Escuela Politécnica Superior de Jaén

Grado en Ingeniería Informática

UNIVERSIDAD DE JAÉN

Trabajo Fin de Grado

RECOPILACIÓN DE INFORMACIÓN EN INTERNET MEDIANTE TÉCNICAS NO INTRUSIVAS

Alumno: Jesús López Chaichío

Tutor: Manuel José Lucena López
Dpto: Informática

Septiembre, 2022

Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don Manuel José Lucena López, tutor del Proyecto Fin de Carrera titulado: Recopilación de información en internet mediante técnicas no intrusivas, que presenta Jesús López Chaichío, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2022

El alumno:

Jesús López Chaichío

LOPEZ CHAICHIO JESUS - 77433559B

Firmado digitalmente por LOPEZ CHAICHIO JESUS - 77433559B
Fecha: 2022.09.07 15:42:43 +02'00'

Los tutores:

Manuel José Lucena López

Firmado digitalmente por LUCENA LOPEZ, MANUEL JOSE.
Fecha: 07.09.2022 - 09:15:10.

Índice

CAPÍTULO 1. Introducción 5
1.1 Introducción al trabajo 6
1.2 Objetivos 7
CAPÍTULO 2: Fuentes abiertas de Información 8
2.1 Fuentes de datos en internet 9
2.2 OSINT (open-source intelligence) 12
2.2.1 Fases del OSINT 13
2.3 Motores de búsqueda 14
2.3.1 Hacking de buscadores 15
2.3.2 Búsquedas y qué podemos encontrar 18
2.4 Herramientas 23
2.4.1 TheHarvester 24
2.4.2 Crt.sh 27
2.4.3 DNSdumpter 31
2.4.4 Shodan 33
2.4.5 Wayback Machine 37
2.4.6 Foca 38
2.4.7 IntelligenceX 40
2.4.8 Maltego 41
2.5 Herramientas en Redes Sociales 46
2.5.1 Nombres de usuario 46
2.5.2 Osintgram 48
2.5.3 Socialbearig 49
CAPÍTULO 3: Ingeniería del Software 51
3.1 Proceso de ingeniería del software 52
3.2 Análisis 53
3.2.1 Requerimientos funcionales 53
3.2.2 Requerimientos no funcionales 54
3.3 Diagramas de Casos de uso 54
3.4 Diseño 57
3.4.1 Diagrama de clases 57
3.4.2 Diagramas de secuencia 58
3.5 Implementación 63

3.5.1 Lenguaje de programación 63
3.5.2 Herramienta de desarrollo 63
3.5.3 Implementación del código fuente 64
3.6 Pruebas 72
3.7 Mantenimiento y posibles mejoras 73
CAPÍTULO 4: Conclusiones 75
Bibliografía 77
Anexo: Instalación 78

CAPÍTULO 1. Introducción

1.1 Introducción al trabajo

Actualmente con el desarrollo que se ha producido de la tecnología durante estas últimas décadas ha sido posible que la mayoría de las personas puedan acceder a dispositivos (ordenadores, móviles, tablets, sistemas embebidos, etc) con la capacidad de poder conectarse a internet. También se ha facilitado la adquisición de una conexión a internet en todo momento, los costes de contratar este servicio han disminuido y mejorado sus características como ancho de banda, latencia y calidad entre otras.

En diez años el número de usuarios en internet ha crecido a más del doble. En la gráfica se puede observar un crecimiento más grande en los primeros años y un repunte en el periodo donde ha transcurrido la pandemia, la cual ha obligado a que muchas empresas y usuarios se actualicen de manera digital.

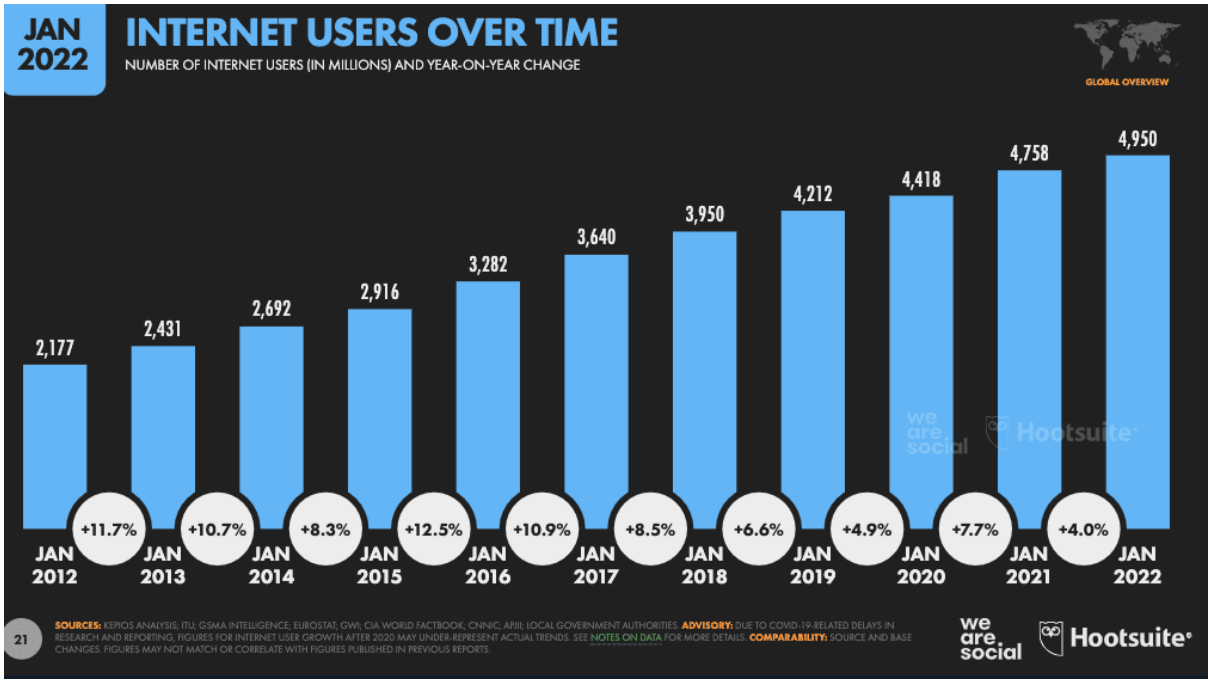


Ilustración 1. Estadística de "We Are Social and Hootsuite"

Por lo cual no ha ido solo en aumento el número de usuarios que utilizan internet, sino que también la cantidad de información que dejamos. Prácticamente la mayoría de servicios o necesidades ya pueden ser resueltas a través de internet con una mayor eficiencia. Todo esto por una parte va a ser bueno, básicamente toda "creación" tiene por objetivo hacer la vida del ser humano más fácil, por eso no siempre

pensamos en las desventajas que estas pueden tener mientras sea rápido, sencillo, gratis (nada es gratis en internet) y que no sea necesario pensar.

Surge también el concepto de la denominada “Huella Digital” la cual es todo aquello que deja rastro o un registro cuando un usuario utiliza internet. Esta información puede llegar a ser perjudicial sin el debido tratamiento puesto que esta es un reflejo de lo que hacemos, nuestros gustos, preferencias, ideas, etc. Un mal uso puede atacar a la seguridad y privacidad de los usuarios, dado que esta información puede ser recopilada.

1.2 Objetivos

El objetivo de este Trabajo de Fin de Grado es realizar un recorrido por las distintas fuentes de información disponibles en internet y los métodos de recopilación, además de realizar una prueba de concepto para demostrar su eficacia. Como:

1. Comprobar las distintas fuentes de datos disponibles en internet.
2. Usar distintos tipos de herramientas viendo el tipo de información que se puede obtener y cómo se extrae.
3. Diseñar un prototipo de aplicación capaz de obtener información de un usuario, a partir de sus redes sociales y fuentes abiertas de información.
4. Concienciar sobre el uso de internet y los riesgos que este puede traer, puesto que en muchos casos no somos conscientes de hasta qué punto ponemos en peligro nuestra privacidad.

CAPÍTULO 2: Fuentes abiertas de Información

2.1 Fuentes de datos en internet

Internet tiene casi 5.000 millones de usuarios en todo el mundo (de un total de casi 8000 millones) que diariamente generan grandes cantidades de datos y esto aumenta de manera exponencial.

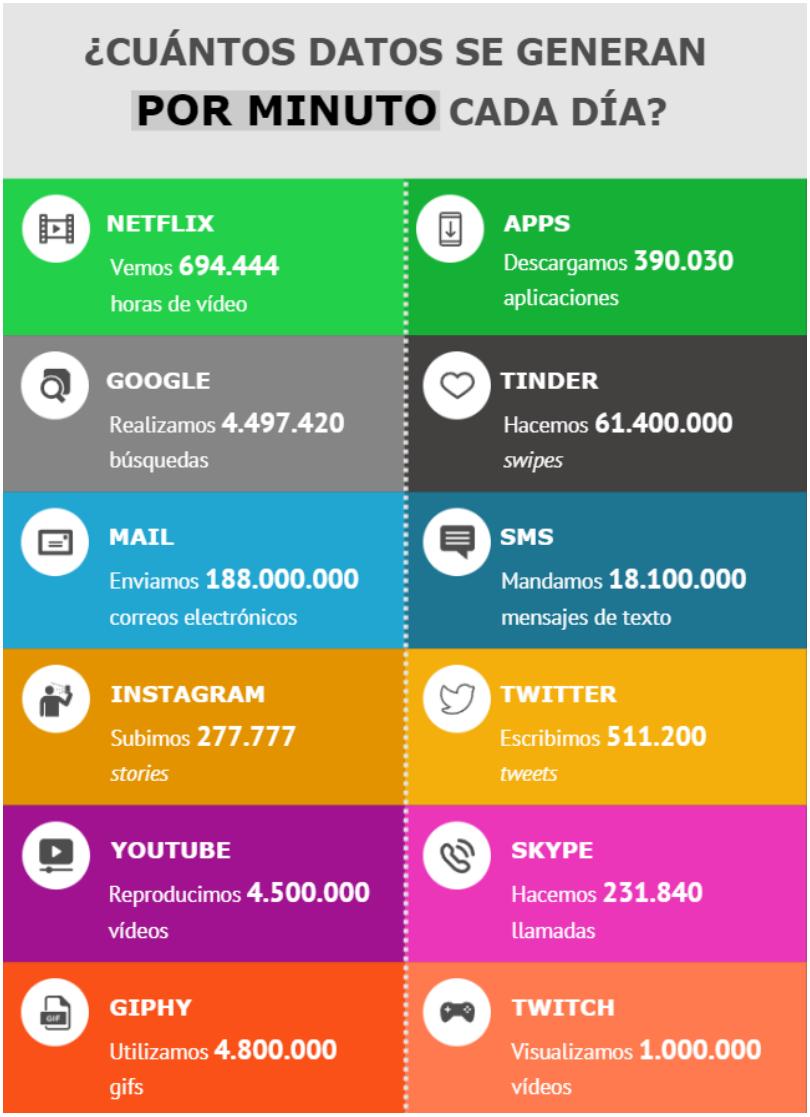


Ilustración 2. Datos generados por minuto

Toda esta información que se genera es almacenada y no por eso tiene que ser pública. Los datos más importantes suelen guardarse en servidores de los cuales una empresa/usuario tiene el control de acceso, aunque una gran parte si es pública o ha dejado un registro durante un tiempo determinado en el que se podía acceder a esta información. Algunas de las fuentes de datos más útiles que podemos encontrar son:

- **Ficheros de texto:** unos de los más básicos y útiles, los cuales son archivos de texto plano, sin tipografía. Son archivos simples que permiten que la mayoría de programas puedan leer y escribir en ellos.

- **Bases de datos:** de lo más utilizado actualmente, consiste en una herramienta donde recopilar y organizar información, para el uso de consultas obteniendo los datos de una manera rápida. Suelen estar estructuradas en tablas y son muy útiles para el análisis de datos.

- **CMS (sistemas de gestión de contenido):** nos permiten gestionar los contenidos que hay en una página web. Una de sus funciones principales es presentar la información con el objetivo de separar el contenido del diseño.

- **Foros:** son sitios web donde se publican mensajes con la capacidad de poder comunicar distintos usuarios entre sí, de manera que la idea es crear un hilo donde se habla de un tema de manera asíncrona.

- **Wikis:** son comunidades virtuales en las cuales los usuarios pueden agregar páginas, modificarlas y borrarlas. Estas se basan en la colaboración de una comunidad, el problema puede venir cuando se añade información falsa o poco contrastada dado que quien forma parte de esta comunidad tiene acceso a modificarla.

- **Bibliotecas digitales:** son colecciones de datos de distintos tipos con un cierto orden que cuentan con herramientas para su conservación y preservación, estas pueden ser públicas o privadas.

- **Blogs:** sitios web en los que un usuario puede publicar su propio contenido, estos suelen tener un carácter más personal donde el usuario puede hablar de distintos temas, incluyendo opiniones, experiencias, información relacionada con uno mismo, etc.

- **Redes sociales:** se podría decir que las redes sociales se han convertido en una fuente de datos debido a su creciente uso en el que se estima que más de un 50% de la población mundial usa alguna red social. Por esto investigar dentro de estas redes puede llegar a ser útil, puesto que encontramos gran cantidad de información actual y veraz, dado que en la mayoría de los casos son los mismos usuarios los que proporcionan esta información.

- **Canales RSS (Rich Site Summary):** son ficheros con formato XML que tienen como fin publicar resúmenes de sitios web, es decir, reducen la necesidad de tener que navegar para ver qué se ha añadido a un sitio web, muy útil para páginas que cambian constantemente. Normalmente para acceder a ellos es necesario suscribirse.

Como es normal la información que vayamos encontrando no tiene por qué estar actualizada. Aquí entraría una parte importante en la cual se debe examinar la veracidad de la información encontrada, puesto que no debemos quedarnos con lo primero que encontramos. Es necesario realizar una búsqueda y un contraste de datos exhaustivos con el objetivo de poder encontrar la mayor cantidad posible de información útil siendo de calidad y veraz.

Hay que destacar que la mayoría de la información que hay en internet circula por la Deep Web, alrededor de un 90%. Dentro de esta encontraríamos toda la información que no es indexada por los motores de búsqueda, es decir, no podemos encontrar estas páginas al realizar búsquedas dentro Google o Firefox por ejemplo. La Deep Web también incluiría a la Dark Web aunque no ocupa un amplio porcentaje de información ya que es usada para temas con un aspecto más inmoral o que sobrepasa la ilegalidad. Para poder acceder a estas distintas redes sería necesario el uso de programas específicos o motores de búsqueda que permitan acceder a estas páginas.

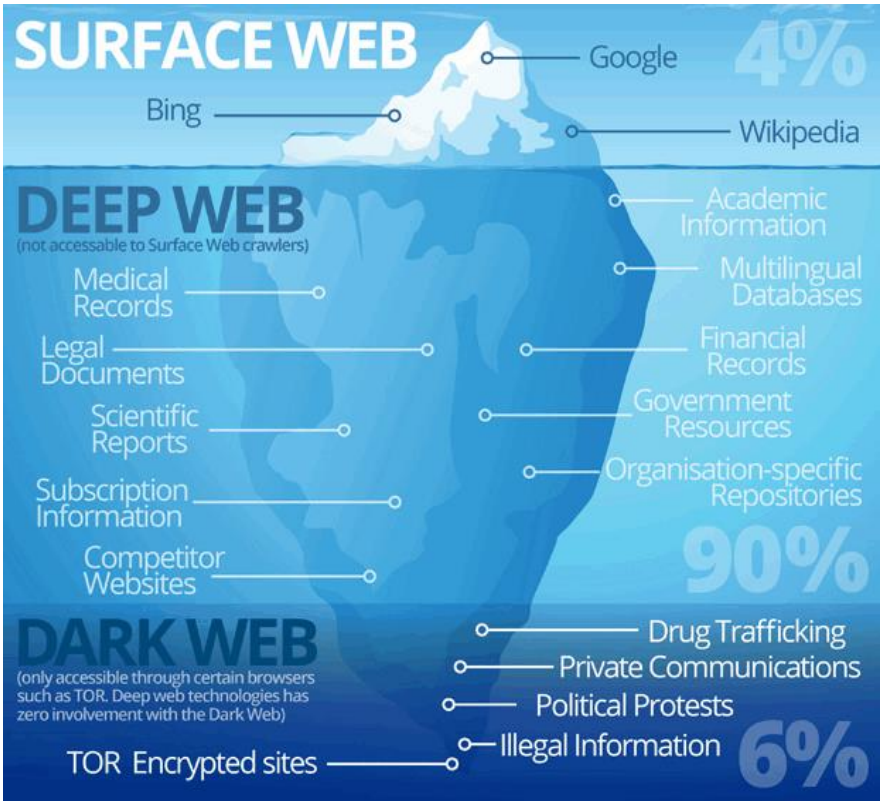


Ilustración 3. Estructura visual de la información en internet

2.2 OSINT (open-source intelligence)

Se puede considerar que el OSINT nace durante la Segunda Guerra Mundial derivado del OSS (Research and Analysis del Office of Strategic Services) cuya función era recopilar toda la información abierta posible, escuchando emisoras de radio extranjeras, obteniendo periódicos extranjeros y accediendo a fuentes de información o librerías de los estados.

Pero para definir este concepto empecemos por las fases que tendría un proceso de recopilación de información:

- Recopilación de información de forma **pasiva**: se basa en la recogida de información de un objetivo específico sin que las acciones de quien realiza esta tarea sean detectadas. Es una parte complicada de realizar puesto que en algunos casos los resultados no son concluyentes. La forma más habitual de obtener esta información es a partir de fuentes abiertas.

- Recopilación de información de forma **semi-pasiva**: podemos considerarla una ampliación de la recopilación pasiva puesto que se basa en “actuar de forma normal”, es decir, si visitamos el sitio web de nuestro objetivo para obtener información disponible esto sería una acción normal, pero ya implica un contacto con el objetivo.

- Recopilación de información de forma **activa**: en esta parte ya se entraría en un contacto más agresivo con el objetivo, debido a que nuestra actividad es más fácil de detectar ya que se genera más tráfico que es considerado como sospechoso o malicioso.

Entonces la Inteligencia de Fuentes Abiertas (OSINT) se basará en obtener cualquier tipo de información que no esté desclasificada y sea públicamente accesible para crear inteligencia a partir de esta, por esto se usa un enfoque más centrado en la recopilación pasiva de información. El proceso se fundamenta en búsqueda, selección, recopilación de la información y posteriormente de un procesamiento, análisis de los datos y correlación para convertirlos en conocimiento útil.

El OSINT se usa en distintos campos: militar (el cual se considera su origen), empresarial, económico, tecnológico, etc. El que más nos interesa es el tecnológico,

sobre todo enfocado a la ciberseguridad. Podemos destacar algunas de sus aplicaciones.

- **Disminución de riesgo de ciberataques:** buscar información sensible que pueda llegar a comprometer a nuestra organización o empresa.
- **Ingeniería social:** búsqueda de la información que hay disponible de un usuario, útil para atacar a un objetivo concreto o para examinar si hay información pública perjudicial de un usuario.
- **Auditorías:** con la finalidad de encontrar subdominios, hosts, ficheros públicos sin conocimiento de una empresa, DNS, etc.

Al igual que un usuario puede utilizar el OSINT para “el bien”, no debemos olvidar que los ciberdelincuentes también van a hacer uso de este con diferentes fines. Por tanto, esta parte de la ciberseguridad ha adquirido mucha importancia para las empresas, las cuales no quieren tener expuesta información sensible que pueda tener como consecuencia brechas de seguridad y pérdidas de reputación.

2.2.1 Fases del OSINT

Podemos definir una estructura con la que aplicar OSINT para optimizar el proceso de recopilación de información.

- **Requisitos:** en la primera fase estableceremos las necesidades que queremos cubrir. Un pequeño esquema o lista de distintos objetivos e información que queremos obtener puede sernos de utilidad para comenzar.
- **Fuentes de información:** examinar las distintas fuentes de datos que pueden ser más relevantes para obtener nuestro objetivo.
- **Adquisición:** obtención de información de las fuentes de datos examinadas en la fase anterior. Es más importante obtener información de calidad que una gran cantidad de datos de los cuales no podamos llegar a obtener un resultado concluyente, dado que los datos que obtengamos en esta fase van a determinar el desarrollo de las siguientes fases.
- **Procesamiento:** consiste en el tratamiento de la información adquirida y ajustarla a un formato con el cual sea más fácil trabajar.

- **Análisis:** se genera inteligencia con el análisis de los datos obtenidos, de forma que eliminará la información inservible, que no tenga relación con lo que buscamos o que no pueda aportar suficiente valor y veracidad a la investigación.

- **Inteligencia:** última fase en la cual se presenta toda la información útil obtenida durante el proceso, de manera que esta se presenta de forma comprensible para poder explotarla.

Ilustración 4. Fases del OSINT

2.3 Motores de búsqueda

Los motores de búsqueda aparecen como mecanismos que organizan, clasifican y proporcionan información proveniente de la red a usuarios que realizan consultas, mostrando los resultados de dichas consultas como listas de redirección hacia la web donde estaría la información. Estos realizan un escaneo constante de Internet para indexar los contenidos actuales o de nuevos sitios webs. Al estar indexados se obtienen mejores tiempos para servir los resultados, es decir, conseguimos una búsqueda más rápida. Aunque no siempre somos conscientes de la información que se indexa, por eso en algunos casos puede llegar a publicarse información sensible, sin que se llegue a tener conocimiento de esto.

2.3.1 Hacking de buscadores

El hacking de buscadores va a consistir en utilizar los distintos operadores que tienen integrados los buscadores, los cuales nos permiten filtrar las búsquedas y hacer que estas sean más precisas.

Se debería realizar un análisis de cuál es el motor de búsqueda que más nos interesa, pero los números hablan por sí solos, Google tiene el 91,4% del tráfico de búsquedas arrasando con sus demás competidores. Claramente interesa más utilizar los operadores de este, aunque no hay que descartar realizar las búsquedas que consideremos más importantes en los demás buscadores, pues tal vez se encuentre información que sea útil.

