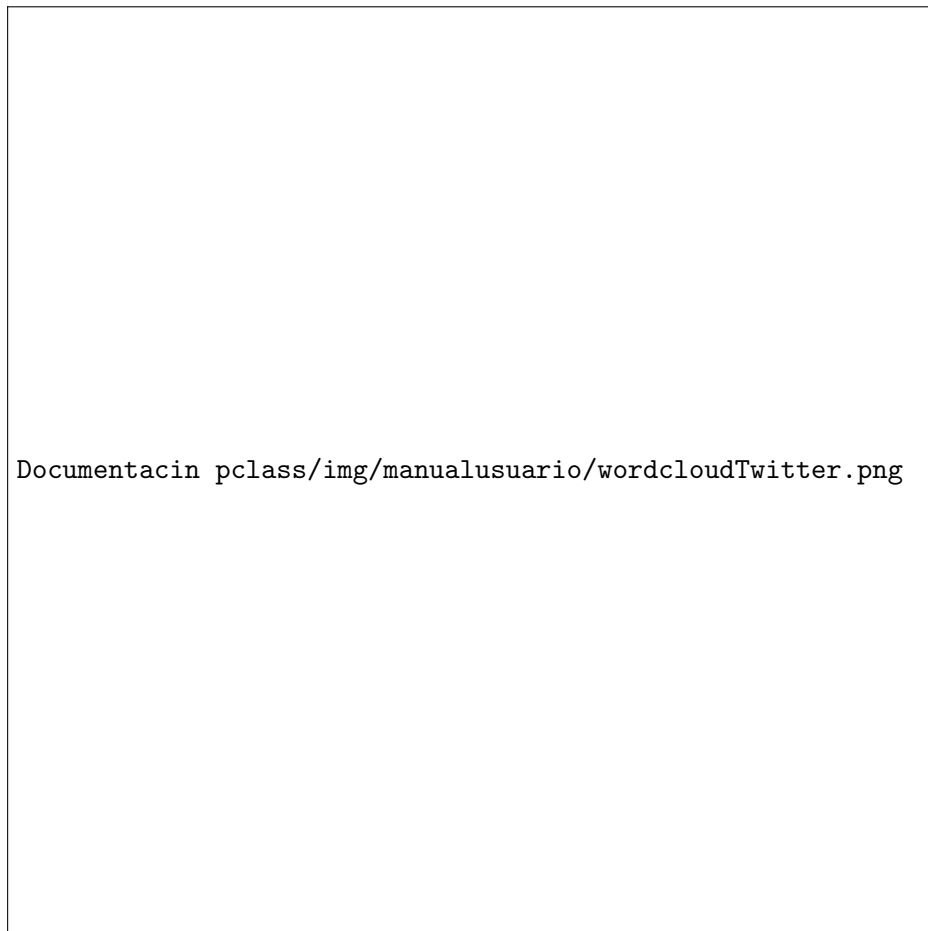


Figura 8.11: Wordcloud de Twitter.

A continuación se tratará el apartado de análisis de sentimientos, donde se podrá ver la tabla con los tweets que más puntuación tienen sobre cada sentimiento y una gráfica interactiva con el número total de tweets que están clasificados en los distintos sentimientos posibles, como se puede ver en la Figura 8.12.



Figura 8.12: Resultado análisis de sentimientos de Twitter.

El siguiente apartado se tratará del grafo TOP de interacciones, el cual podemos interactuar moviendo los nodos, ampliando el marco, clickando en cada nodo para hacer más detalles, etc. Como se puede apreciar en la Figura 8.13.



Figura 8.13: Resultado grafo TOP de interacciones.

El apartado siguiente se trata del apartado de Comunidades, al principio se puede ver un frame con el grafo en sí de comunidades, cada grafo conexo (comunidad) de un color y cada nodo con su nombre correspondiente, en él también se puede interactuar. Más abajo aparecerá una tabla con datos importantes sobre cada comunidad, y aún más abajo existirá un botón, el cual al clickar despliega como una leyenda explicando matemáticamente que significa cada valor de la tabla mostrado sobre los grafos basado en teoría de grafos, como se puede ver en las Figuras ??, ?? y ??.



Documentacin pclass/img/manualusuario/comunidadesResultadoGrafo.png

Figura 8.14: Resultado grafo de comunidades.