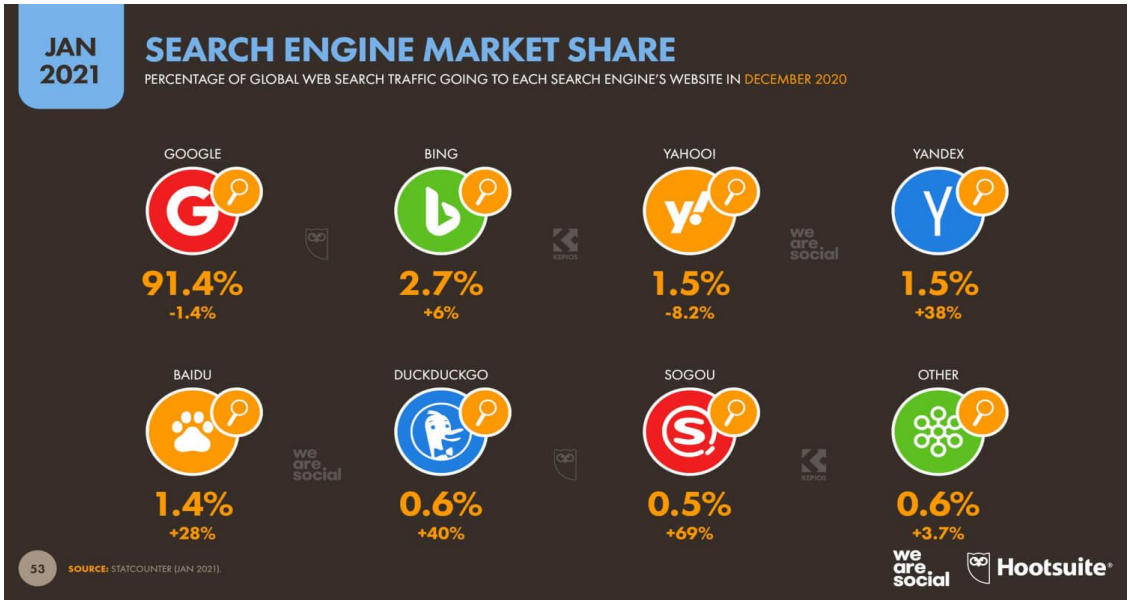


Ilustración 5. Tráfico de búsquedas

Algunos de los operadores avanzados que proporciona Google son:

- define: término - Muestra las definiciones que dan los distintos sitios web sobre ese término.
- filetype o ext: término - Las páginas mostradas acaban con el término introducido, es usado principalmente para búsqueda de archivos.
- site: dominio - Las búsquedas se restringen a ese sitio/dominio concreto.
- link: url - Muestra páginas que apuntan a la definida por dicha url.

- cache: url - Se muestra la versión definida por la url que tiene Google almacenada.
- info: url - Se presenta información correspondiente a la url.
- related: url - Para encontrar páginas similares a la url específica.
- intext: término - Muestra búsquedas las cuales el texto contiene el término introducido.
- allinurl: término - Solo presenta las busquedas que contiene el termino en la url

-Booleanos

- “ ”: Busca la palabra exacta que se escribe entre comillas.
- - : Excluye la palabra escrita seguida al guión.
- | : Sirve como operador OR, realiza búsquedas que tenga un término u otro.
- + : Incluye palabras que el buscador no tiene en cuenta por ser comunes.
- * : Se utiliza para sustituir palabras.

Todos estos operadores se pueden combinar para poder acotar mucho más una búsqueda, pero antes de pensar en las distintas consultas que vamos a hacer con estos operadores, la primera debería ser una búsqueda normal.

Ilustración 6. Ejemplo 1 de búsqueda en Google

Ilustración 7. Ejemplo 2 de búsqueda en Google

Como podemos ver, usar el navegador de manera común proporciona la información más relevante que tenga expuesta nuestro objetivo, esta puede ser interesante para empezar a hacernos una idea de cómo y qué vamos a poder obtener con futuras búsquedas. Además, como vemos en el primer ejemplo Google nos proporciona aproximadamente 19 millones de resultados relacionados con nuestra búsqueda, una cantidad excesiva que vamos a reducir utilizando operadores para centrarnos en información más específica.

2.3.2 Búsquedas y qué podemos encontrar

Una página muy interesante que podemos encontrar en internet, es “Google Hacking Database”, dentro de esta página podemos encontrar distintos “Google Dorks” (búsquedas utilizando operadores) publicados por usuarios.

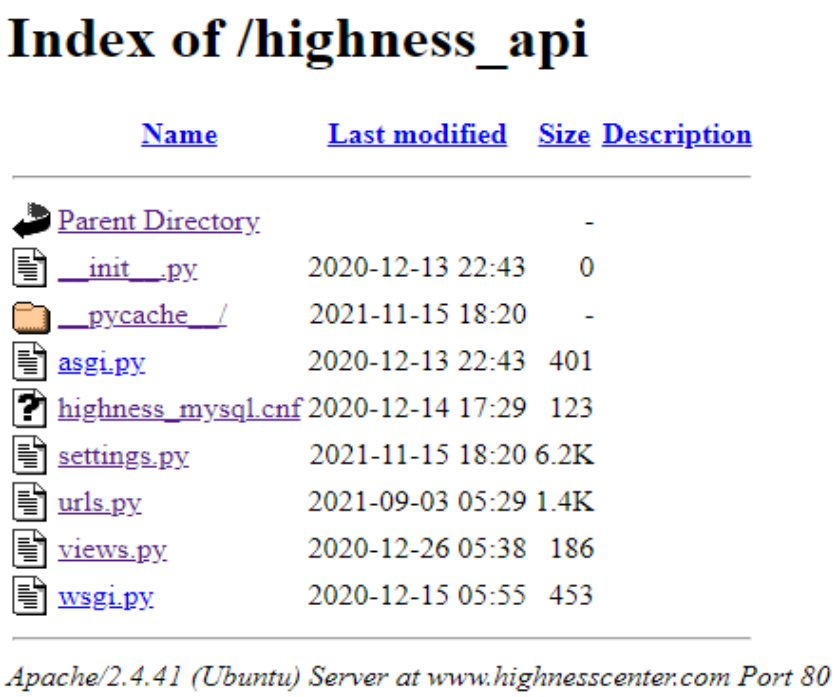
Ilustración 8. Google Hacking Database.

Veamos qué podemos encontrar utilizando alguno de estos Dorks.

- Directorios de aplicaciones que han sido indexados, los cuales nos pueden permitir conocer el funcionamiento de un servicio web al cual queremos acceder.

Búsqueda -> **intitle:index of settings.py**

Esta operación nos va a permitir encontrar archivos de ajustes escritos en Python, los cuales se encuentran en un directorio indexado de la aplicación, además podemos saber la versión del servidor web que utiliza y su sistema operativo.



Index of /highness_api

Name	Last modified	Size	Description
Parent Directory		-	
__init__.py	2020-12-13 22:43	0	
__pycache__/	2021-11-15 18:20	-	
asgi.py	2020-12-13 22:43	401	
highness_mysql.cnf	2020-12-14 17:29	123	
settings.py	2021-11-15 18:20	6.2K	
urls.py	2021-09-03 05:29	1.4K	
views.py	2020-12-26 05:38	186	
wsgi.py	2020-12-15 05:55	453	

Apache/2.4.41 (Ubuntu) Server at www.highnesscenter.com Port 80

Ilustración 9. Directorio de aplicación publica

Pulsando en “Parent Directory” nos redirecciona a una carpeta anterior, en la cual encontramos una carpeta llamada security que contiene un par de claves pública y privada. Esto no debería ser público ya que sólo quien crea este par de claves debería tener acceso a su clave privada.

Index of /security

Name	Last modified	Size	Description
Parent Directory		-	
highness_dev3_rsa	2020-12-13 22:45	3.2K	
highness_dev3_rsa.pem	2020-12-13 22:43	775	

Apache/2.4.41 (Ubuntu) Server at www.highnesscenter.com Port 80

Ilustración 10. Claves pública y privada.

- Archivos de usuarios y contraseñas que han sido hechos públicos o simplemente indexados dentro una aplicación.

Búsqueda -> **intext:"@gmail.com" intext:"password" inurl:/files/ ext:txt**

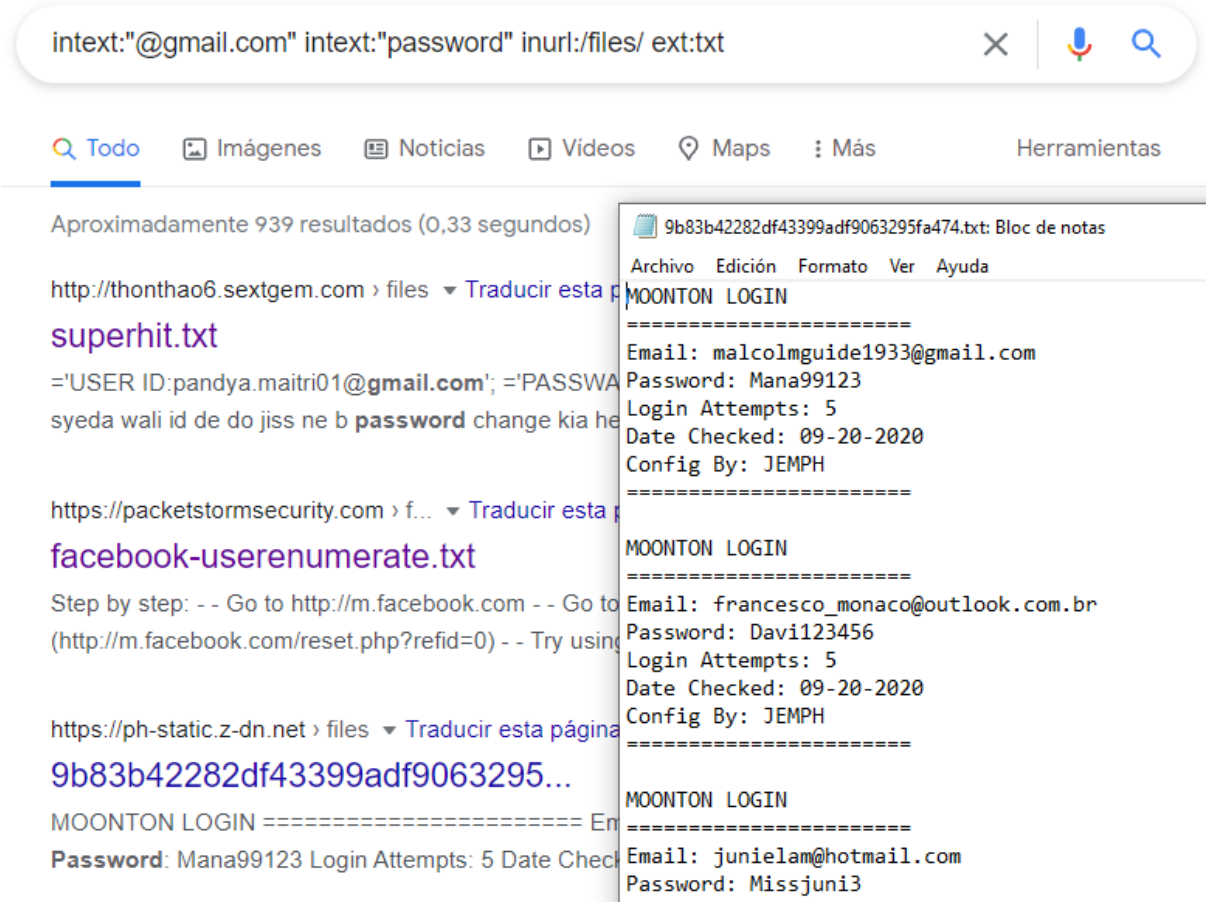


Ilustración 11. Direcciones de correo y contraseñas

- Archivos de bases de datos: con estos operadores encontraremos distintos ficheros de bases de datos de los cuales podemos obtener información de la estructura que tendría esta. Obtener la versión de un sistema siempre nos puede resultar útil para investigar si tiene vulnerabilidades que puedan ser explotadas, incluso dentro de estos ficheros podemos encontrar inserciones a la base de datos que contienen contraseñas.

Búsqueda -> **intitle: "dumping data for table" "password" filetype:sql**

Encontramos esto en las primeras líneas del fichero

```
-- phpMyAdmin SQL Dump
-- version 3.5.4
-- http://www.phpmyadmin.net
```

Si buscamos “CVE phpmyadmin 3.5.4” encontramos las diferentes vulnerabilidades que han sido reportadas.

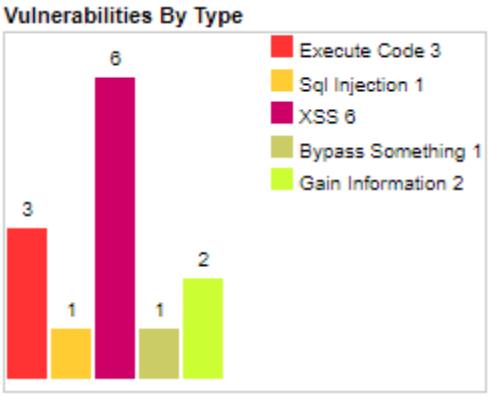


Ilustración 12. Vulnerabilidades relacionadas con el CVE

También encontramos una inserción de datos donde aparecen las contraseñas de usuario.

```
--
-- Table structure for table `reset_password`
--

DROP TABLE IF EXISTS `reset_password`;
/*!40101 SET @saved_cs_client     = @@character_set_client */;
/*!40101 SET character_set_client = utf8 */;
CREATE TABLE `reset_password` (
  `rp_username` varchar(100) DEFAULT NULL,
  `rp_code` varchar(20) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=cp1256 CHECKSUM=1 DELAY_KEY_WRITE=1 ROW_FORMAT=DYNAMIC;
/*!40101 SET character_set_client = @saved_cs_client */;

--
-- Dumping data for table `reset_password`
--

LOCK TABLES `reset_password` WRITE;
/*!40000 ALTER TABLE `reset_password` DISABLE KEYS */;
INSERT INTO `reset_password` VALUES ('Gamma1','Fj5$AJWypY17'),('Gamma1','SXgKXRPY6B9s'),('Gamma9','RBxNNmIogMuD')
/*!40000 ALTER TABLE `reset_password` ENABLE KEYS */;
UNLOCK TABLES;
```

Ilustración 13. Inserción SQL

- Robots.txt: estos ficheros consisten en establecer qué partes de una web quieres que Google no indexe, es decir, qué partes no mostrar al público, pero son accesibles. Muchas veces se indexan estos archivos, si se obtiene da la

posibilidad a un atacante de conocer determinadas zonas del sitio web, más privadas.

Búsqueda -> **inurl: /wp-content/uploads/ inurl:"robots.txt" "Disallow:" filetype:txt**

```
User-Agent: *
Allow: /wp-admin/admin-ajax.php
Allow: /wp-content/uploads/
Disallow: /wp-content/uploads/ithemes-security/
Allow: /wp-content/plugins/*.js
Allow: /wp-content/plugins/*.css
Allow: /wp-content/plugins/*.png
Disallow: /wp-content/plugins/
Disallow: /wp-content/backups-bbdd/
#Disallow: /wp-content/cache/
Allow: /wp-content/themes/*.js
Allow: /wp-content/themes/*.css
Allow: /wp-content/themes/*.pdf
Allow: /wp-content/themes/*.jpg
Allow: /wp-content/themes/*.png
#Disallow: /wp-content/themes/
Allow: /wp-includes/*.js
Allow: /wp-includes/*.css
Disallow: /wp-includes/
Disallow: /wp-admin/
Disallow: /trackback/
Disallow: /xmlrpc.php
Disallow: /wp-login
Disallow: /wp-admin
```

Ilustración 14. Fichero Robots.txt

2.4 Herramientas

Un recurso muy interesante con el cual podemos obtener una visión de toda la información que podemos a llegar conseguir de un objetivo, es OSINT Framework. Esta consiste en un conjunto de librerías de código libre que permite recopilar distintos tipos de datos, como por ejemplo:

- Nombres de usuario
- Emails
- Dominios
- Documentos
- Geolocalización

Este tiene una estructura de árbol cuyos nodos padres son las categorías de información que podemos obtener, donde al seleccionar un nodo se desplegará mostrando la fuente de la cual podemos obtener información.

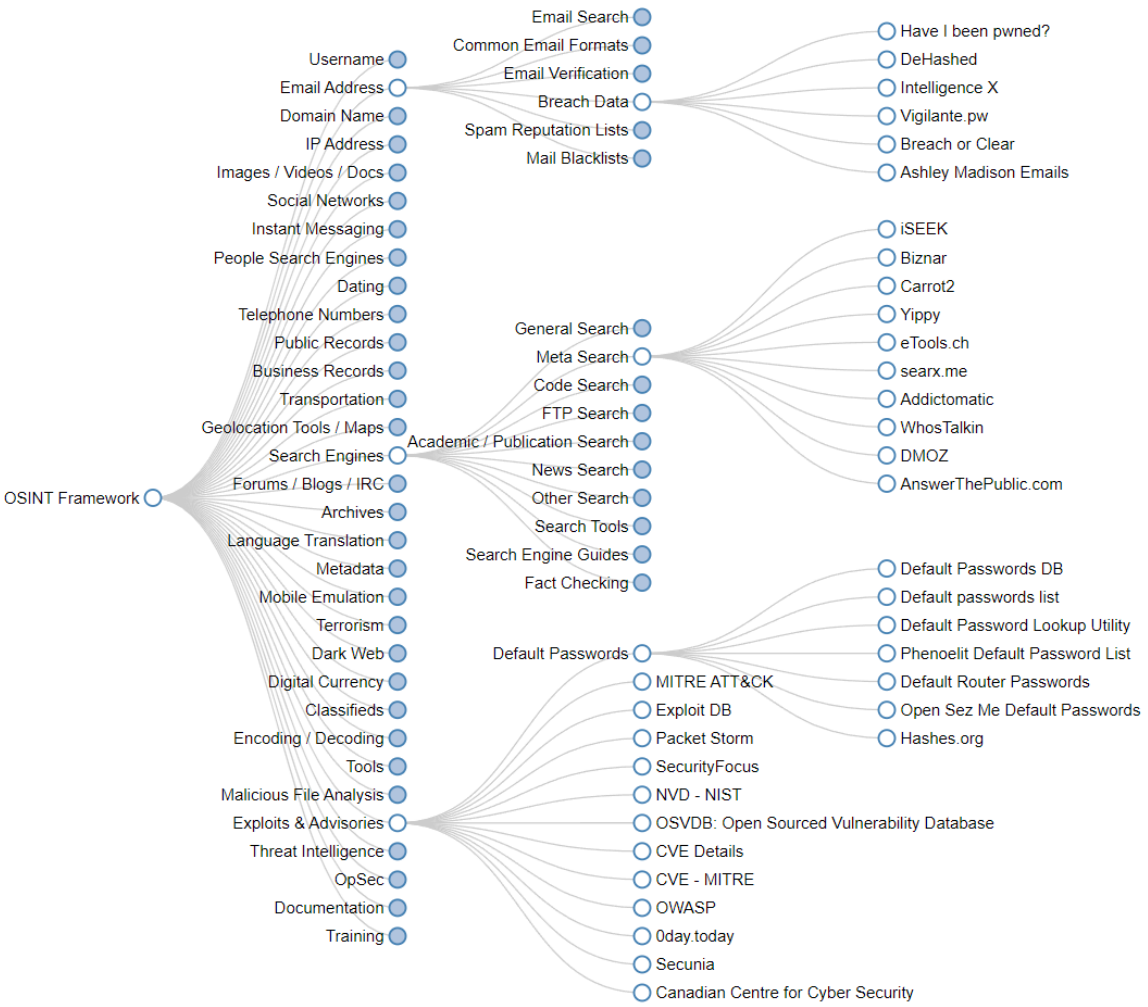


Ilustración 15. OSINTFramework

2.4.1 TheHarvester

TheHarvester es una herramienta de código abierto, enfocada a las primeras fases del proceso de obtener información, que permite conseguir correos electrónicos, nombres, subdominios, IPs, puertos abiertos, etc, mediante múltiples fuentes de datos públicas.

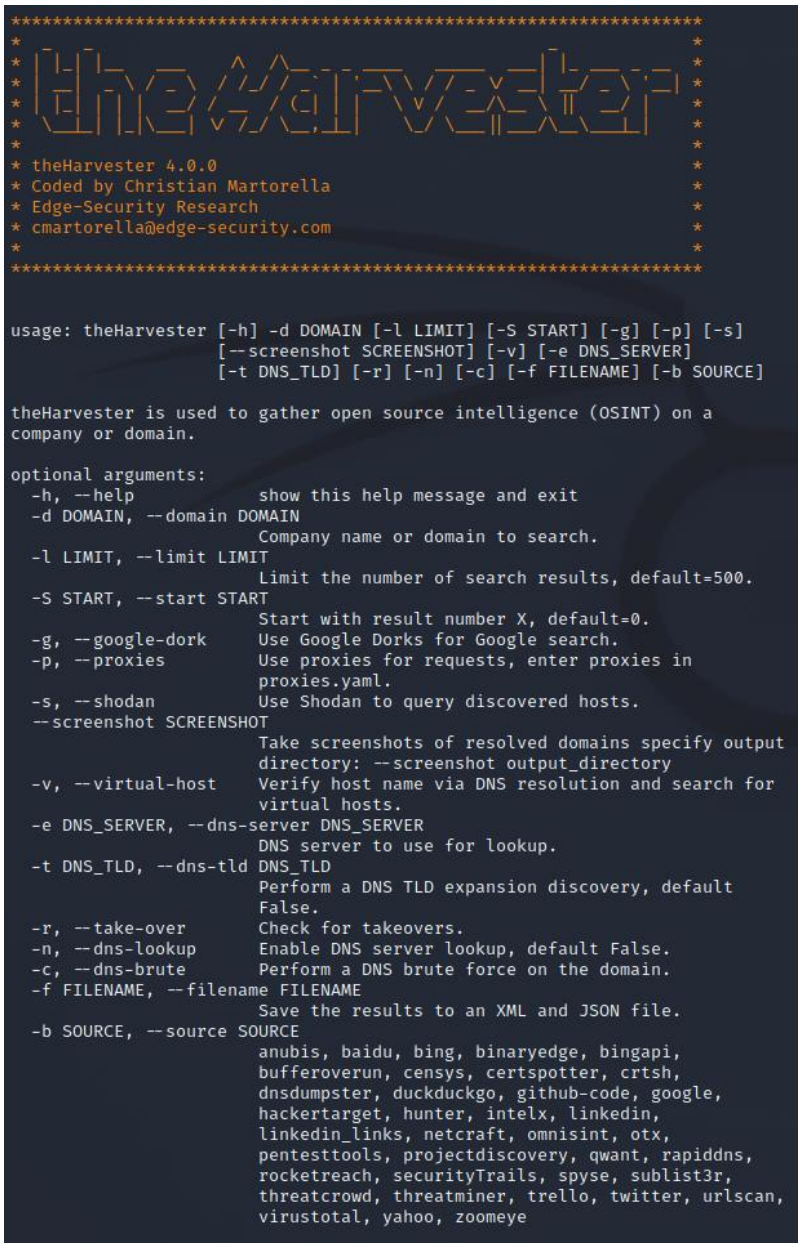


Ilustración 16. TheHarvester

Realizaremos una prueba para comprobar el alcance de esta herramienta. Mediante el siguiente comando por el cual, se examina el dominio ujaen.es en todas

las fuentes de información que componen la herramienta con un número de máximo de 1000 búsquedas, donde incluimos también la búsqueda de host virtuales almacenando toda la información que se obtenga en los ficheros results.json y results.xml

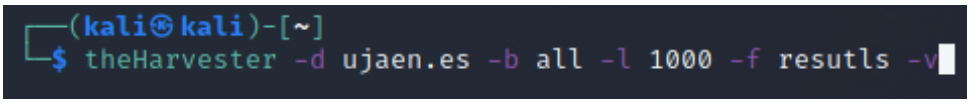


Ilustración 17. Estadística de "We Are Social y Hootsuite"

Se obtendrán aquellas URLs que se consideran más interesantes.

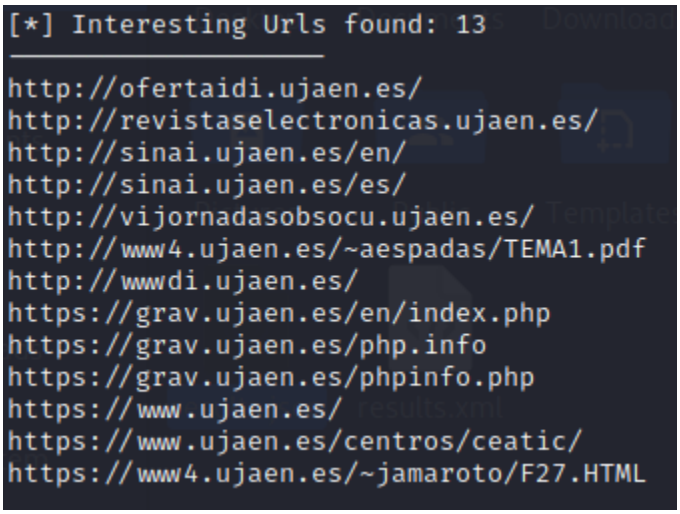


Ilustración 18. Resultado de URLs

Emails asociados al dominio.

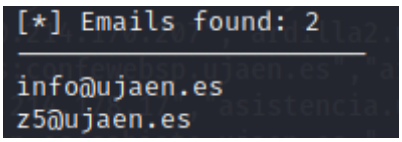


Ilustración 19. Emails encontrados.

Distintas IPs y Hosts que componen el dominio.

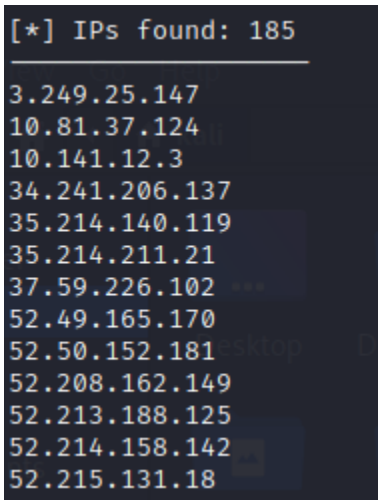


Ilustración 20. Lista de IPs

Ilustración 21. Lista de dominios

Finalmente se guarda toda la información encontrada, permitiendo la persistencia de los datos y la comparación de estos, puesto se recomienda realizar varias ejecuciones de una misma búsqueda.

2.4.2 Crt.sh

Se trata de una herramienta que permite obtener información sobre los subdominios de un dominio, los certificados de estos y las entidades certificadoras emisoras de dichos certificados, pudiendo detectar subdominios que contengan certificados erróneos o maliciosos.

Su funcionamiento es simple, deberemos introducir un dominio o el hash de un certificado.

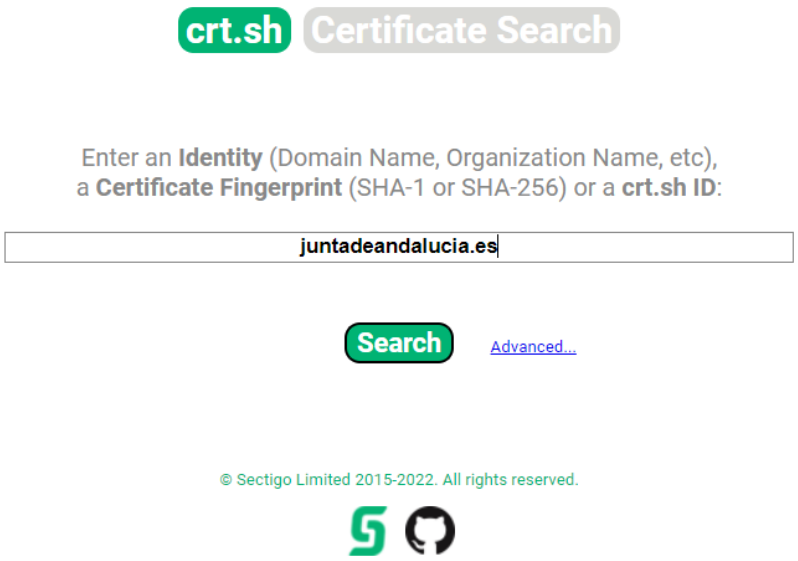


Ilustración 22. Crt.sh

Tras introducir por ejemplo un dominio, se nos muestra una lista de subdominios asociado a la entrada.

crt.sh Identity Search Group by Issuer

Criteria | Type: Identity | Match: ILIKE | Search: 'juntadeandalucia.es'

Certificates

crt.sh ID	Logged At ⇧	Not Before	Not After	Common Name	Matching Identities	Issuer Name
7154220705	2022-07-19	2022-07-19	2023-07-28	rcjcle01.juntadeandalucia.es	rcjcle01.juntadeandalucia.es www.rcjcle01.juntadeandalucia.es	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo RSA Domain Validation Secure Server CA
7154220748	2022-07-19	2022-07-19	2023-07-28	rcjcle01.juntadeandalucia.es	rcjcle01.juntadeandalucia.es www.rcjcle01.juntadeandalucia.es	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo RSA Domain Validation Secure Server CA
7154177853	2022-07-19	2022-07-19	2023-07-28	rcjcle02.juntadeandalucia.es	rcjcle02.juntadeandalucia.es www.rcjcle02.juntadeandalucia.es	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo RSA Domain Validation Secure Server CA
7154177611	2022-07-19	2022-07-19	2023-07-28	rcjcle02.juntadeandalucia.es	rcjcle02.juntadeandalucia.es www.rcjcle02.juntadeandalucia.es	C=GB, ST=Greater Manchester, L=Salford, O=Sectigo Limited, CN=Sectigo RSA Domain Validation Secure Server CA
6932337773	2022-06-14	2022-06-14	2023-06-14	*.epc.juntadeandalucia.es	*.epc.juntadeandalucia.es epc.juntadeandalucia.es	C=ES, O=Firmaprofesional S.A., organizationIdentifier=VATES-A62634068, OU=Security Services, CN=AC Firmaprofesional - Secure Web 2021
6879921085	2022-06-06	2022-06-06	2023-06-20	clustervcse.vcf.juntadeandalucia.es	clustervcse.vcf.juntadeandalucia.es vcf.juntadeandalucia.es vcseint1.vcf.juntadeandalucia.es vcseint2.vcf.juntadeandalucia.es	C=US, O=DigiCert Inc, OU=www.digicert.com, CN=GeoTrust RSA CA 2018
6839412509	2022-05-31	2022-05-31	2023-05-31	ap-pemea.epc.juntadeandalucia.es	ap-pemea.epc.juntadeandalucia.es	C=ES, O=Firmaprofesional S.A., organizationIdentifier=VATES-A62634068, OU=Security Services, CN=AC Firmaprofesional - Secure Web 2021
6806911173	2022-05-26	2022-05-26	2023-05-26	psap-pemea.epc.juntadeandalucia.es	psap-pemea.epc.juntadeandalucia.es	C=ES, O=Firmaprofesional S.A., organizationIdentifier=VATES-A62634068, OU=Security Services, CN=AC Firmaprofesional - Secure Web 2021
6786028289	2022-05-23	2022-05-23	2023-05-23	ws185.juntadeandalucia.es	ws185.juntadeandalucia.es	C=ES, O=Firmaprofesional S.A., organizationIdentifier=VATES-A62634068, OU=Security Services, CN=AC Firmaprofesional - Secure Web 2021

Ilustración 23. Lista de subdominios

Si seleccionamos un “crt.sh ID” podremos ver el certificado de un dominio con un mayor nivel de detalle, los mecanismos de revocación, los algoritmos de cifrado utilizados, clave pública del certificado, extensiones de certificado, precertificados, firmas, etc.

crt.sh ID 7154220705

Summary Leaf certificate

Certificate Transparency

Log entries for this certificate:

Timestamp	Entry #	Log Operator	Log URL
2022-07-19 07:02:56 UTC	26469925	Cloudflare	https://ct.cloudflare.com/logs/nimbus2023
2022-07-19 07:02:56 UTC	99538823	Google	https://ct.googleapis.com/logs/xenon2023

Revocation

Report a problem with this certificate to the CA

Mechanism	Provider	Status	Revocation Date	Last Observed in CRL	Last Checked (Error)
OCSP	The CA	Check	?	n/a	?
CRL	The CA	Not Revoked	n/a	n/a	2022-08-03 13:16:06 UTC
CRLSet/Blocklist	Google	Not Revoked	n/a	n/a	n/a
disallowedcert.stl	Microsoft	Not Revoked	n/a	n/a	n/a
OneCRL	Mozilla	Not Revoked	n/a	n/a	n/a

Certificate Fingerprints **SHA-256** 2BE67901BF9637F559A86005AC287A8559B33C0A9BA998B5B33BAB70042D3F94 **SHA-1** 43E8CEA34B5F7E14340F709A76A6B7C48C0510D5