Figura 8.15: Resultado tabla de valores de las comunidades.



Figura 8.16: Leyenda explicativa sobre los valores de la tabla de comunidades.

Finalmente el último apartado, donde se muestra una tabla con todas las localizaciones y fechas donde ha tuiteado el usuario, agrupadas por lugar y con una opción de filtro sobre las fechas, tanto en año, como mes, como día, como hora, minutos y segundos. Al final de este apartado existe un pequeño mapamundi que reúne todas las localizaciones visitadas por el usuario, para un vistazo más en general.



Documentacin pclass/img/manualusuario/LocalizacionesResultado.png

Figura 8.17: Tabla de recopilación de todos las localizaciones y fechas.



Figura 8.18: Mapamundi con todas las localizaciones.

8.2.3. Búsqueda de personas

En esta última sección se explicará el apartado de búsqueda de personas, al igual que pasaba con el de análisis de Twitter se podrá entrar a dicho formulario desde la página principal del proyecto, clickando en el botón *Búsqueda de Persona*, como se puede apreciar en la Figura 8.8.

Esta vez nos llevará al formulario de búsqueda de personas, se rellenará dicho formulario con toda la información posible del objetivo, en este manual daremos como ejemplo datos míos propios, el formulario en sí sería como el de la Figura 8.19.



Figura 8.19: Formulario búsqueda de personas.

De nuevo al finalizar la búsqueda y llevarnos a la página con los resultados lo primero que se puede apreciar es la opción de descarga en formato JSON de los resultados generados, como se puede apreciar en la Figura 8.20.

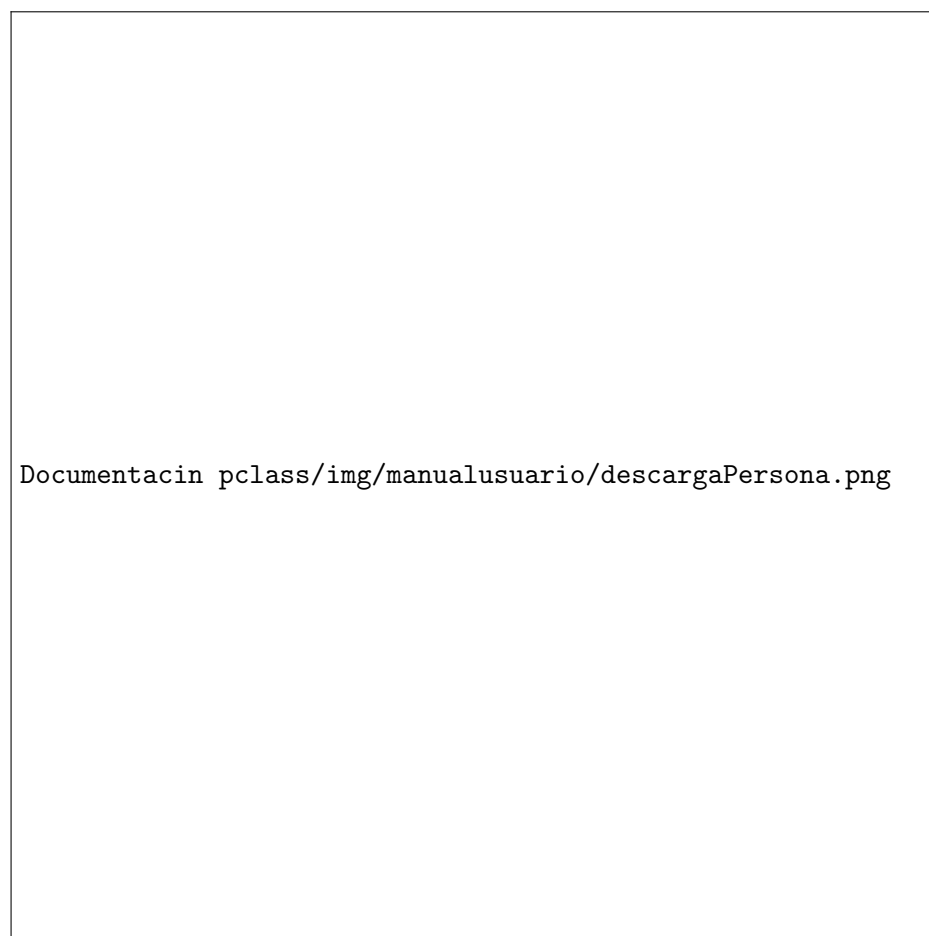


Figura 8.20: Descarga resultados de la búsqueda del objetivo.