Ilustración 24. Información sobre un certificado

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            d8:32:82:14:0c:96:14:30:dc:9c:c7:a3:ac:c4:49:87
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: (CA ID: 105493)
            commonName                = Sectigo RSA Domain Validation Secure Server CA
            organizationName          = Sectigo Limited
            localityName              = Salford
            stateOrProvinceName       = Greater Manchester
            countryName               = GB
        Validity
            Not Before: Jul 19 00:00:00 2022 GMT
            Not After : Jul 28 23:59:59 2023 GMT
        Subject:
            commonName                = rcjcle01.juntadeandalucia.es
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                RSA Public-Key: (2048 bit)
                Modulus:
                    00:ab:80:88:58:66:37:30:76:1f:07:18:dc:95:90:
                    ce:ff:79:05:65:73:64:af:66:6f:e9:6e:a1:61:fb:
                    80:63:67:aa:ce:e0:c6:ba:6d:92:6f:31:47:4e:4a:
                    ae:9c:0a:88:2d:6c:db:d4:ed:9f:2f:c6:47:3f:89:
                    3f:78:23:2c:51:44:23:b0:8a:ee:7a:1a:a9:16:89:
                    38:b6:0b:66:f1:0c:a7:14:6e:7a:78:2c:11:53:15:
                    02:c6:d5:0f:a4:7f:21:ab:07:1f:5a:dd:ea:6c:d5:
                    7c:bd:56:32:4b:f8:1c:e4:9b:15:76:cf:90:9c:dd:
                    dd:b4:97:64:6b:bc:59:7b:8f:68:b8:13:0f:9e:3c:
                    7a:51:42:da:35:21:57:e8:6a:dd:c7:63:77:48:52:
                    9f:71:87:dd:37:6d:12:78:53:90:12:da:e2:42:25:
                    ee:93:cf:96:d4:59:05:10:b9:93:7d:9d:ba:25:62:
                    9b:42:cf:5c:43:ef:f1:f5:2a:35:92:55:83:7d:a4:
                    09:64:04:26:1d:13:ea:ce:2b:58:77:17:a8:a5:d8:
                    96:d2:11:7e:a3:33:fb:16:59:37:ba:de:78:b4:dc:
                    e4:45:3c:a8:5f:cb:e2:df:1a:a7:b9:6b:16:17:a4:
                    dc:43:8f:84:de:ef:a5:fe:f9:67:c9:9e:6d:65:1f:
                    47:bb
```

Ilustración 25. Claves del certificado

Por otro lado, también se puede examinar la entidad certificadora, seleccionando en la columna “Issuer Name”.

crt.sh CA ID	105493
CA Name/Key	Subject: commonName = Sectigo RSA Domain Validation Secure Server CA organizationName = Sectigo Limited localityName = Salford stateOrProvinceName = Greater Manchester countryName = GB Subject Public Key Info: Public Key Algorithm: rsaEncryption RSA Public-Key: (2048 bit) Modulus: 00:d6:73:33:d6:d7:3c:20:d0:00:d2:17:45:b8:d6: 3e:07:a2:3f:c7:41:ee:32:30:c9:b0:6c:fd:f4:9f: cb:12:98:0f:2d:3f:8d:4d:01:0c:82:0f:17:7f:62: 2e:e9:b8:48:79:fb:16:83:4e:ad:d7:32:25:93:b7: 07:bf:b9:50:3f:a9:4c:c3:40:2a:e9:39:ff:d9:81: ca:1f:16:32:41:da:80:26:b9:23:7a:87:20:1e:e3: ff:20:9a:3c:95:44:6f:87:75:06:90:40:b4:32:93: 16:09:10:08:23:3e:d2:dd:87:0f:6f:5d:51:14:6a: 0a:69:c5:4f:01:72:69:cf:d3:93:4c:6d:04:a0:a3: 1b:82:7e:b1:9a:b9:ed:c5:9e:c5:37:78:9f:9a:08: 34:fb:56:2e:58:c4:09:0e:06:64:5b:bc:37:dc:f1: 9f:28:68:a8:56:b0:92:a3:5c:9f:bb:88:98:08:1b: 24:1d:ab:30:85:ae:af:b0:2e:9e:7a:9d:c1:c0:42: 1c:e2:02:f0:ea:e0:4a:d2:ef:90:0e:b4:c1:40:16: f0:6f:85:42:4a:64:f7:a4:30:a0:fe:bf:2e:a3:27: 5a:8e:8b:58:b8:ad:c3:19:17:84:63:ed:6f:56:fd: 83:cb:60:34:c4:74:be:e6:9d:db:e1:e4:e5:ca:0c: 5f:15 Exponent: 65537 (0x10001)
Certificates	crt.sh ID: 924467861 | Not Before: 2018-11-02 | Not After: 2030-12-31 | Issuer Name: C=US, ST=New Jersey, L=Jersey City, O=The USERTRUST Network, CN=USERTrust RSA Certification Authority

Ilustración 26. Entidad Certificadora.

También contará con un apartado de confianza para diferentes sistemas.

Issued Certificates

Population	Unexpired	Expired	TOTAL
Certificates	17401812	121067949	138469761
Precertificates	17947193	121071451	139018644
TOTAL	35349005	242139400	277488405

Select search type: IDENTITY — commonName (Subject), emailAddress (Subject), organizationalUnitName (Subject), organizationName (Subject), dNSName (SAN), rfc822Name (SAN), iPAddress (SAN)

Enter search term: (% = All certificates) Search

Search options: Autoselect Identity matching; Exclude expired certificates?; Deduplicate (pre)certificate pairs?; Show SQL?; Or, Search on censys?

Trust

Purpose	Context (Version) Shortest Path Disabled From NotBefore Until									
	360 Browser (2021-08-05)	Apple (macOS 12.4)	Microsoft (2022-06-28)	Mozilla (2022-06-14)	Chrome (2020-05-15)	Android (2021-10-06)	Java (16.0.1)	Adobe CDS	Adobe AATL (2022-05-31)	Adobe EUTL (2022-06-09)
EV Server Authentication (1.3.6.1.4.1.6449.1.2.1.5.1)	n/a	Valid [2]	Valid [2]	Valid [2]	Valid [2]	n/a	n/a	n/a	n/a	n/a
EV Server Authentication (1.3.6.1.4.1.782.1.2.1.8.1)	n/a	No	No	No	Expired [3]	n/a	n/a	n/a	n/a	n/a
EV Server Authentication (2.23.140.1.1)	n/a	Valid [2]	No	No	No	n/a	n/a	n/a	n/a	n/a
EV Server Authentication (2.23.140.1.3)	n/a	No	Valid [2]	No	No	n/a	n/a	n/a	n/a	n/a
Server Authentication	Valid [2]	Valid [2]	Valid [2]	Valid [2]	Defer to OS	Valid [2]	Valid [2]	n/a	n/a	n/a
Client Authentication	n/a	n/a	Valid [2]	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Secure Email	n/a	No	No	No	n/a	n/a	n/a	n/a	No	No
Code Signing	n/a	No	No	n/a	n/a	n/a	No	n/a	No	No
Kernel Mode Code Signing	n/a	n/a	No	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Time Stamping	n/a	No	No	n/a	n/a	n/a	n/a	n/a	n/a	n/a
OCSP Signing	n/a	n/a	No	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Document Signing	n/a	n/a	No	n/a	n/a	n/a	n/a	n/a	No	No
Encrypting File System	n/a	n/a	No	n/a	n/a	n/a	n/a	n/a	n/a	n/a
IP security end system	n/a	n/a	No	n/a	n/a	n/a	n/a	n/a	n/a	n/a
IP security IKE intermediate	n/a	n/a	No	n/a	n/a	n/a	n/a	n/a	n/a	n/a
IP security tunnel termination	n/a	n/a	No	n/a	n/a	n/a	n/a	n/a	n/a	n/a
IP security user	n/a	No	No	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Adobe Authentic Document	n/a	n/a	n/a	n/a	n/a	n/a	n/a	No	No	No

Parent CAs: C=US, ST=New Jersey, L=Jersey City, O=The USERTRUST Network, CN=USERTrust RSA Certification Authority

Child CAs: *None found*

Ilustración 27. Reputación de la CA.

2.4.3 DNSdumpter

Esta herramienta se basa en el análisis de un dominio llegando a obtener información asociada con su Host. Para comprender un poco mejor lo que obtenemos, definamos qué hace un servidor DNS. Este se encarga de realizar la traducción de nombres de dominio a IPs. Estos servidores intervienen desde el momento que introducimos un nombre o un dominio en el navegador, ya que para acceder a estos es necesario conocer su IP. Introduciendo un dominio podemos obtener donde están alojados los servidores.

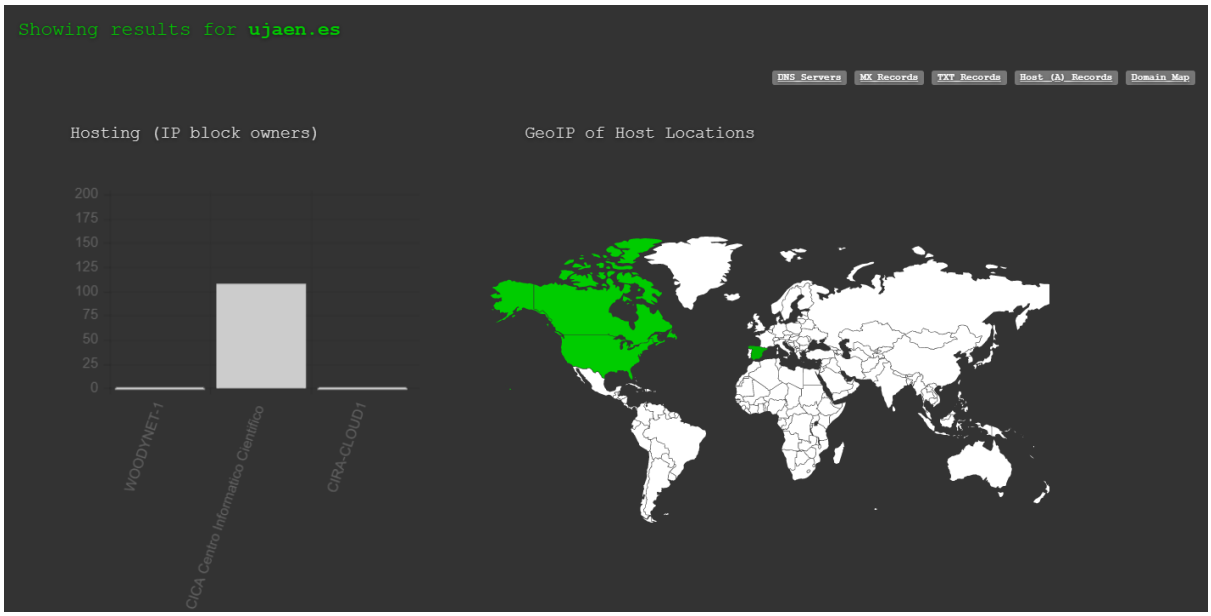


Ilustración 28. Localización de los servidores

Obtener los servidores DNS, con el nombre del subdominio y la IP asociada a este.

Ilustración 29. Servidores DNS

Registros MX que son un tipo de DNS usados para especificar en qué servidor deben recibirse los correos electrónicos.

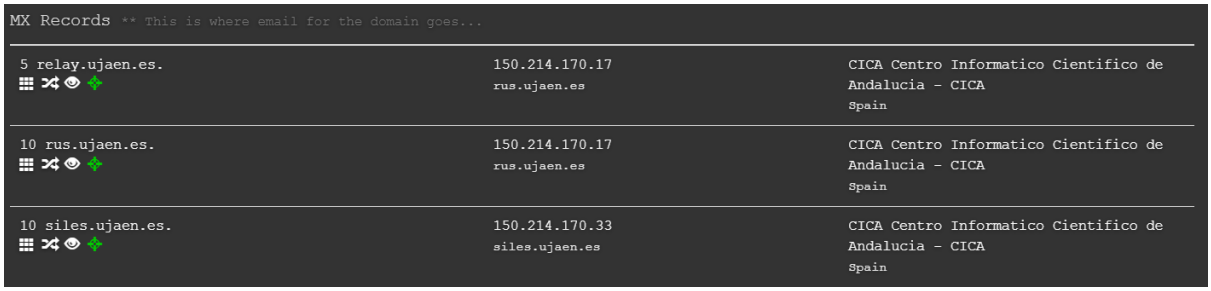


Ilustración 30. Registros MX

Registros TXT permiten añadir recursos sobre el DNS de manera legible para humanos.

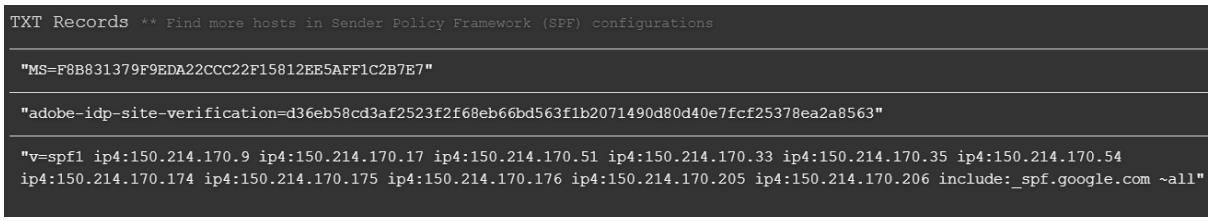


Ilustración 31. Registros TXT

Registros A contienen las direcciones IPs de un dominio.

Ilustración 32. Registros A

También aporta la opción de descargar la información encontrada como un fichero xlsx, un grafo o un mapa.

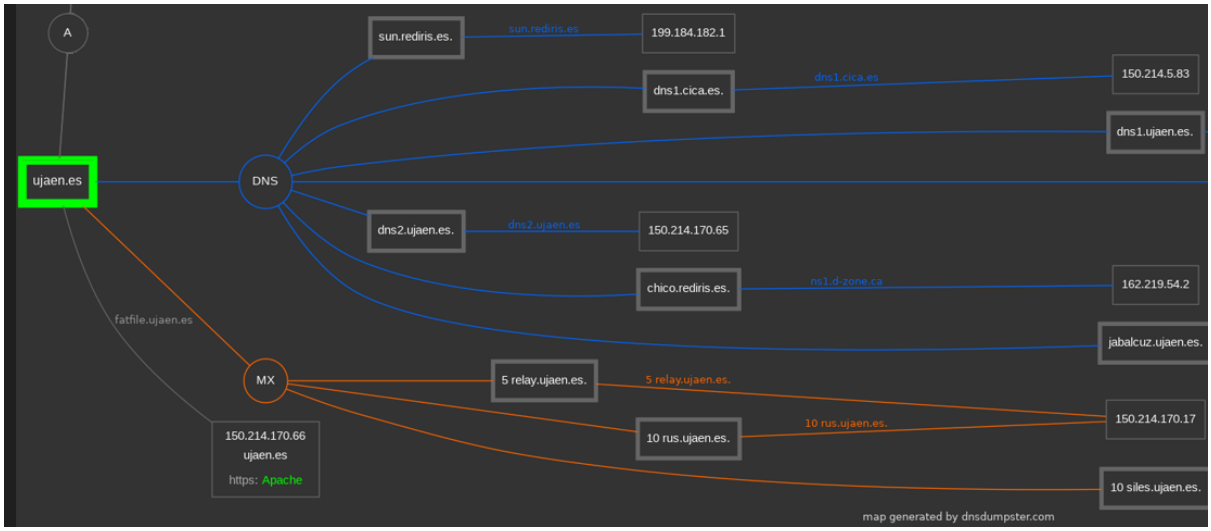


Ilustración 33. Mapa de la información obtenida

2.4.4 Shodan

Shodan es un motor de búsqueda, pero con la diferencia de que no es como los motores de búsqueda convencionales como Google, Bing, etc, dado que con estos podemos encontrar principalmente páginas web donde se sirven imágenes, documentos, productos, información del frontend entre otros. Con Shodan se puede obtener los sistemas y servidores que están expuestos en internet.

El funcionamiento de este motor de búsqueda consiste en recorrer internet realizando distintas peticiones a IPs y puertos asociados a estas, indexando de esta manera la información obtenida.

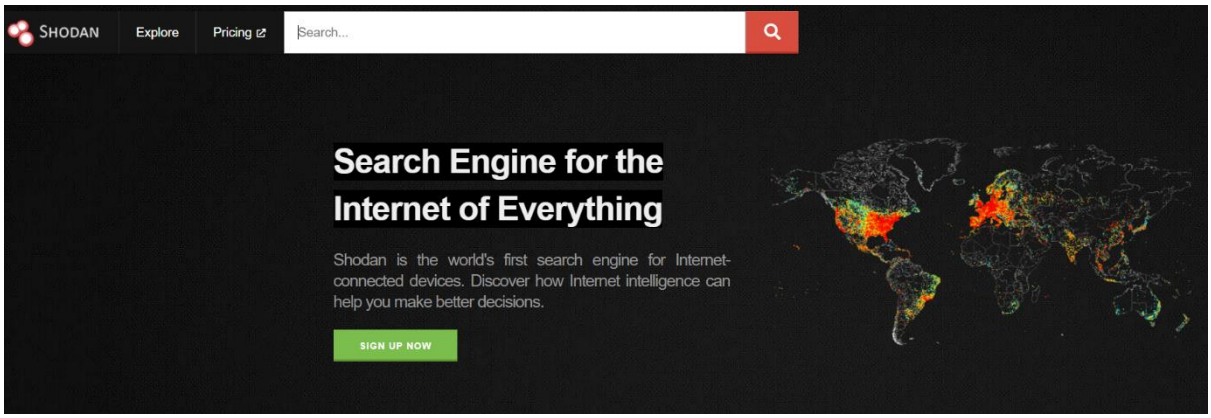


Ilustración 34. Shodan

De igual manera que Google tiene comandos para acotar y mejorar las búsquedas, Shodan también proporciona este tipo de comandos. Algunos de los más relevantes son:

- before/after: Muestra resultados antes/después de una fecha(dd/mm/aaaa)
- asn: Es un grupo de redes de direcciones IP que son gestionadas por uno o más operadores de red que poseen una clara y única política de ruteo.
- city: Nombre de una ciudad en concreto
- country: Filtrar por el código del país (2 caracteres: US)
- geo: Búsqueda por latitud y longitud.
- isp: Proveedor de internet
- net: Rango de red en notación CIDR (199.4.1.0/24)
- ip: Dirección IP.
- os: Sistema operativo.
- org: Nombre de empresa.
- vuln: Identificador de CVE.

Realizando la siguiente búsqueda**: ftp anonymous login ok country:"ES" port:"21"** solicitaremos obtener información sobre los sistemas alojados en España, los cuales utilicen el protocolo FTP y tengan abierto el puerto 21, además de permitir el login anónimo.

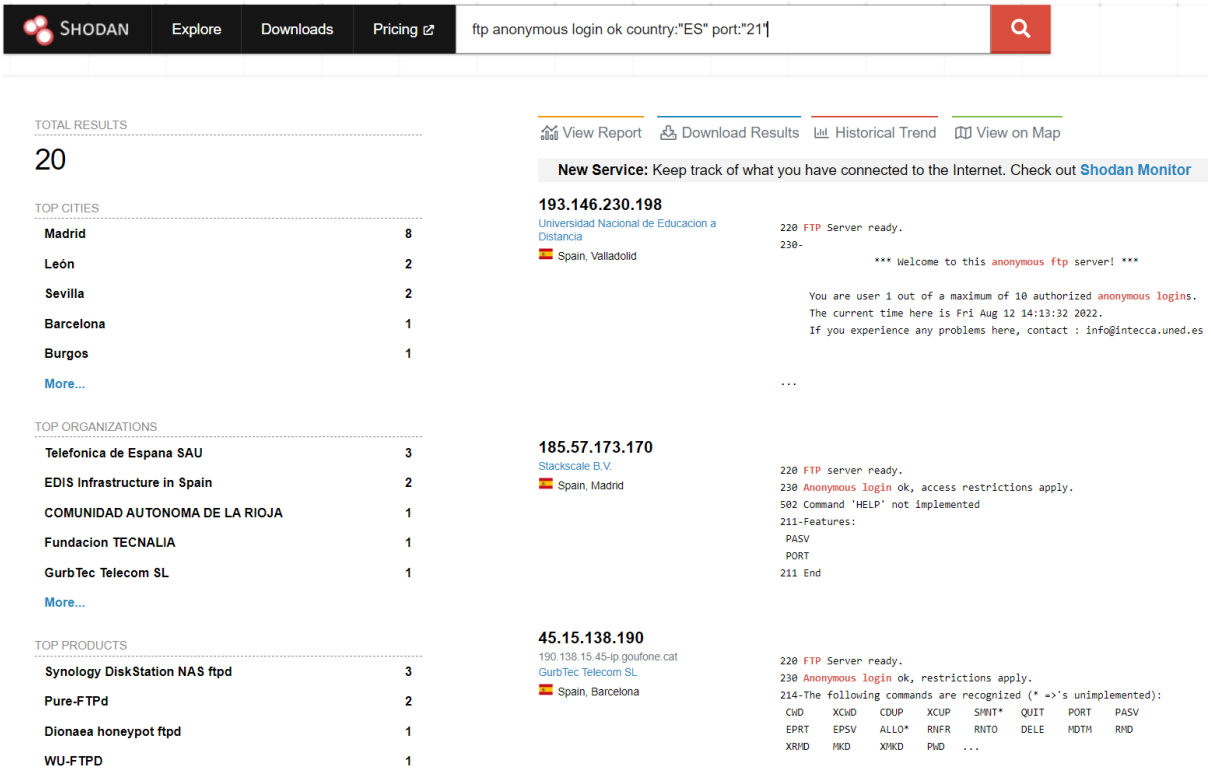


Ilustración 35. Búsqueda en Shodan

Seleccionando el primer resultado obtendremos lo siguiente:

1. Información de carácter general.

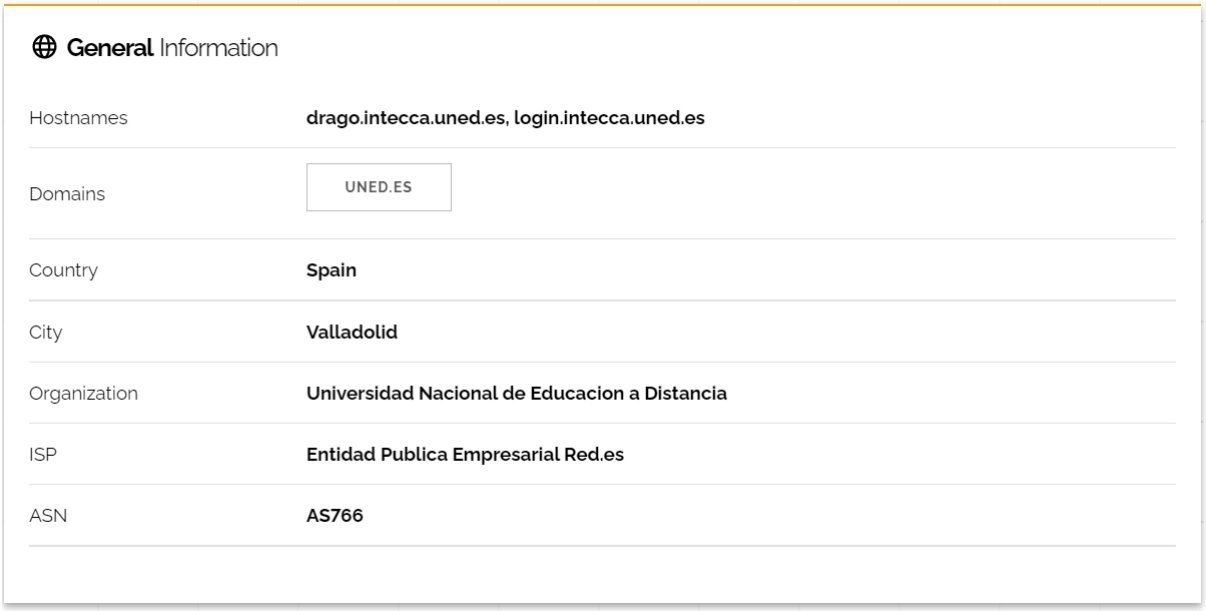


Ilustración 36. Información general.

2. Todos los puertos abiertos.

Ilustración 37. Puertos abiertos.

3. Información de cada uno de los puertos.

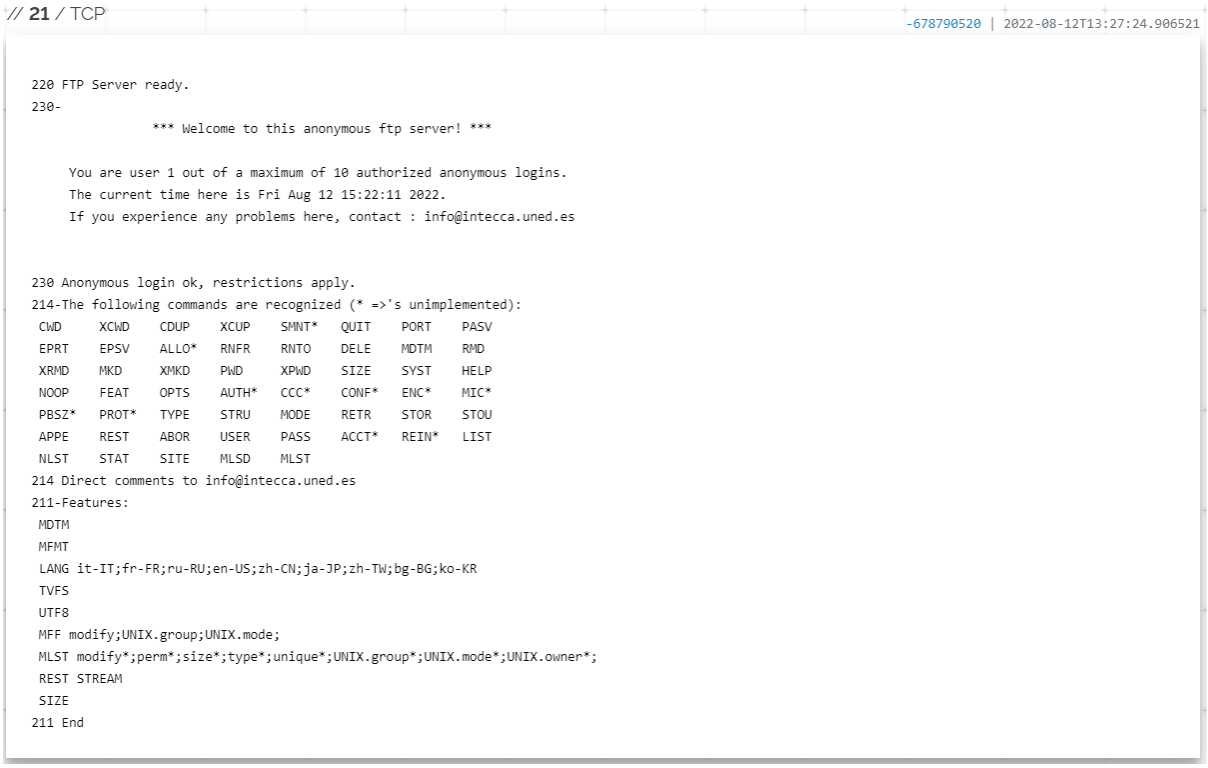
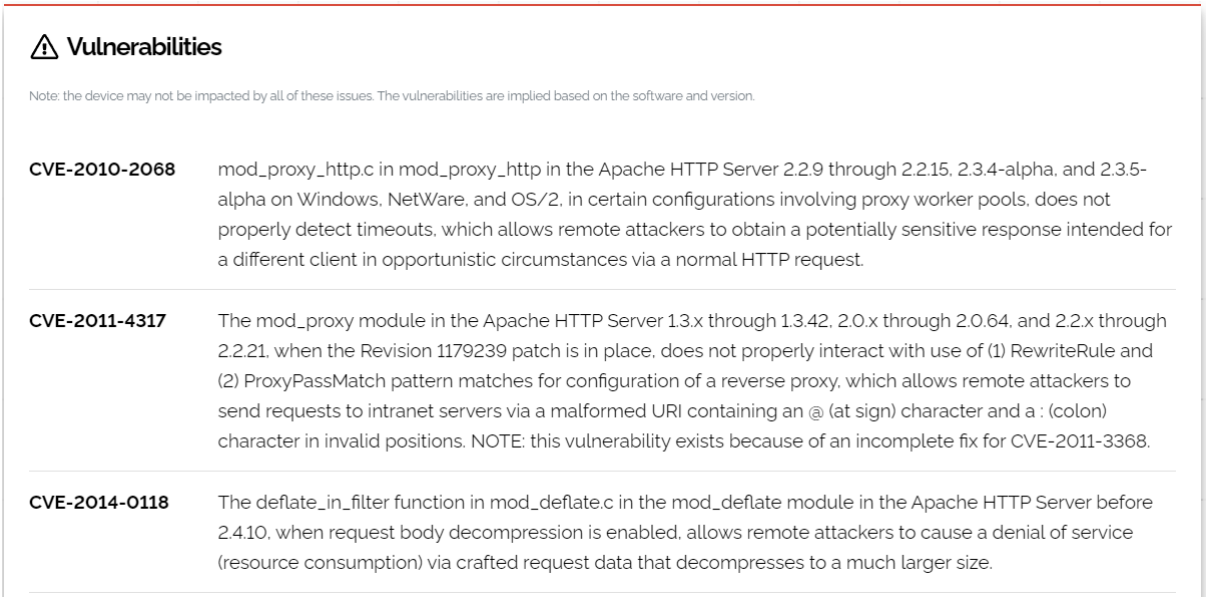


Ilustración 38. Información sobre un puerto.

4. Posibles vulnerabilidades de los servicios en los distintos puertos y su CVE asociado.



⚠ Vulnerabilities

Note: the device may not be impacted by all of these issues. The vulnerabilities are implied based on the software and version.