En el apartado de nombre y/o apellido(s) y ciudad se muestra primero los datos generados por el INE, como se puede apreciar en la Figura 8.21. A continuación se mostrarán los datos recopilados de Internet, primero un wordcloud con la recopilación de palabras más comunes encontradas en cada resultado y una gráfica con los tipo de dominios donde se ha indexado nuestro nombre como se aprecia en la Figura 8.22, finalmente abajo del todo se muestra un apartado que muestra uno por uno los resultados encontrados en distintas páginas donde estuvieran nuestro nombre y/o apellidos, para mostrar más solo habría que clickar en el botón azul, como se aprecio en la Figura 8.23. Finalmente, si hemos añadido la opción de búsqueda en la Darknet aparecerá una última sección con dichos resultados recopilados más datos sobre cada indexación donde apareció nuestro nombre y/o apellido(s), como se aprecia en la Figura 8.24.



Figura 8.21: Resultados búsqueda de datos del INE sobre el nombre y/o apellido(s).



Figura 8.22: Resultado búsqueda de datos sobre el nombre y/o apellido(s) y ciudad.



Figura 8.23: Gráficas sobre datos recopilados del nombre y/o apellido(s) y ciudad.



Figura 8.24: Resultado de la Darknet sobre el nombre y/o apellido(s).

La siguiente sección es la de búsqueda del nickname sobre distintos motores de búsquedas, cada uno asociado a una categoría. Al principio se puede apreciar una gráfica interactiva que describe el tipo de categorías y su frecuencia al encontrar resultados sobre el nickname del objetivo. Abajo de esta gráfica encontraremos resultado por resultado de las cuentas creadas en distintos motores de búsquedas con su categoría y link para entrar en dicha plataforma y ver el perfil del nickname asociado a ella, como se puede apreciar en la Figura 8.25. Finalmente, si hemos añadido la opción de búsqueda en la Darknet aparecerá una última sección con dichos resultados recopilados más datos sobre cada indexación donde apareció nuestro nickname, como se aprecia en la Figura 8.26.



Figura 8.25: Resultado búsqueda de datos sobre el nickname.

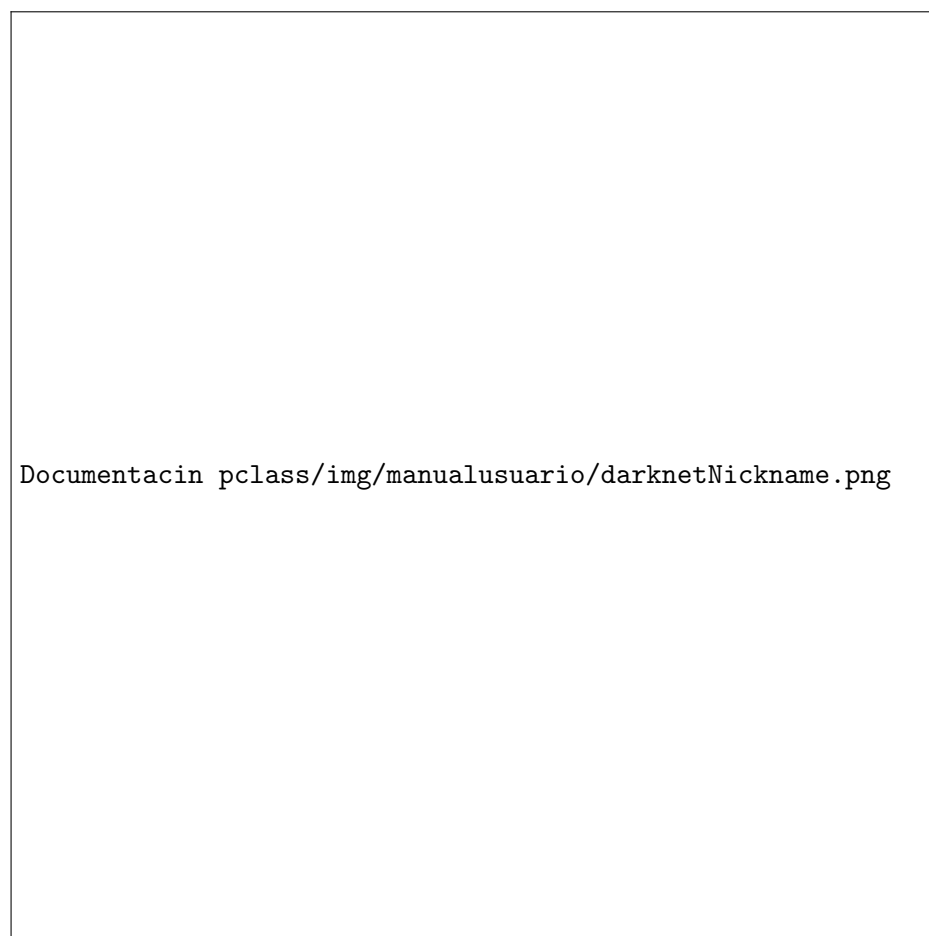


Figura 8.26: Resultado de la Darknet sobre el nickname.

A continuación está la sección de los resultados encontrados sobre el correo electrónico. Estará dividida principalmente en dos partes, la primera que trata sobre los datos encontrados en la base de datos de IntelX y la otra parte de los datos encontrados en HIBP. Como se puede apreciar, para el caso de IntelX en la Figura 8.27 y el caso de HIBP en la Figura 8.28 ambas se tratan de dashboards con gráficas interactivables y abajo de ellas se encuentra cada resultado registrado donde esté indexado el correo en sus bases de datos, como se ve en la Figura 8.29 para el caso de IntelX y la Figura 8.30 para el caso de HIBP.



Figura 8.27: Dashboard de resultados de IntelX sobre el correo.



Figura 8.28: Dashboard de resultados de HIBP sobre el correo.



Figura 8.29: Resultado búsqueda de datos sobre el correo en IntelX.



Figura 8.30: Resultado búsqueda de datos sobre el correo en HIBP.

Finalmente, si hemos añadido la opción de búsqueda en la Darknet aparecerá una última sección con dichos resultados recopilados más datos sobre cada indexación donde apareció nuestro correo, como se aprecia en la Figura 8.31.



Figura 8.31: Resultado de la Darknet sobre el email.

Como última sección está la de los resultados encontrados sobre el número de teléfono. Estará dividida principalmente en dos partes, la primera que trata sobre los datos encontrados en la base de datos de IntelX y la otra parte de los datos encontrados en HIBP. Como se puede apreciar, para el caso de IntelX en la Figura 8.32 y el caso de HIBP en la Figura 8.33 ambas se tratan de dashboards con gráficas interactivas y abajo de ellas se encuentra cada resultado registrado donde esté indexado el teléfono en sus bases de datos, como se ve en la Figura 8.34 para el caso de IntelX y la Figura 8.35 para el caso de HIBP.



Figura 8.32: Dashboard de resultados de IntelX sobre el número de teléfono.



Figura 8.33: Dashboard de resultados de HIBP sobre el número de teléfono.



Documentacin pclass/img/manualusuario/resultadosphoneintelx.png

Figura 8.34: Resultado búsqueda de datos sobre el número de teléfono en IntelX.