CVE-2010-2068	mod_proxy_http.c in mod_proxy_http in the Apache HTTP Server 2.2.9 through 2.2.15, 2.3.4-alpha, and 2.3.5-alpha on Windows, NetWare, and OS/2, in certain configurations involving proxy worker pools, does not properly detect timeouts, which allows remote attackers to obtain a potentially sensitive response intended for a different client in opportunistic circumstances via a normal HTTP request.
CVE-2011-4317	The mod_proxy module in the Apache HTTP Server 1.3.x through 1.3.42, 2.0.x through 2.0.64, and 2.2.x through 2.2.21, when the Revision 1179239 patch is in place, does not properly interact with use of (1) RewriteRule and (2) ProxyPassMatch pattern matches for configuration of a reverse proxy, which allows remote attackers to send requests to intranet servers via a malformed URI containing an @ (at sign) character and a : (colon) character in invalid positions. NOTE: this vulnerability exists because of an incomplete fix for CVE-2011-3368.
CVE-2014-0118	The deflate_in_filter function in mod_deflate.c in the mod_deflate module in the Apache HTTP Server before 2.4.10, when request body decompression is enabled, allows remote attackers to cause a denial of service (resource consumption) via crafted request data that decompresses to a much larger size.

Ilustración 39. Posibles vulnerabilidades.

2.4.5 Wayback Machine

Se trata de una base de datos que realiza regularmente copias de los diferentes sitios web que son indexados en internet, obteniendo así un histórico y permitiendo ver la evolución de estos sitios web a lo largo del tiempo.

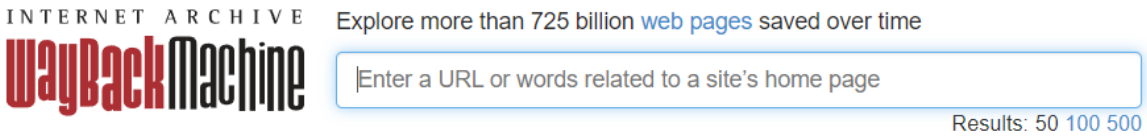


Ilustración 40. WaybackMachine

Cuando introducimos un sitio web obtenemos el número de copias que se han realizado a lo largo de un periodo de tiempo determinado. En la siguiente imagen podemos observar que para ujaen.es en el periodo del 17/6/1996 al 12/8/2022 se han realizado 6688 copias. También es posible filtrar por año, mes y día, lo cual aporta una gran utilidad si se busca una fecha en concreto, para ver la información publicada en un sitio web y que actualmente ya no estaría disponible.

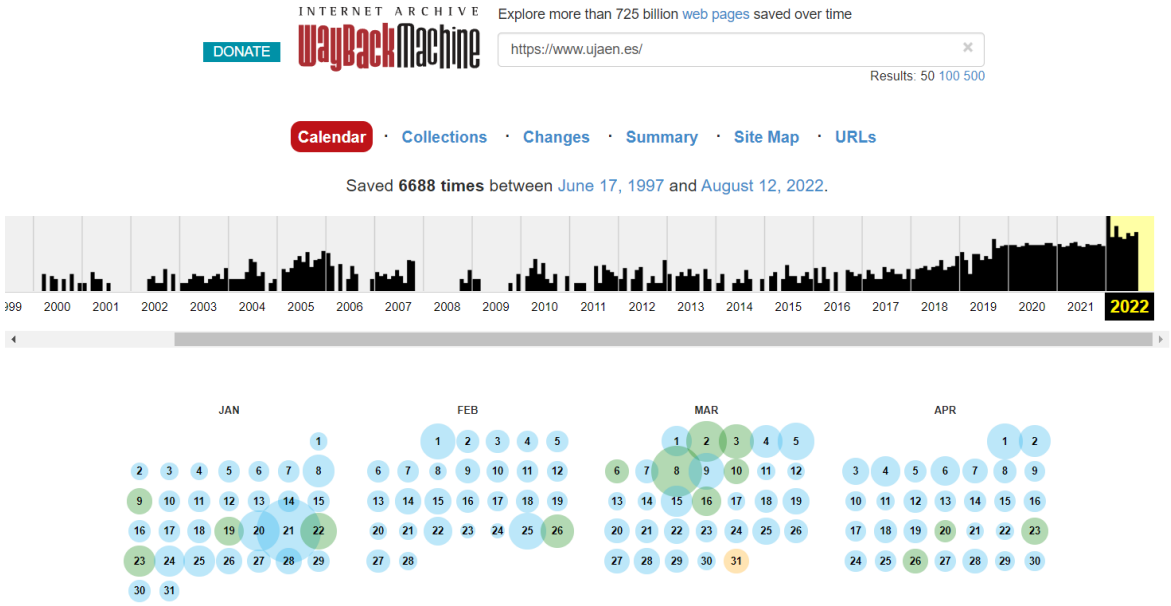


Ilustración 41. Línea temporal de datos almacenados

2.4.6 Foca

Es una herramienta cuya tarea principal es obtener metadatos. Esta examina un dominio específico buscado distintos tipos de ficheros públicos que poder descargar para un posterior análisis y extracción de los metadatos que posea.

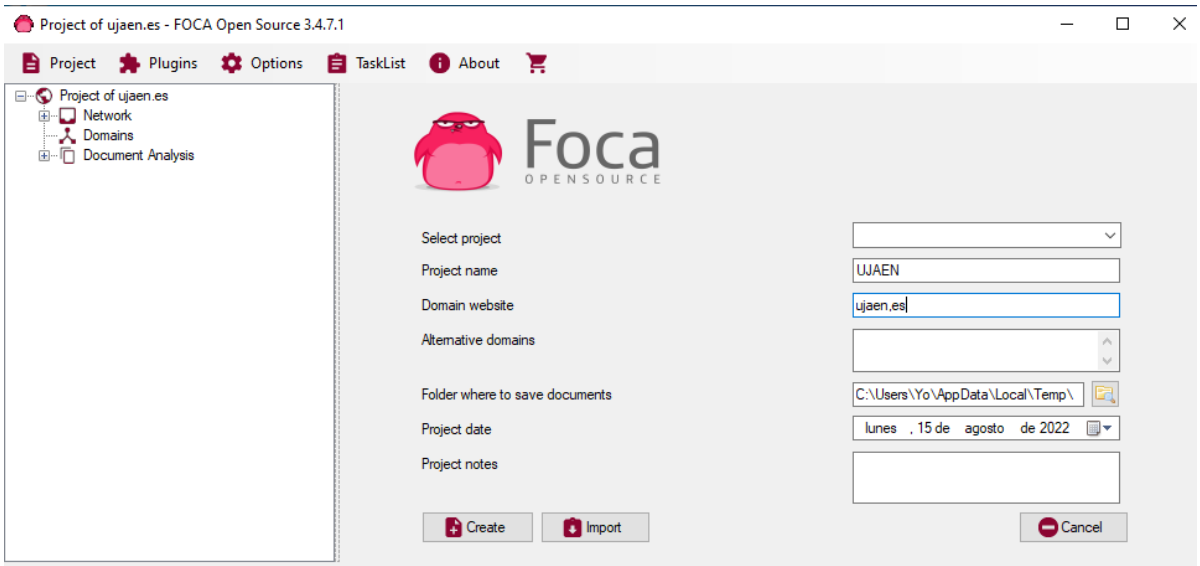


Ilustración 42. Foca

Mediante los navegadores Google, Bing y DuckDuckGo para el dominio ujaen.es en una ejecución, se llegan a encontrar 775 archivos descargables de los cuales poder extraer información.

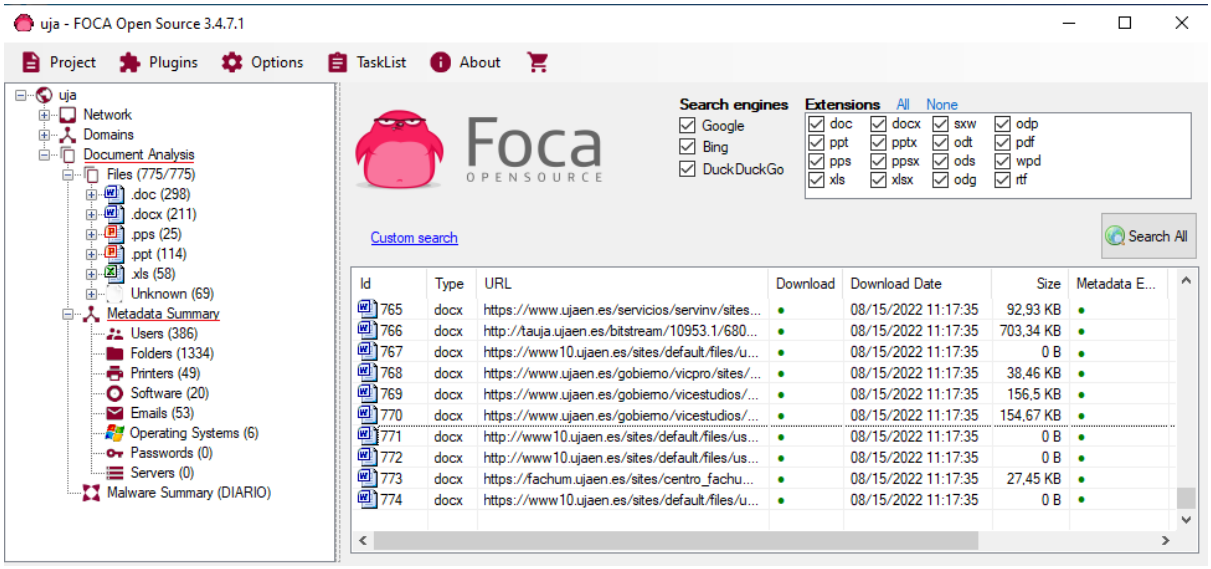


Ilustración 43. Obtención de ficheros públicos.

Seleccionando el apartado “Users” vemos como se muestra una lista de los nombres que se han encontrado en todos los ficheros. De igual forma podemos ver datos sobre las carpetas, impresoras, software, emails y sistemas operativos.

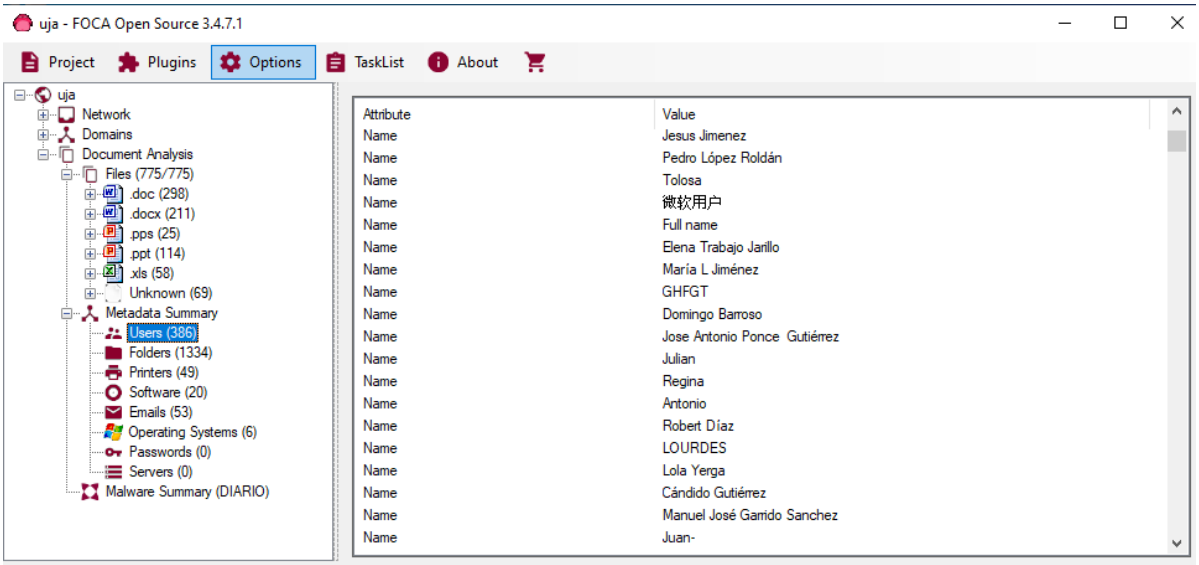


Ilustración 44. Ejemplo de metadatos conseguidos.

2.4.7 IntelligenceX

Es un sitio web donde se pueden encontrar filtraciones de datos de diferentes servicios o dominios de internet.

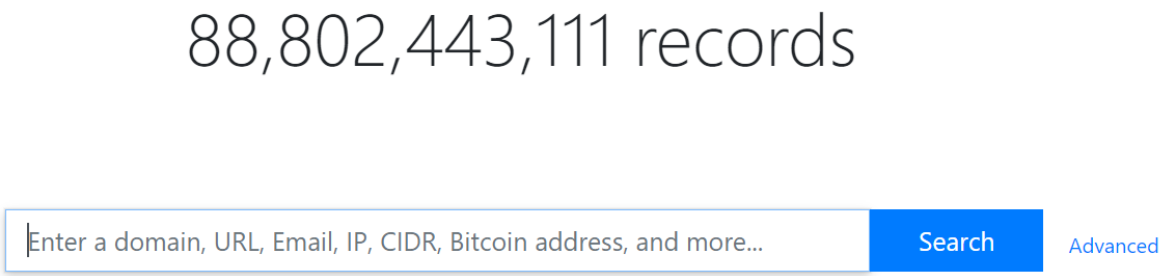


Ilustración 45. IntelligenceX.

Permite distintos parámetros de búsqueda, entre los cuales podemos encontrar dominios, IPs, URL, Hash, coordenadas, números de teléfono, correos, etc. Como resultado se obtienen distintos ficheros de datos, normalmente en formato de tabla.

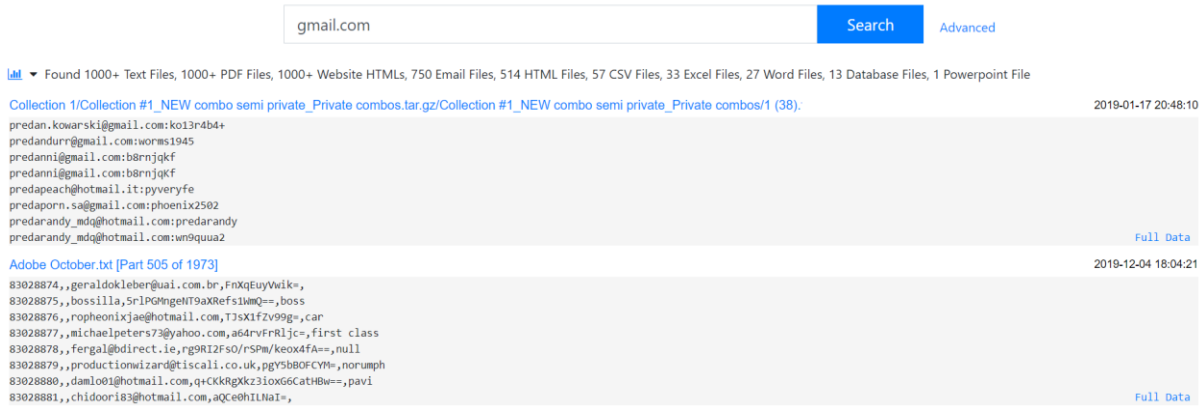


Ilustración 46. Lista de colecciones de datos.

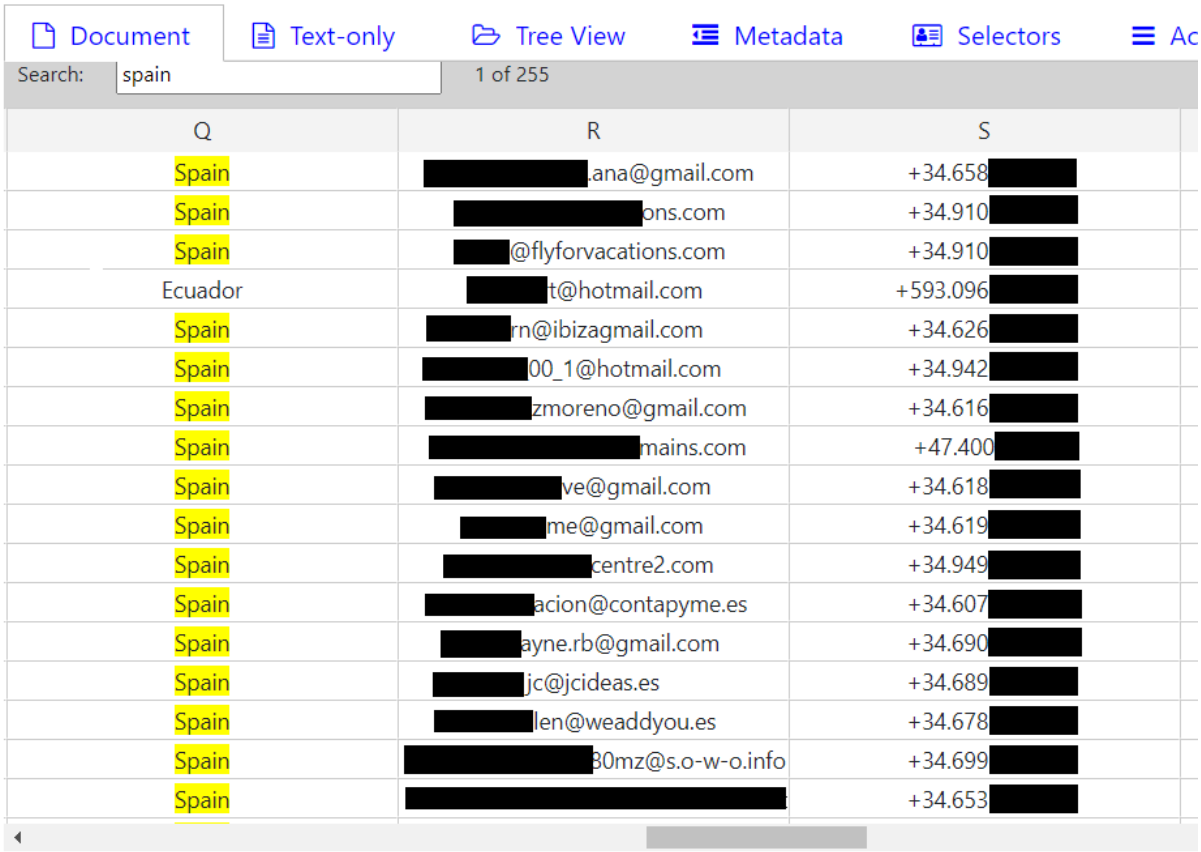
Dentro de estos encontraremos la información asociada a la búsqueda realizada, pudiendo filtrar y descargar los datos tanto en formato tabla como en RAW. En este caso, podemos observar que encontramos nombres de personas, su dirección, provincia código postal, correo y número de teléfono.


Document | Text-only | Tree View | Metadata | Selectors | Actions

Search: spain — 1 of 255

L	M	N	O	P	Q
Ana [redacted]	C/ Periodista Eduardo Chinarro	Sevilla	ES-SE	41019	Spain
[redacted]	Calle Cartagena 80	Madrid	M	28028	Spain
[redacted]	Calle Cartagena 80	Madrid	M	28028	Spain
DARWIN [redacted]	la joya cdla onix mz 1 villa 30	Quito	PICHINCHA	111111	Ecuador
[redacted]	c/ Miguel Caieta Soler, n8	EIVISSA	ILLES BALEARS	7800	Spain
[redacted]	CONCHA ESPINA, 9, 17B	COLINDRES	CANTABRIA	39750	Spain
[redacted]	Delgado Serrano	Ceuta	CE	51001	Spain
[redacted]	P.O. Box 240	Arguineguin	GC	35120	Spain
[redacted]	Eve Online Universe	Castelldefels	ES-CT	8860	Spain
[redacted]	Avinguda Diagonal 158 4o 2a	Barcelona	Barcelona	8018	Spain
[redacted]	CONSTITUCION, 16 BAJO(LOCAL)	GUADALAJARA	GUADALAJARA	19003	Spain
[redacted]	Calle San Bernardo 117	Madrid	M	28015	Spain
sofia [redacted]	calle progreso 43	Sant Antoni de Portmany	Illes Balears	7820	Spain
Juan [redacted]	Ctra Nacional 340	Benicasim	CS	12560	Spain
[redacted]	Calle Gobernador Viejo, 29	Valencia	Valencia	46003	Spain
[redacted]	Fernan Caballero, 8	Mairena del Alcor		41510	Spain
[redacted]	Juan Saraza Ortiz	Las Palmas de Gran Canaria		35014	Spain

Ilustración 47. Tabla con nombres y direcciones.


Document | Text-only | Tree View | Metadata | Selectors | Ac

Search: spain — 1 of 255

Q	R	S
Spain	[redacted]ana@gmail.com	+34.658[redacted]
Spain	[redacted]ons.com	+34.910[redacted]
Spain	[redacted]@flyforvacations.com	+34.910[redacted]
Ecuador	[redacted]t@hotmail.com	+593.096[redacted]
Spain	[redacted]rn@ibizagmail.com	+34.626[redacted]
Spain	[redacted]00_1@hotmail.com	+34.942[redacted]
Spain	[redacted]zmoreno@gmail.com	+34.616[redacted]
Spain	[redacted]mains.com	+47.400[redacted]
Spain	[redacted]ve@gmail.com	+34.618[redacted]
Spain	[redacted]me@gmail.com	+34.619[redacted]
Spain	[redacted]centre2.com	+34.949[redacted]
Spain	[redacted]acion@contapyme.es	+34.607[redacted]
Spain	[redacted]ayne.rb@gmail.com	+34.690[redacted]
Spain	[redacted]jc@jcideas.es	+34.689[redacted]
Spain	[redacted]len@weaddyou.es	+34.678[redacted]
Spain	[redacted]30mz@s.o-w-o.info	+34.699[redacted]
Spain	[redacted]	+34.653[redacted]

Ilustración 48. Continuación de la tabla: emails y teléfonos.

2.4.8 Maltego

Maltego está categorizada como una de las mejores herramientas para la inteligencia de fuentes abiertas. Su potencial se basa en la gran cantidad de fuentes abiertas a las que realiza consultas, denominadas transformadores, para obtener diferentes tipos de datos. Además su interfaz permite crear una estructura ordenada y relacional en forma de grafo, capaz de generar informes completos de los datos

obtenidos para un posterior análisis, comparativa o incluso en un proceso de hacking ético.

Al iniciar la herramienta se muestran distintos transformadores e información detallada sobre estos y que pueden llegar a obtener. Maltego ya incluye instalados algunos transformadores, pero brinda la posibilidad de añadir más.

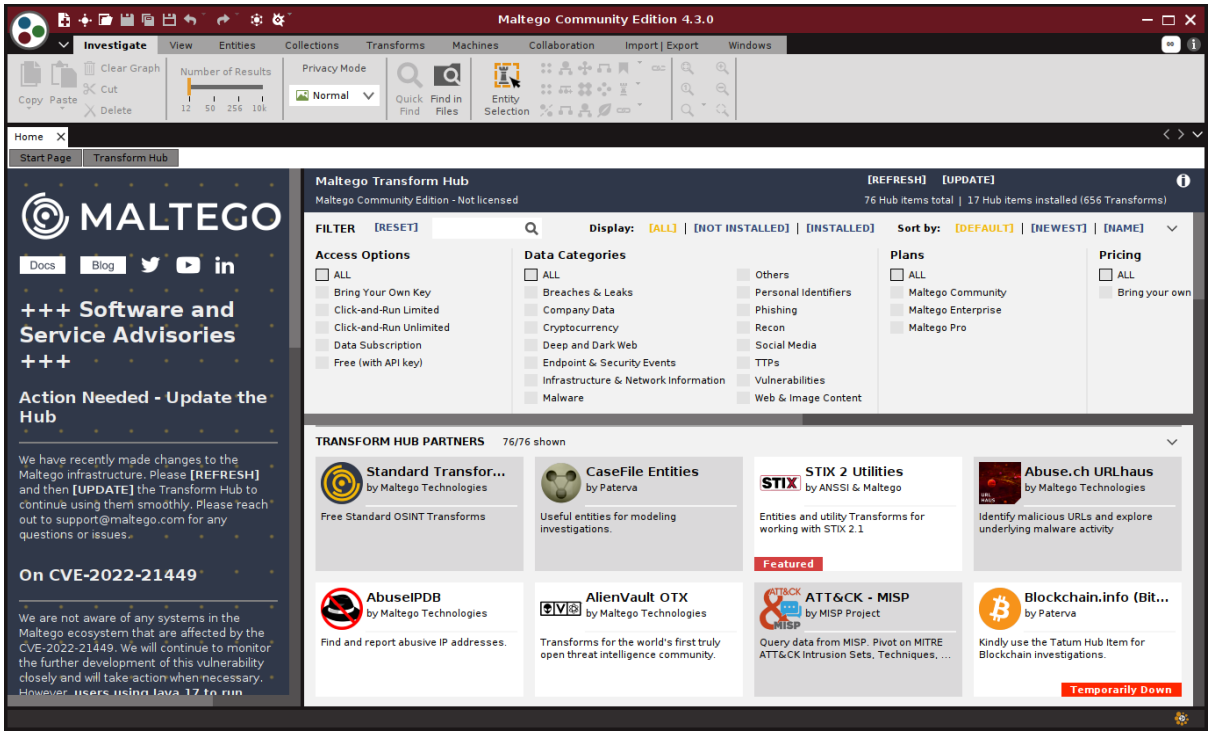


Ilustración 49. Maltego

Para empezar a trabajar deberemos crear un nuevo grafo, en esta nueva ventana aparecerá un lienzo en blanco, donde se podrán añadir entidades mediante una paleta en la cual existe distintas categorías de entidades sobre las que ejecutar los transformadores.

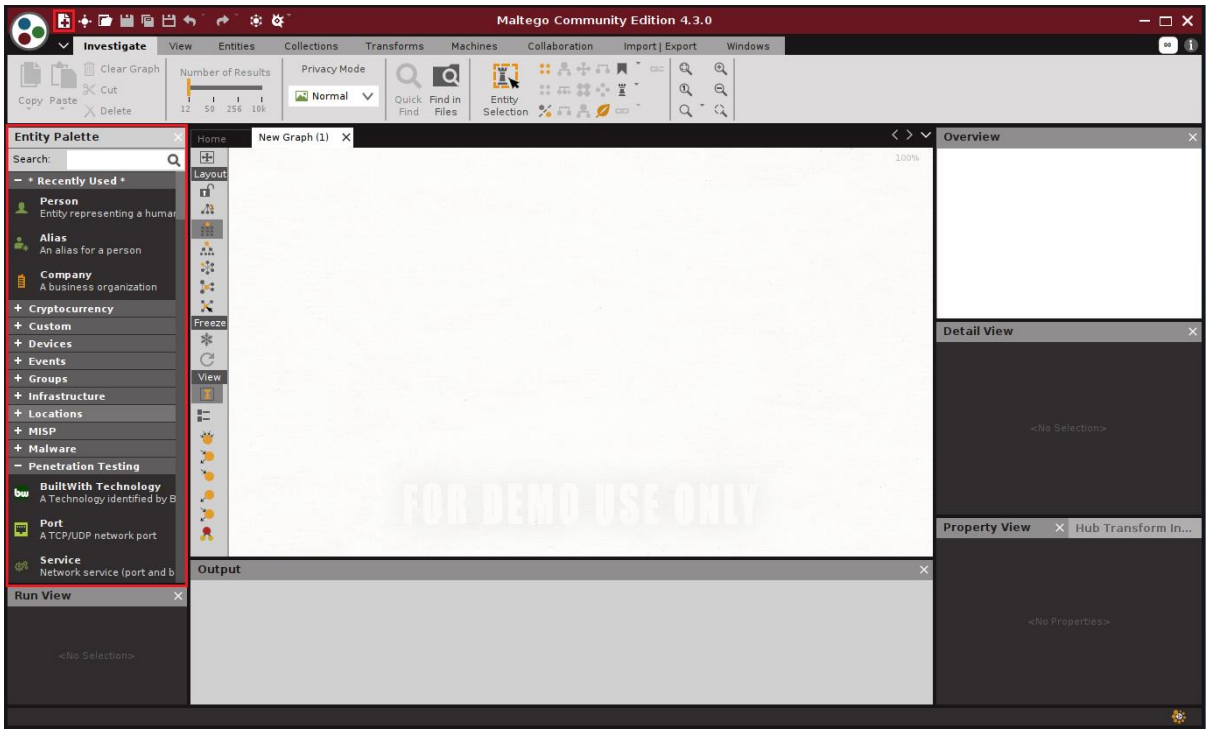


Ilustración 50. Ventana de trabajo

A partir de información que hayamos obtenido mediante ingeniería social u otras herramientas se podrá generar un grafo con esta información.

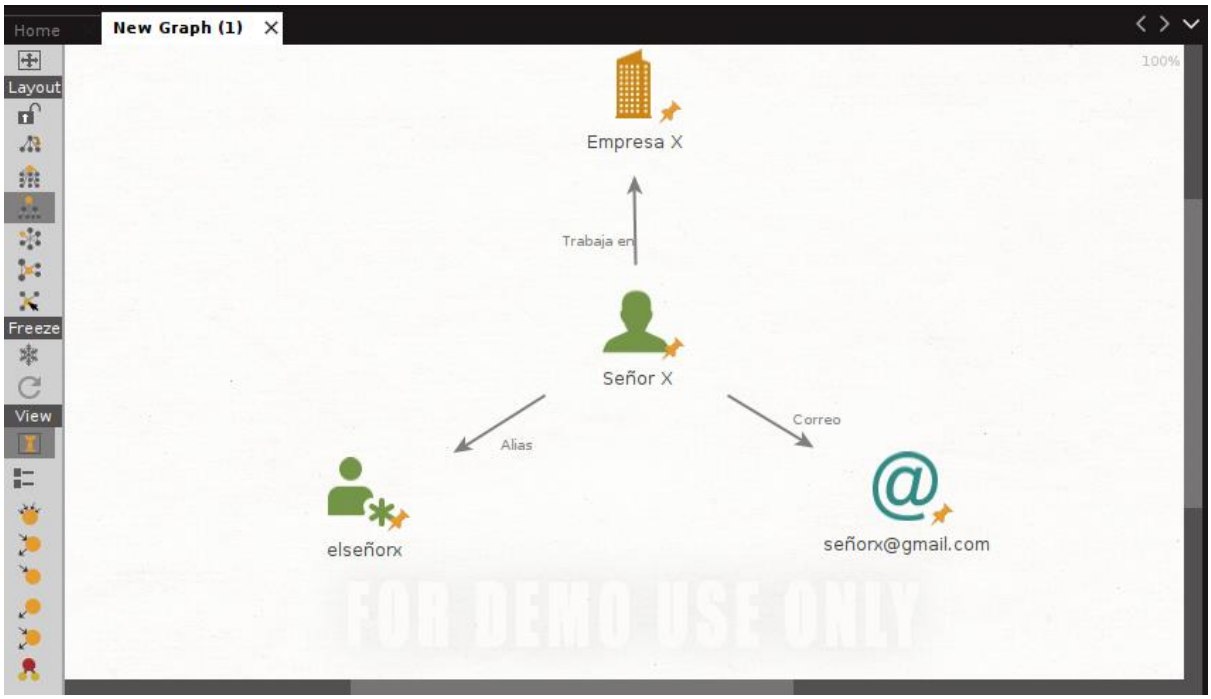


Ilustración 51. Ejemplo de grafo.

Marcando todos los elementos que componen el lienzo y haciendo click derecho veremos las transformaciones que se pueden aplicar.

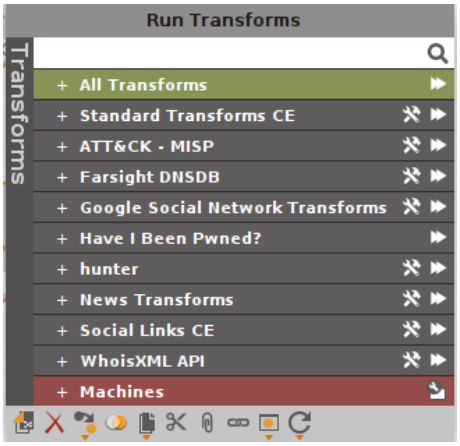


Ilustración 52. Lista de transformaciones.

Como resultado de ejecutar todas las trasformaciones obtenemos un grafo aun mayor con todos los elementos asociados a cada entidad y una leyenda de los datos obtenidos.

Ilustración 53. Resultado de las transformaciones.

Ampliando sobre cada entidad se podrá ver mejor la información encontrada y pulsando sobre esta obtendremos una ventana con más detalles sobre ese elemento.

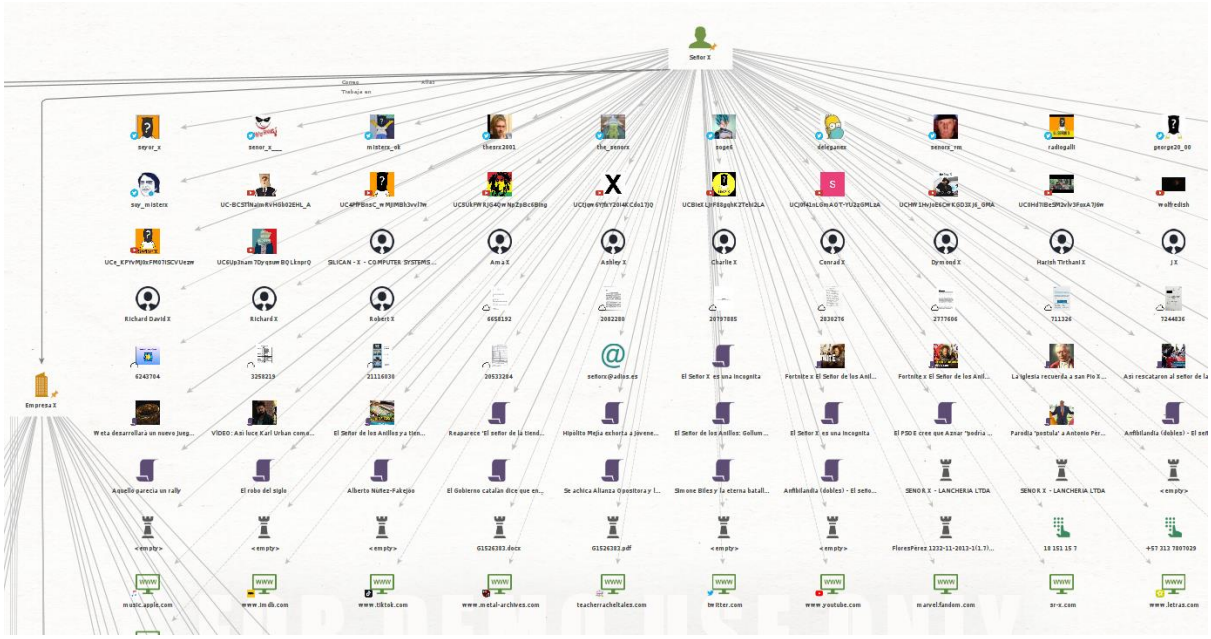


Ilustración 54. Datos obtenidos.

Finalmente, en el menú será posible generar un informe completo de toda la información obtenida.

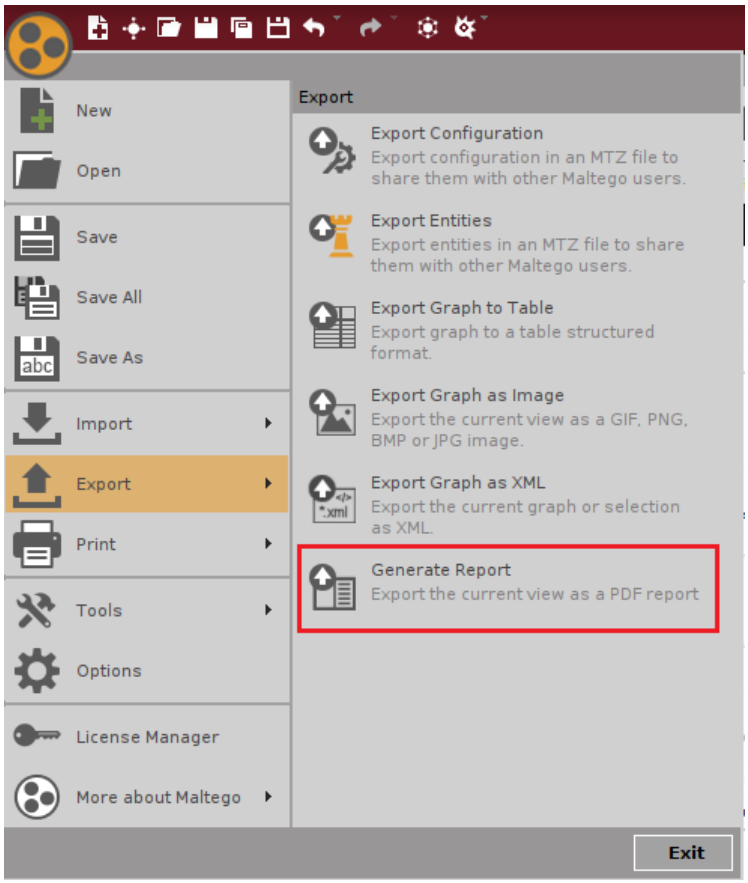


Ilustración 55. Generación de informe detallado.

2.5 Herramientas en Redes Sociales

Actualmente, el uso de las redes sociales sigue en su apogeo y la mayoría de los usuarios de internet tiene conocimiento sobre su existencia o forma parte de alguna de estas. Realmente ya no existe un rango de edad para su uso, desde los más jóvenes hasta los más mayores, tienen la capacidad de acceder a ellas.

Ni siquiera es necesario limitarse a usar solo una red social, porque el tipo de contenido no tiene que ser el mismo en todas, ni el uso que se hace de cada una de ellas. En ellas se publica información personal a la cual es posible acceder, hacer un seguimiento y analizar, debido a que gran parte de la información que contienen es pública.

2.5.1 Nombres de usuario

Conociendo un nombre de usuario, es posible obtener con mucha facilidad en qué redes sociales está registrado este nombre. Existe una gran cantidad de herramientas que comprueban la existencia de estos usuarios realizando diferentes consultas para obtener una verificación de su registro.

Una herramienta web bastante útil es Namecheckr la cual muestra los sitios donde no está disponible un nombre de usuario, indicando la existencia de estos y dominios que también han sido registrados con ese nombre.

Ilustración 56. Namecheckr.

De igual forma también tenemos herramientas por línea de comandos, un ejemplo es Sherlock, la cual busca cuentas de usuario en redes sociales y devuelve como resultado un enlace a dicha cuenta.