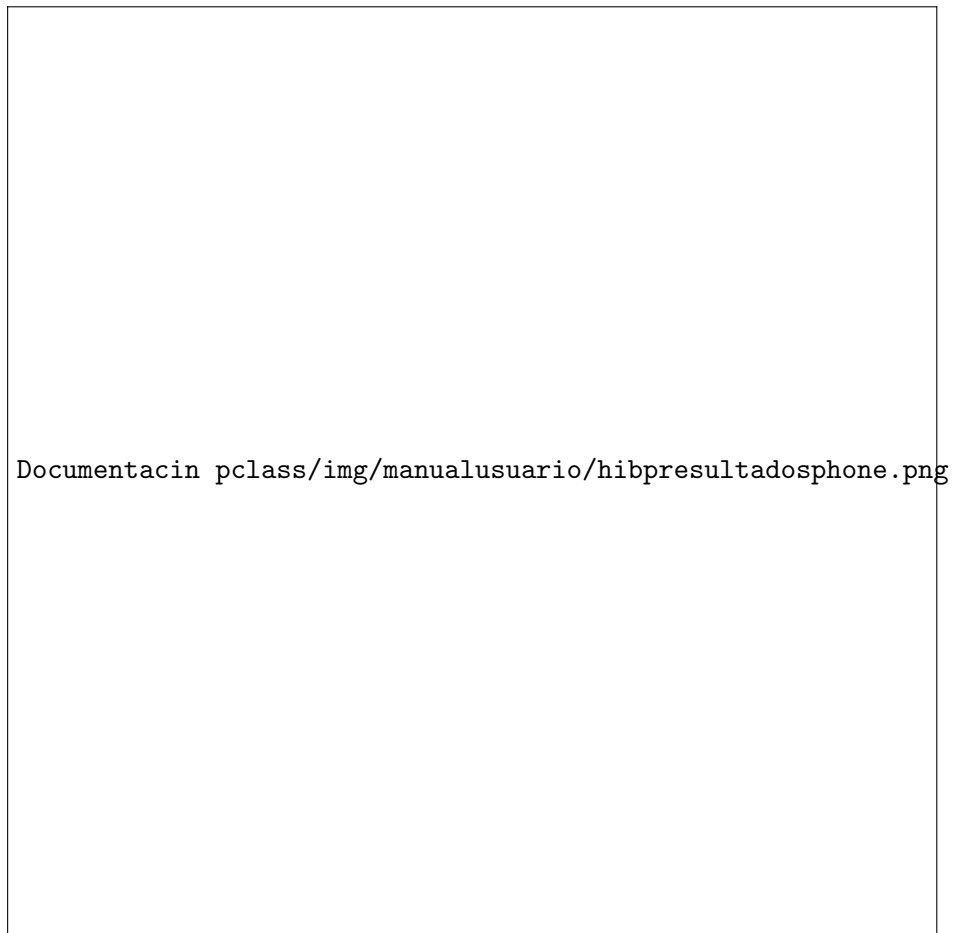


Figura 8.35: Resultado búsqueda de datos sobre el número de teléfono en HIBP.

Finalmente, si hemos añadido la opción de búsqueda en la Darknet aparecerá una última sección con dichos resultados recopilados más datos sobre cada indexación donde apareció nuestro correo, como se aprecia en la Figura 8.36.



Figura 8.36: Resultado de la Darknet sobre el número de teléfono.

Bibliografía

- [1] Cambios en la política de la api de twitter. <https://twitter.com/TwitterDev/status/1641222782594990080>.
- [2] Eliminado el tipo de cuenta académica en twitter. <https://twitter.com/TwitterDev/status/1641222788911624192>.
- [3] Error en la recopilación de tweets usando snsrape. <https://github.com/JustAnotherArchivist/snsrape/issues/846>.
- [4] Librería snsrape. <https://github.com/JustAnotherArchivist/snsrape>.
- [5] Resultados de google search api y google.com no son idénticos. <https://code.google.com/archive/p/google-ajax-apis/issues/43>.
- [6] Documentación de flask. <https://flask.palletsprojects.com/en/2.3.x/>, 2023.
- [7] Apexcharts. Dashboards en apexcharts. <https://apexcharts.com/javascript-chart-demos/dashboards/>, 2023.
- [8] Federico Bianchi, Debora Nozza, and Dirk Hovy. XLM-EMO: Multilingual Emotion Prediction in Social Media Text. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, 2022.
- [9] Pedro Baños Carlos Seisdodos, Vicente Aguilera. *Open Source Intelligence (OSINT): Investigar personas e Identidades en Internet*. Oxword, Móstoles, 1 edition, 2020.
- [10] Ciberseguridad.com. Web scraping. <https://ciberseguridad.com/guias/recursos/web-scraping/>.
- [11] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, Dec 2004.