```
┌──(kali㉿kali)-[~/sherlock]
└─$ python3.10 sherlock universidadjaen
[*] Checking username universidadjaen on:

[+] Blogger: https://universidadjaen.blogspot.com
[+] Facebook: https://www.facebook.com/universidadjaen
[+] Giphy: https://giphy.com/universidadjaen
[+] GitHub Support Community: https://github.community/u/universidadjaen/summary
[+] GuruShots: https://gurushots.com/universidadjaen/photos
[+] Instagram: https://www.instagram.com/universidadjaen
[+] Letterboxd: https://letterboxd.com/universidadjaen
[+] Linktree: https://linktr.ee/universidadjaen
[+] Star Citizen: https://robertsspaceindustries.com/citizens/universidadjaen
[+] Whonix Forum: https://forums.whonix.org/u/universidadjaen

[*] Results: 10

[!] End:  The processing has been finished.
```

Ilustración 57. Sherlock.

2.5.2 Osintgram

Una herramienta de código abierto muy potente para obtener una gran variedad de información de un perfil dentro de Instagram. Con esta es posible analizar un perfil, obteniendo información asociado al ID único de cada usuario, otros perfiles con los que interactúa, descargar fotos, historias, etc.

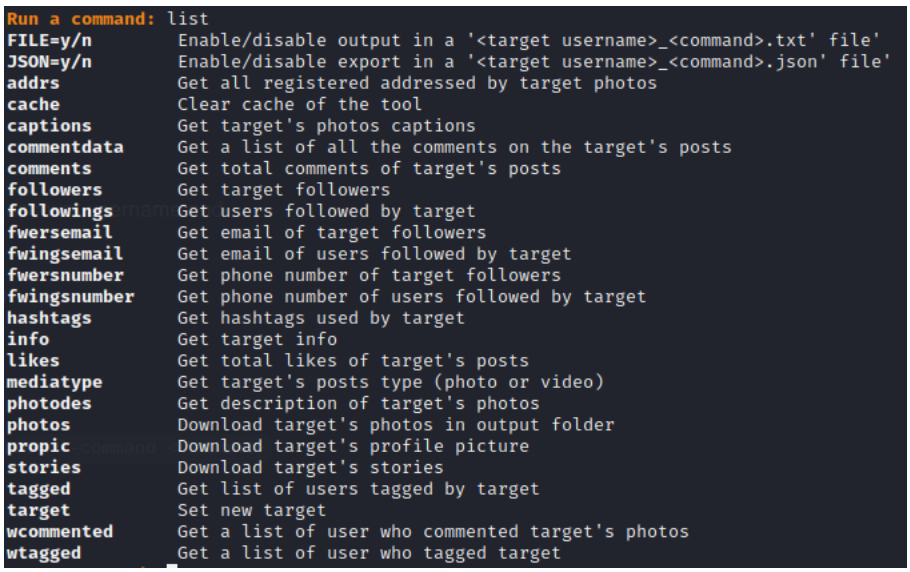


Ilustración 58. Comandos Osintgram.

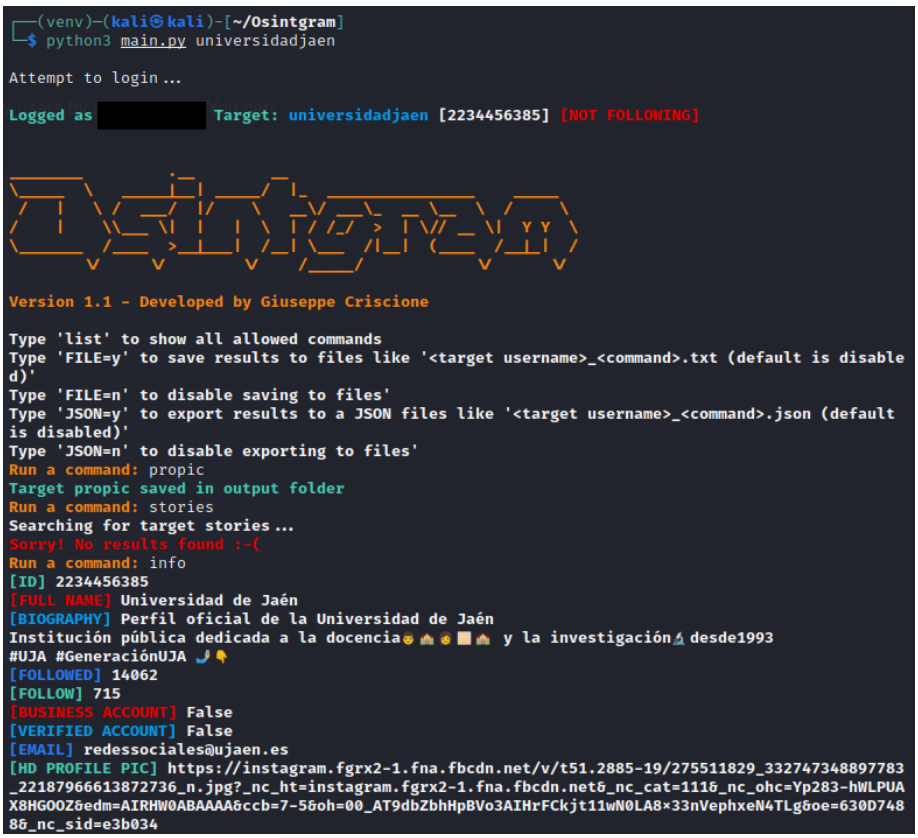


Ilustración 59. Obtención de datos.

2.5.3 Socialbearig

Esta herramienta web permite el análisis de perfiles públicos dentro de la red de Twitter, ya que realiza un escaneo de un perfil específico para obtener estadísticas de la actividad, tweets, retweets, likes, hashtag, etc.

En la siguiente imagen se puede ver como para el perfil “ujaen”, se obtiene en la parte izquierda información general, como su foto de perfil, descripción, fecha de creación de la cuenta, localización, número de tweets, último tweet, entre otras, mientras que en la parte central se obtienen los resultados más relevantes del análisis.

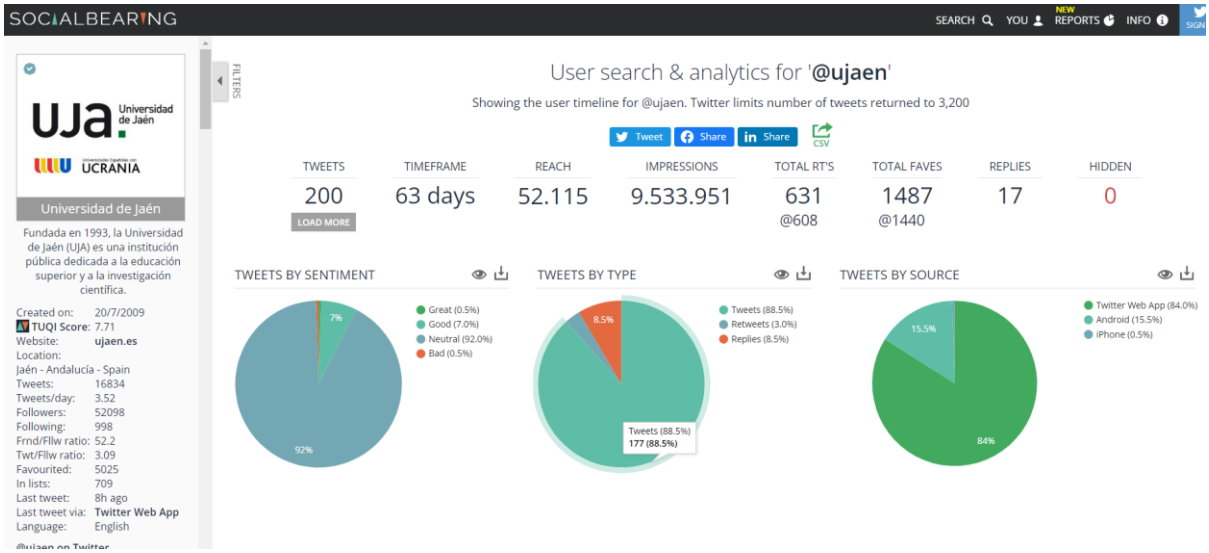


Ilustración 60. Resultados Socialbearing.

Se muestran métricas de la cantidad de tweets publicados en los últimos días y los hashtags más usados.

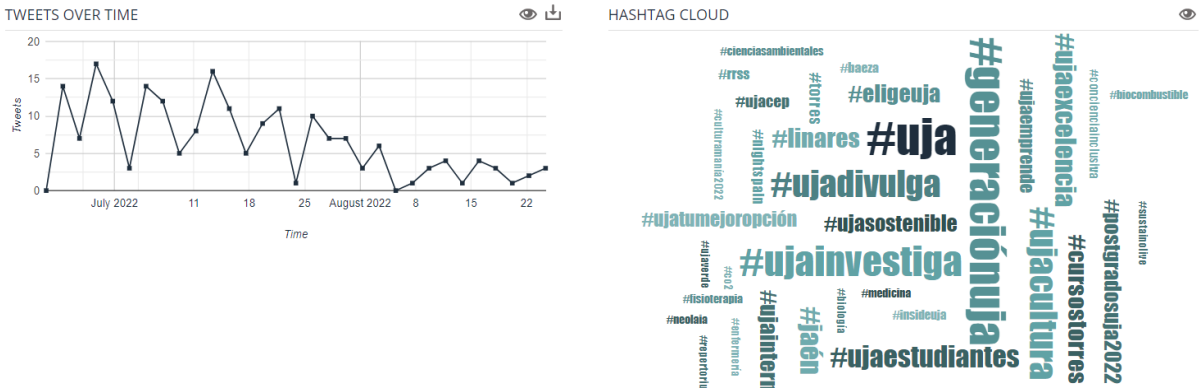


Ilustración 61. Tweets por día y Hashtags.

También se incluye sin necesidad de entrar a la red social los tweets, retweets y likes, mostrando así las publicaciones e interacciones del perfil.

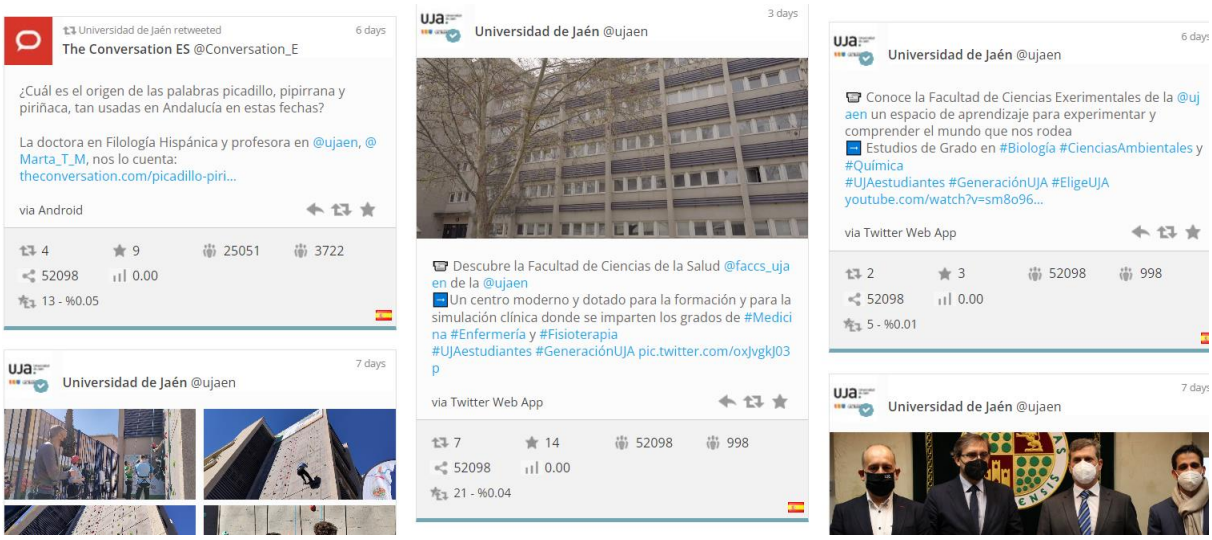


Ilustración 62. Interacciones y publicaciones.

CAPÍTULO 3: Ingeniería del Software

La idea para desarrollar este prototipo, es que a partir de un nombre se pueda relacionar con nombres usuario que tengan sentido y buscar estos dentro de distintas redes sociales. Nos centraremos principalmente en las redes sociales puesto que actualmente son la mejor forma de obtener información sobre una persona. Este sería un punto de partida en el proceso de obtener información sobre un usuario, puesto que este es un proceso complejo que implica bastante tiempo de investigación.

3.1 Proceso de ingeniería del software

La Ingeniería del Software consiste en el conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos. Este proceso es fundamental para poder llegar a obtener un producto de calidad, conforme a unos plazos y costes establecidos.

Para el proyecto se ha determinado que el mejor modelo a seguir es el modelo en cascada, que se caracteriza por dividir las distintas etapas del desarrollo en sucesivas fases. Este modelo consta de cinco fases:

1. **Análisis**: en esta fase se definen, examinan y especifican todos los requisitos que debe cumplir el software a desarrollar. Este apartado es fundamental para el desarrollo de las siguientes fases.
2. **Diseño**: se establece la arquitectura del software, estructura de datos, las posibles representaciones de la interfaz y se detallan los pasos a seguir funcionalmente.
3. **Implementación**: se plasma la fase de diseño en el código, desarrollando así el código fuente del software.
4. **Pruebas**: integración de sistemas, verificación del mismo y detección de errores.
5. **Mantenimiento**: tras la puesta en producción se debe realizar un seguimiento del producto final, puesto que pueden surgir errores no detectados anteriormente o eventualmente se deberán realizar cambios para mejorarlo.

3.2 Análisis

Dentro del proceso de análisis es fundamental que, a través de distintos requerimientos funcionales y no funcionales, se llegue a comprender que se quiere conseguir con el software a desarrollar, como su comportamiento, funciones requeridas, conexiones, etc.

3.2.1 Requerimientos funcionales

Los requerimientos funcionales permiten declarar los servicios que va a prestar el sistema y la forma en que se reaccionara a estos, así como el comportamiento básico del mismo.

- **A partir de un nombre obtener distintos nombres de usuario**: consiste en realizar una búsqueda para obtener distintos nombres de usuario que tengan relación con el nombre introducido.
- **Selección de nombre de usuario**: tras obtener distintos nombres de usuario se podrá seleccionar aquellos que se consideren más interesantes.
- **Búsqueda en profundidad**: consiste en obtener distinta información del o los nombres de usuario seleccionados.
- **Mostrar los datos obtenidos**: se mostrarán todos los datos que sean posibles relacionar con ese usuario o por el contrario la negación de estos. Estos datos incluirán, si el nombre de usuario existe en una red social, foto de perfil, si existen correos electrónicos con ese mismo nombre de usuario que hayan sido filtrados y si es posible la información filtrada.
- **Referencia a enlaces**: consiste en referenciar posibles enlaces a datos, por ejemplo, si existe un usuario en una red social o se ha podido obtener su foto de perfil, se podrá redireccionar al recurso.
- **Atajo de búsqueda**: consiste en introducir un atajo para realizar una búsqueda profunda sobre un nombre de usuario directamente.
- **Registro de datos**: consiste en guardar un registro de los datos obtenidos en cada búsqueda en profundidad realizada y que posteriormente podrán ser visualizados en cualquier momento.

3.2.2 Requerimientos no funcionales

Se tratan de requisitos que no se refieren directamente a las funciones específicas suministradas por el sistema, sino a las propiedades de este.

- **Tiempos de respuesta de las peticiones**: puesto que se van a realizar distintas consultas y algunas de estas pueden tardar varios segundos, será importante la optimización de las consultas para evitar la redundancia y aumento del tiempo.
- **Diseño de interfaz**: se diseñará una interfaz fácil de entender y que se adapte a toda la posible información que va a ser mostrada.

3.3 Diagramas de Casos de Uso

Un caso de uso es una parte de la ingeniería del software que permite definir una secuencia de acciones que tienen interacción con el usuario y el sistema, llegando a obtener un resultado observable. En este proceso se identifican las funcionalidades y se realiza tanto de forma gráfica como textual una especificación de cada caso de uso, permitiendo así obtener una mejor definición de estos.

Dentro de los diagramas se pueden incluir varios casos de uso y las relaciones entre casos de uso y las personas, los grupos o los sistemas que interactúan para llevarlo a cabo. Se denomina actor externo al sistema que guarda una relación con este y que le demanda una funcionalidad. Un artefacto de actor puede utilizarse en varios diagramas de caso de uso.

Para comenzar se realiza un diagrama frontera, cuyo objetivo principal es identificar tanto las fronteras de nuestro sistema como las interacciones que este tiene con sus alrededores.

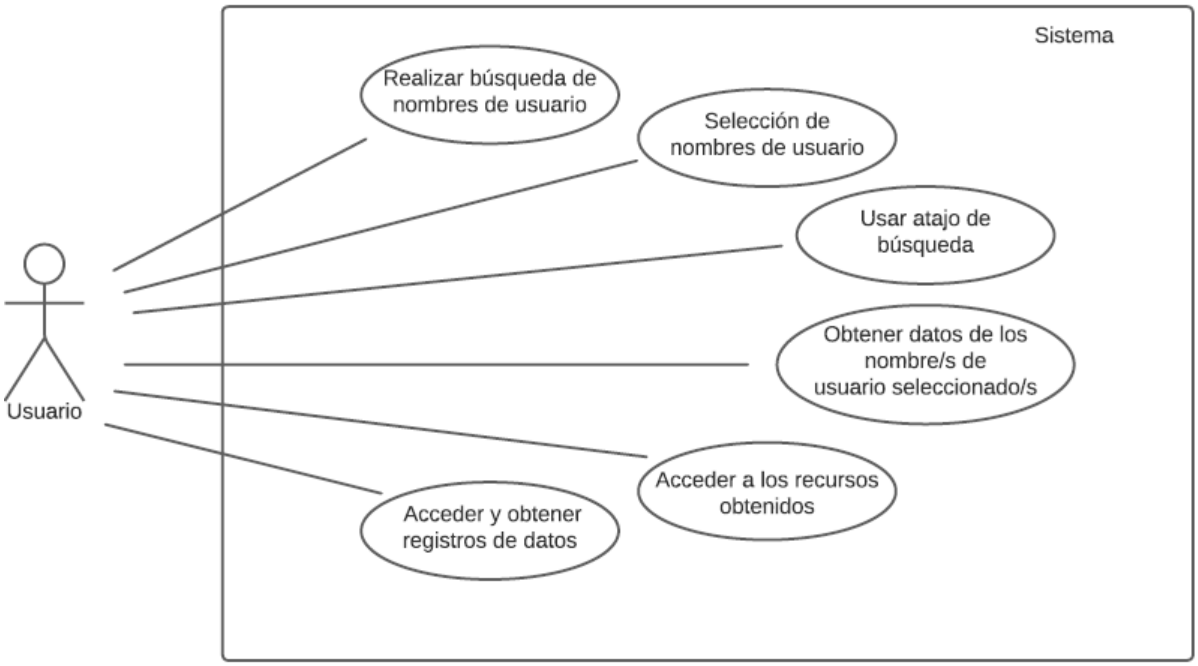


Ilustración 63. Diagrama frontera

La anterior imagen muestra una visión global de los distintos casos de uso, por lo tanto, vamos a pasar a definir con un mayor nivel de detalle cada uno de estos.

Caso 1: Realizar búsqueda de nombres de usuario

- Actores principales: Usuario.
- Condiciones de entrada: Ninguna.
- Flujo de eventos:
    1. El usuario introduce un nombre o palabras clave para encontrar nombres de usuario.
- Condiciones de salida:
    1. Búsqueda exitosa: Lista de nombre de usuario.
    2. Búsqueda fallida: Mensaje de error, no se encuentran nombres de usuario que guarden relación.
- Excepción: Si la búsqueda no puede llegar a completarse se mostrará un mensaje de error indicando que no ha sido posible realizarla.

Caso 2: Selección de nombres de usuario

- Actores principales: Usuario.
- Condiciones de entrada: Tener una lista de nombres de usuario.
- Flujo de eventos:
    1. Marcar los nombres de usuario para buscar información sobre estos con un mayor nivel de detalle.

2. Pulsar el botón de confirmación para comenzar la búsqueda de datos.

- Condiciones de salida:
  1. Muestra de datos exitosa: se podrán visualizar los datos obtenidos.
  2. Muestra de datos fallida: será el peor de los casos, puesto que no habrá sido posible encontrar ningún dato que tenga relación.
- Excepción: Se devuelve un error asociado a los nombres de usuario que se han seleccionado.

Caso 3: Usar atajo de búsqueda
- Actores principales: Usuario.
- Condiciones de entrada: Ninguna.
- Flujo de eventos:
  1. Introducir en la barra de búsqueda “@nombre_usuario”.
- Condiciones de salida:
  1. Muestra de datos exitosa: se podrán visualizar los datos obtenidos del usuario introducido directamente.
  2. Muestra de datos fallida: será el peor de los casos, puesto que no habrá sido posible encontrar ningún dato que tenga relación.
- Excepción: Si no se introduce bien el atajo se mostrará un error.

Caso 4: Obtener datos del nombre/s de usuario seleccionado/s
- Actores principales: Sistema.
- Condiciones de entrada: Lista de nombre de usuario o haber usado el atajo.
- Flujo de eventos:
  1. Devolver datos.
- Condiciones de salida:
  1. Se muestra una vista con todos los datos que hayan sido posible obtener.

- Excepción: Puesto que para obtener los datos se realizarán llamadas a distintas fuentes de información se deberá resolver las excepciones que se produzcan dentro de cada llamada, es decir, siempre se mostrará una vista con toda la información que haya sido posible recaudar, incluyendo indicativos de fallos en las llamadas si se llegasen a producir.

Caso 5: Acceder a los recursos obtenidos
- Contexto: Por recursos se refiere a los elementos que son seleccionables dentro de la vista y redireccionar al sitio donde se encuentra almacenada la información.
- Actores principales: Usuario.
- Condiciones de entrada: Vista con los datos obtenidos.
- Flujo de eventos:
  1. Seleccionar aquellos campos en los que existe una redirección al recurso.

- Condiciones de salida:
    1. Se abre una nueva pestaña con el recurso.

- Excepción: No se podrá acceder al recurso, puesto que el servidor donde esté alojado estará caído (No será lo común).

Caso 6: Acceder y obtener registros de datos

- Contexto: Un registro de datos se refiere a toda la información que se ha obtenido en una búsqueda, puesto que esta se guarda cada vez que se completa una búsqueda con éxito.
- Actores principales: Usuario.
- Condiciones de entrada: Mínimo haber realizado una búsqueda sobre un nombre de usuario
- Flujo de eventos:
    1. Solicitar lista de registro de datos.
    2. Seleccionar un registro
- Condiciones de salida:
    1. Se abre una nueva pestaña con el registro.

- Excepción: Si no se puede crear un registro de datos, se mostrará un mensaje por consola y este no aparecerá en la lista de registro, no pudiendo ser seleccionado.

3.4 Diseño

Se establecen necesidades funcionales que aportarán robustez a las fases posteriores. Se descomponen elementos más grandes en otros más simples para una mejor definición del funcionamiento de estos que permite obtener un arquitectura y estructura con un mayor nivel de detalle.

3.4.1 Diagrama de clases

Un diagrama de clases muestra las relaciones y la estructura de las clases que componen el sistema.

Partiremos de un diagrama sencillo donde tendremos la clase principal, la cual solicitará a las demás clases los datos que obtengan cuando estas realicen consultas a sus respectivos métodos.

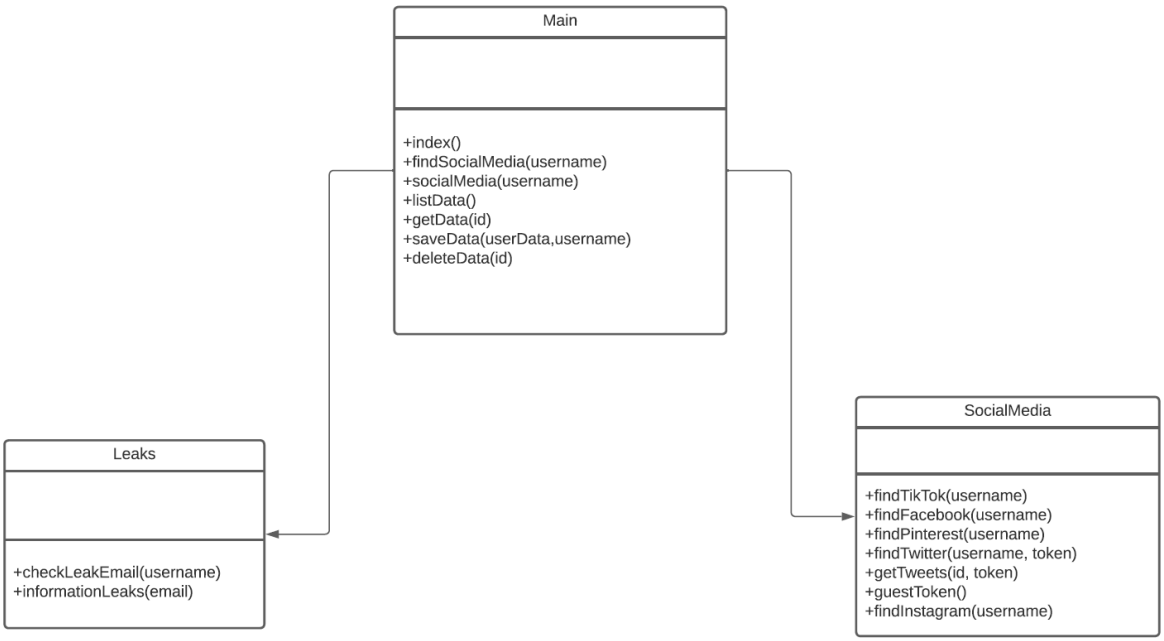


Ilustración 64. Diagrama UML

Este diagrama deberá evolucionar en fases posteriores a su aceptación como prototipo. Un ejemplo de esto sería la clase “SocialMedia” donde se realizan diferentes consultas a redes sociales, en el caso de querer obtener más información o añadir un mayor procesamiento, el cual implique la creación de más funciones, sería recomendable separar cada red social y crear una clase específica para esta. Llegando a obtener una estructura más compleja con la implantación de nuevos módulos que aporten más funcionalidad a la aplicación.

3.4.2 Diagramas de secuencia

Los diagramas de secuencia se centran en los procesos, objetos que coexisten simultáneamente, los mensajes intercambiados entre ellos para ejecutar una función antes de que la línea de vida termine. Describiendo así cómo y en qué orden un grupo de objetos y clases funcionan en conjunto.

*Anotación: Para el diagrama de secuencia 2, hay que tener en cuenta que “View” es la vista HTML que se le muestra al usuario (donde se tendrá la barra de búsqueda y los datos obtenidos) por tanto siempre estará entre el usuario y el sistema.

En los demás diagramas de secuencia será obviado puesto que no se requiere una gran interacción con esta, pero en ese caso es importante debido a las interacciones que se realizan.

- **Diagrama de secuencia 1**: búsqueda de nombres de usuario.

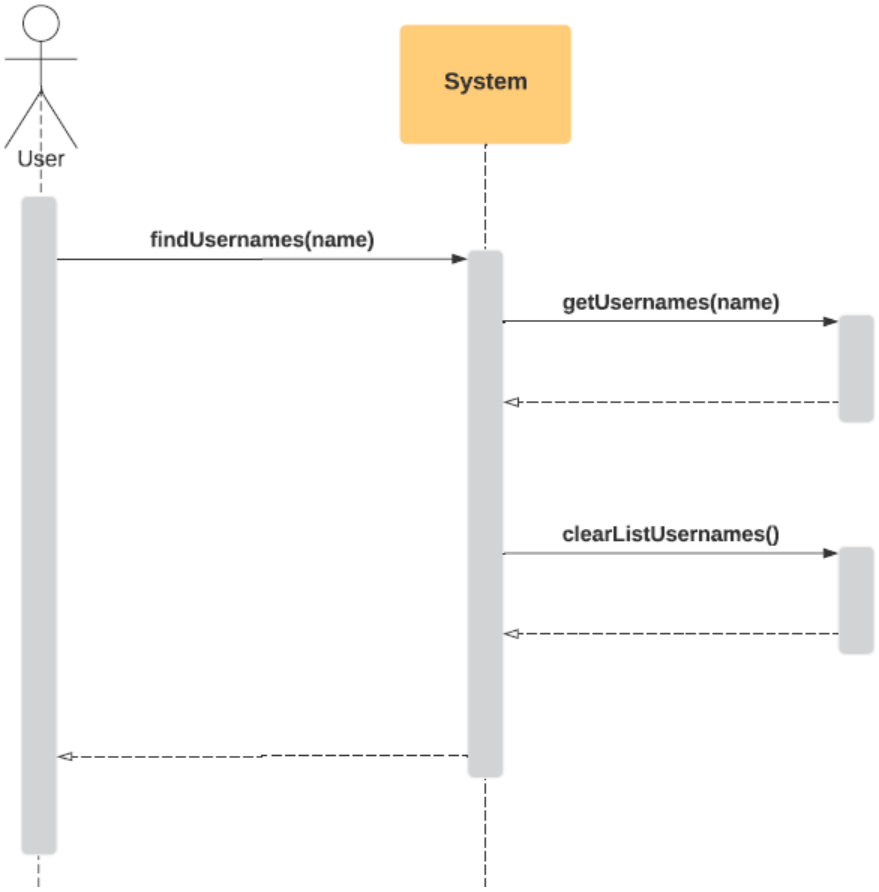


Ilustración 65. Diagrama de secuencia 1

- **Diagrama de secuencia 2**: selección de nombres de usuario.

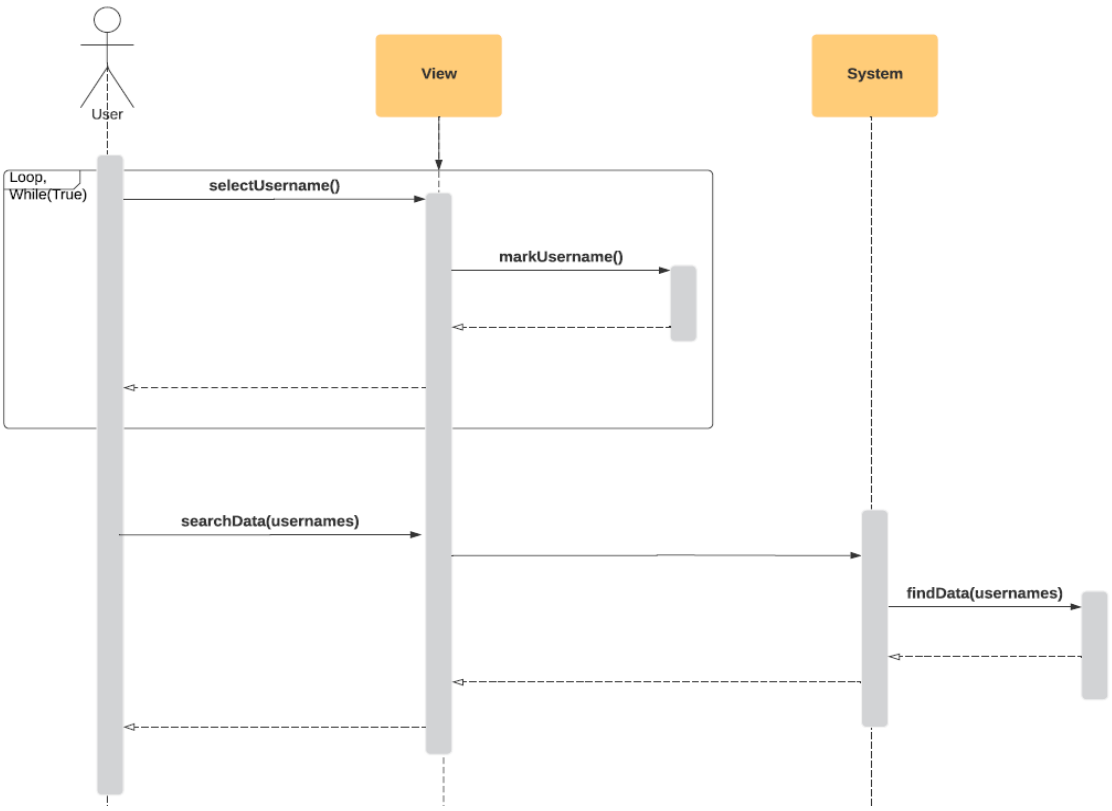


Ilustración 66. Diagrama de secuencia 2

- **Diagrama de secuencia 3**: atajo de búsqueda.

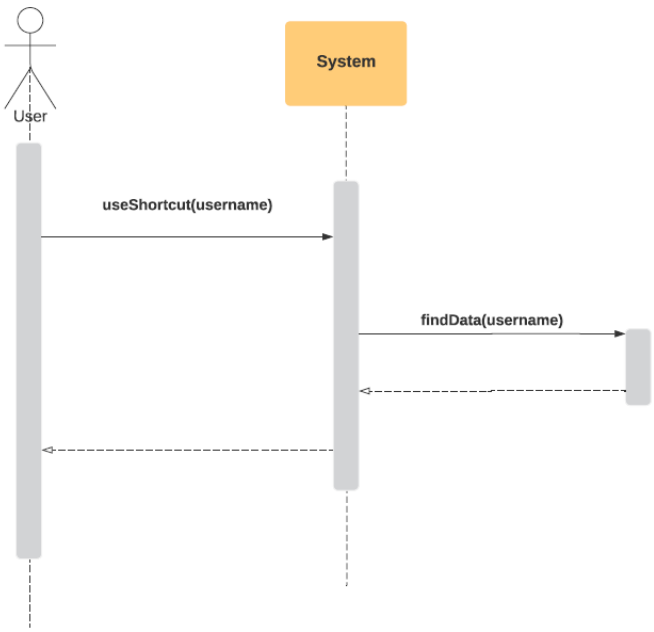


Ilustración 67. Diagrama de secuencia 3

- **Diagrama de secuencia 4**: obtener datos del nombre/s de usuario seleccionado/s.

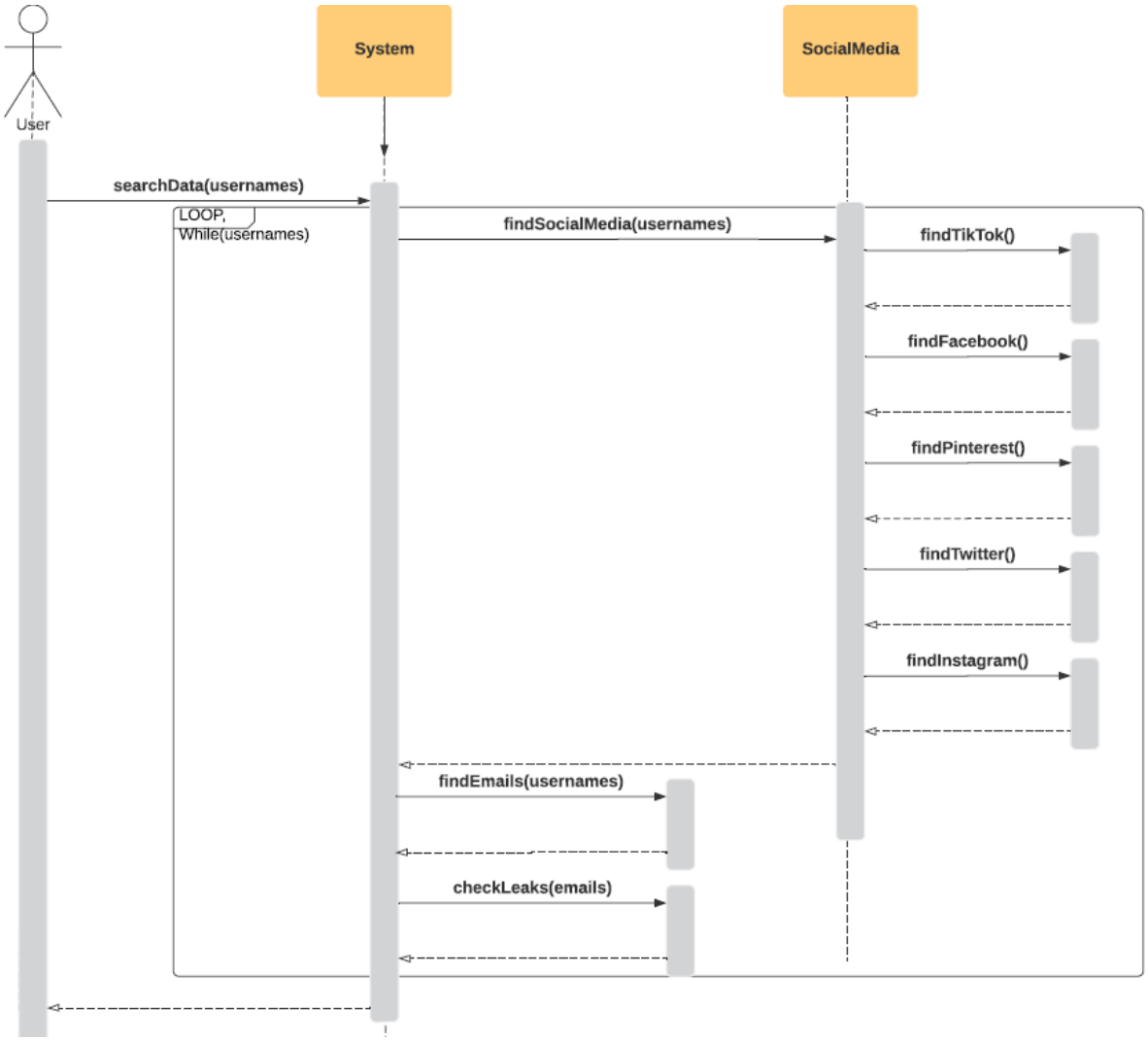


Ilustración 68. Diagrama de secuencia 4

- **Diagrama de secuencia 5**: acceder a los recursos obtenidos.

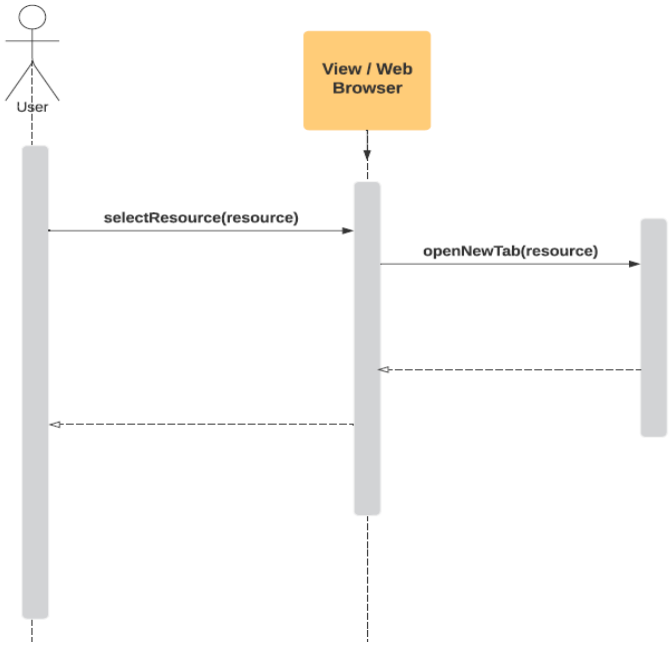


Ilustración 69. Diagrama de secuencia 5

- **Diagrama de secuencia 6**: acceder y obtener registros de datos.

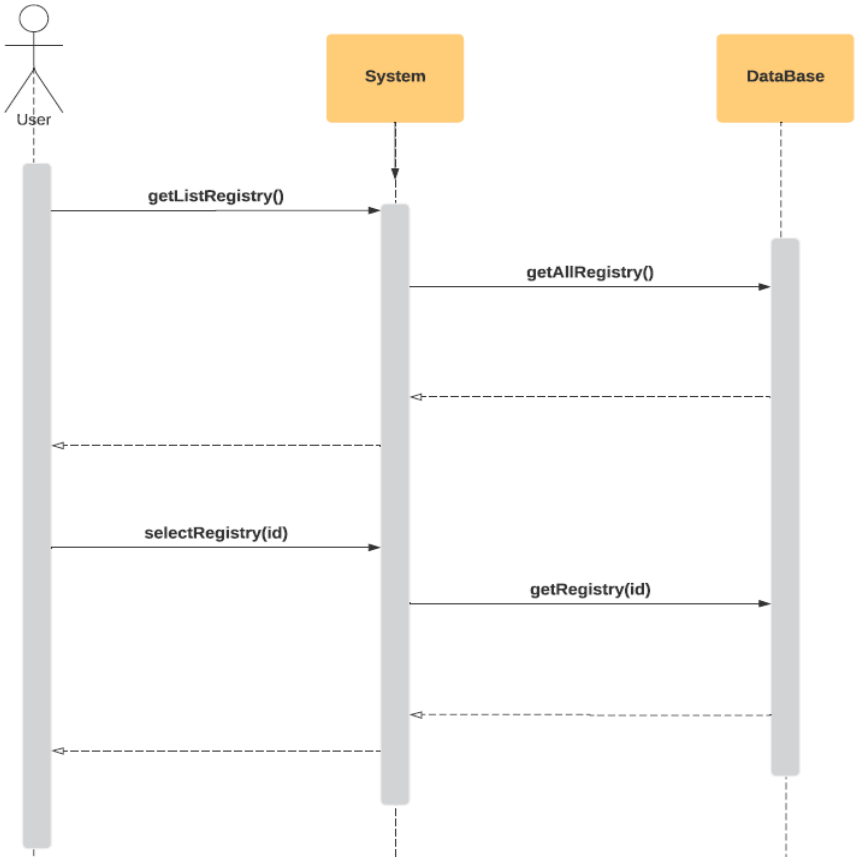


Ilustración 70. Diagrama de secuencia 6

3.5 Implementación

La implementación es la tarea donde se elaboran, adaptan y añaden los elementos anteriormente definidos. Se deberá seleccionar un lenguaje en el cual programar el código fuente y un entorno donde pueda ser desarrollado.

3.5.1 Lenguaje de programación

Para seleccionar el lenguaje de programación deberemos tener en cuenta las necesidades de nuestro proyecto. En este caso el lenguaje de programación que más funcionalidades y utilidades podrá aportar es Python en su versión 3.8.10 (aunque también se emplea JavaScript y HTML). Como se mostrarán imágenes, vídeos y almacenaremos información dentro de una base de datos, utilizaremos el framework de desarrollo Djando en su versión 3.2, el cual utiliza el MVC (modelo–vista–controlador).

De esta forma, aunque se esté desarrollando un prototipo, se podrá brindar una mejor experiencia al usuario y una visión de lo que puede llegar a ser la aplicación a un nivel empresarial.

3.5.2 Herramienta de desarrollo

Como herramienta para el desarrollo del prototipo, se elige Visual Studio Code la cual es un editor de código fuente que incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, refactorización de código, extensión de lenguajes de programación como Python, JavaScript, HTML, entre otras funcionalidades.

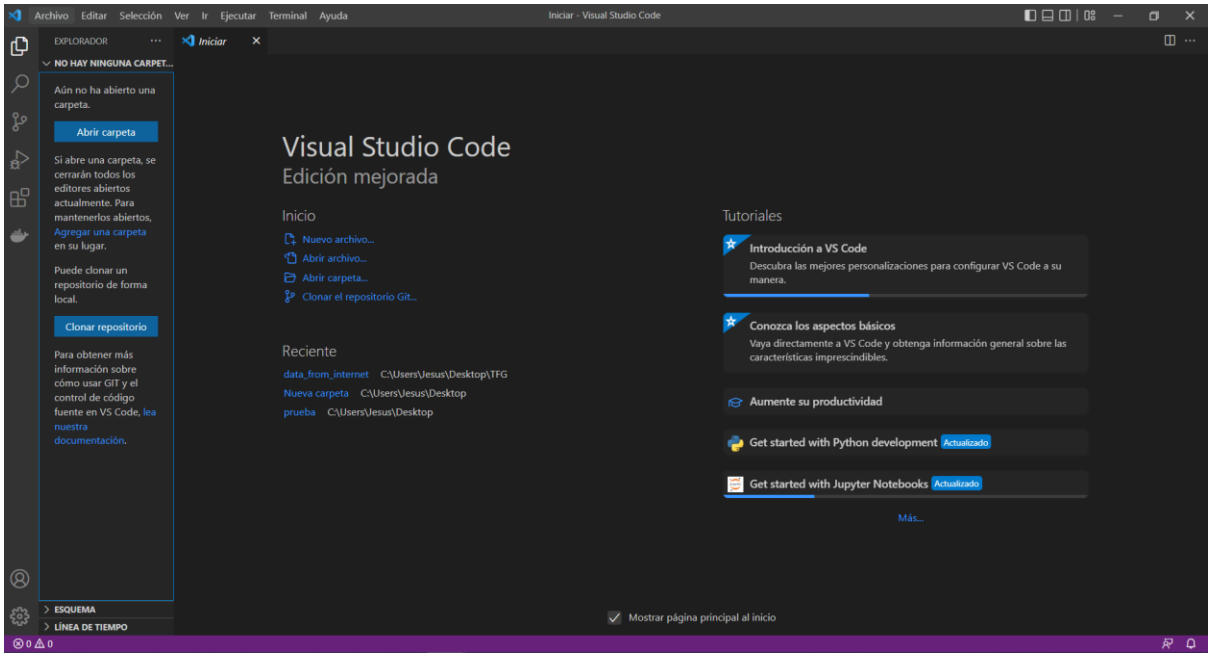


Ilustración 71. Visual Studio Code

3.5.3 Implementación del código fuente

Primero veremos la estructura de carpetas del proyecto, donde nos centraremos en “information” que es la aplicación web. Destacaremos:

1. **static**: carpeta donde se encuentran los archivos estáticos como la hoja de estilos y las descargas tanto de imágenes como de vídeos que estén asociados a un nombre de usuario y a una entrada en la base de datos.
2. **templates**: estas vistas HTML serán la interfaz de usuario por la cual interactúa con el sistema.
3. **models**: definiremos el modelo de información que se guarda en la base de datos.
4. **constats**: fichero de las variables de entorno.
5. **views**: fichero principal donde se realiza la ejecución del código.
6. **socialMedia**: fichero donde se encuentran las peticiones a distintas redes sociales y obtención de información.
7. **leaks**: fichero para la comprobación de filtraciones.

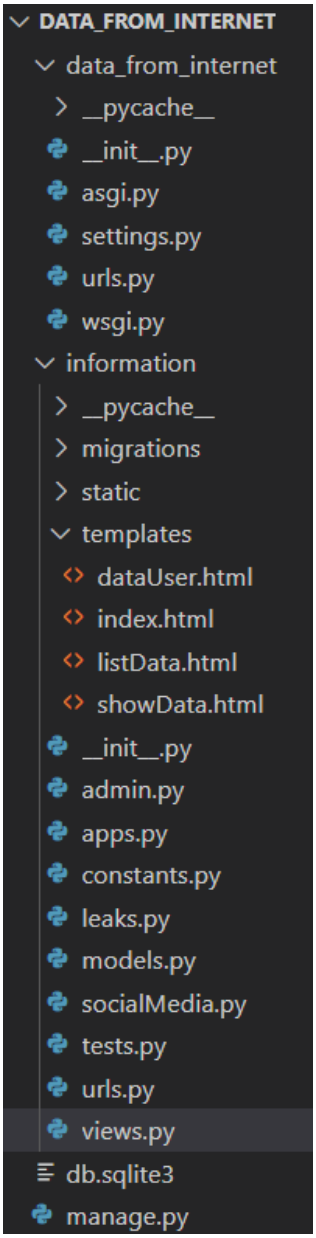


Ilustración 72. Estructura de carpetas

Pasaremos a explicar los métodos más importantes de la aplicación:

Dentro de view.py tendremos el método **index**, este se encargará de mostrar la página de inicio de la aplicación la cual contará con una barra de búsqueda. Si el usuario introduce un nombre se lanzará una petición de tipo POST a este mismo método y mediante SerApi podremos realizar una búsqueda personalizada utilizando Google Dorks para encontrar posibles nombres de usuarios asociados a la entrada introducida. Tras esto se vuelve a mostrar la vista de inicio, pero con una lista de

nombres de usuario, los cuales se podrán marcar para buscar información relacionada con ellos. Otra opción será introducir en la barra de búsqueda el atajo "@nombre_usuario" para buscar directamente información relacionada con este.

*Anotación: será necesario tener una cuenta de SerApi y añadir el api_key al fichero constants.py. SerApi proporciona 100 búsquedas mensuales en su plan gratuito.