- [12] ClickUp. Clickup, One app to replace them all. <https://clickup.com/>, 2022.
- [13] X Corp. Help with username registration. <https://help.twitter.com/en/managing-your-account/twitter-username-rules#:~:text=Your%20username%20cannot%20be%20longer,of%20underscores%2C%20as%20noted%20above.>, 2023.
- [14] NVIDIA Corporation. Nvidia cuda. <https://docs.nvidia.com/cuda/doc/index.html>.
- [15] Marco de desarrollo de la Junta de Andalucía. Guía para la redacción de casos de uso. <https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/416>, 2016.
- [16] Gobierno de España. Ley orgánica 1/1982, de 5 de mayo, de protección civil del derecho al honor, a la intimidad personal y familiar y a la propia imagen. *BOE-A-1982-11196.*, 6, 1982, última modificación 2010.
- [17] Gobierno de España. Ley orgánica 3/2018, de 5 de diciembre, de protección de datos personales y garantía de los derechos digitales. *BOE-A-2018-16673.*, 70, 2018.
- [18] Instituto Nacional de Estadística. Encuesta sobre equipamiento y uso de tecnologías de información y comunicación en los hogares 2021 (tic_h 2021). *TIC_H 2021.*, 20, 2021.
- [19] Instituto Nacional de Estadística. Encuesta sobre equipamiento y uso de tecnologías de información y comunicación en los hogares 2021 (tic_h 2021). informe metodológico., 2021.
- [20] Instituto Nacional de Estadística. Demografía y población. padrón, población por municipios. apellidos y nombres más frecuentes, últimos datos. https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736177009&menu=ultiDatos&idp=1254734710990, 2023.
- [21] NetworkX Developers. Greedy modularity communities. https://networkx.org/documentation/stable/_modules/networkx/algorithms/community/modularity_max.html#greedy_modularity_communities, 2023.
- [22] Didsoft. Servidores proxies gratuitos actualizados. <https://free-proxy-list.net/>, 2023.
- [23] MDN Web Docs. Agentes de usuario. <https://developer.mozilla.org/es/docs/Web/HTTP/Headers/User-Agent>, 2022.

- [24] Hugging Face. Transformers python library. <https://huggingface.co/docs/transformers/index>.
- [25] Glassdoor. Búsqueda de sueldos y remuneración en reino de españa. <https://www.glassdoor.es/Sueldos/index.htm>, 2023.
- [26] HIBP. Plataforma verificación de datos comprometidos. <https://haveibeenpwned.com/About>, 2023.
- [27] IEEE. Recommended practice for software requirements specifications. <https://standards.ieee.org/ieee/830/1222/>, 2009.
- [28] Juha Nurmi. Motor de búsqueda de la darknet. <https://ahmia.fi/documentation/>, 2023.
- [29] Octoparse. Servidores proxies para el raspado de datos. <https://www.octoparse.es/blog/utilizar-el-servidor-proxy-para-web-scraping>, 2023.
- [30] U.S. Department of Defense. NATIONAL DEFENSE AUTHORIZATION ACT FOR FISCAL YEAR 2006. <https://www.govinfo.gov/content/pkg/PLAW-109publ163/pdf/PLAW-109publ163.pdf>.
- [31] Scrum org. What is scrum?. <https://www.scrum.org/learning-series/what-is-scrum>.
- [32] Pallets Projects. Documentación seguridad en flask. <https://flask.palletsprojects.com/en/2.3.x/security/>.
- [33] ProxyMesh. Proxy anonymity levels, elite proxies. [https://docs.proxymesh.com/article/78-proxy-anonymity-levels#:~:text=Elite%20Proxies%20\(Level%201\),was%20made%20through%20a%20proxy](https://docs.proxymesh.com/article/78-proxy-anonymity-levels#:~:text=Elite%20Proxies%20(Level%201),was%20made%20through%20a%20proxy).
- [34] Joseph E. Roop. FBIS - History Part I: 1941-1947. <https://apps.dtic.mil/sti/pdfs/ADA510770.pdf>.
- [35] Scale SERP. Documentación scaleserp google search api. <https://www.scaleserp.com/docs/search-api/searches/google/search>.
- [36] Inc. The Tor Project. Historia del proyecto tor. <https://www.torproject.org/es/about/history/>.
- [37] Thomas Wood. Softmax function definition. <https://deepai.org/machine-learning-glossary-and-terms/softmax-layer>.
- [38] Intelligence X. Plataforma de búsqueda de información sensible. <https://intelx.io/about>, 2023.