```
def index(request):

    try:
        data = []
        if request.method == 'POST' and request.POST.get("name"):
            if "@" in request.POST.get("name"):
                return findSocialMedia(request)

            search = GoogleSearch({
            "q": '' + request.POST.get("name").strip() + ' intitle:"(@" site:twitter.com OR site:instagram.com OR site:facebook.com' +
            'OR site:tiktok.com OR site:pinterest.com',
            "location": "Spain",
            "hl": "es",
            "gl": "es",
            "num":"30",
            "api_key": SERAPI_API_KEY,
            })

            results = search.get_json()
            if 'organic_results' in results:
                for i in results['organic_results']:
                    if "(@" in i['title'] and ")" in i['title']:
                        user = re.split('\(@|\)', i['title'])
                        data.append( user[0]+" @"+ user[1])
            else:
                data.append('Número de intentos mensuales superados')

        return render(request, 'index.html', {'names':data})
```

Ilustración 73. Método "index"

En el método **findSocialMedia** comprobaremos la forma en la que se llama a este, es decir, si se ha usado el atajo o no. Tras esta comprobación pasaremos el nombre de usuario al método **socialMedia** que nos devolverá toda la información que sea posible recaudar sobre este. Finalmente almacenaremos los datos obtenidos en una variable de tipo JSON, la cual pasaremos a la vista donde mostraremos los datos y se guardará todo lo que se ha obtenido en la base de datos.

*Anotación: el método **saveData** creará una nueva entrada en la base de datos, además de crear una carpeta dentro de static/users_media con las imágenes y vídeos que hayan sido posible obtener. Con esto podremos saber la información que tenía

un usuario en una determinada fecha y compararla con otra, puesto que esta puede cambiar o ser eliminada del servidor donde esté almacenada a lo largo del tiempo.

```
def findSocialMedia(request):
    try:
        if request.method == 'POST':
            userData = {}
            values = []
            username = ''

            subprocess.check_output('del /Q information\static\images\*', shell=True)

            if request.POST.get("name"):
                if "@" in request.POST.get("name"):
                    socialMediaName = request.POST.get("name").replace("@","")
                    values.append(socialMedia(socialMediaName.strip()))
                    username = request.POST.get("name")

            else:
                if request.POST.getlist("users"):
                    for nameUser in request.POST.getlist("users"):
                        values.append(socialMedia(nameUser.split("@")[1].strip()))
                    username = request.POST.getlist("users")



            userData['users'] = values
            saveData(userData, username)

        return render(request, 'dataUser.html', {'account':request.POST.get("socialMediaName") ,'data': userData })

    except:
        return HttpResponseServerError("<h1>Se ha producido un error durante la búsqueda de datos</h1>")
```

Ilustración 74. Método "findSocialMedia"

En el método **socialMedia** se realizarán las distintas peticiones a las redes sociales de las que deseamos obtener información. Cada una de estas devolverá un JSON con los datos que ha podido recopilar. A continuación, pasaremos a ver una petición a una red social (alojada en socialMedia.py) y a explicar los métodos **checkLeaksEmail** e **informationLeaks** (alojados en leaks.py).

```python
def socialMedia(socialMediaName):
    try:
        userJson = {}
        leaks = []

        userJson['account'] = socialMediaName
        userJson['tiktok'] = findTikTok(socialMediaName)
        userJson['facebook'] = findFacebook(socialMediaName)
        userJson['pinterest'] = findPinterest(socialMediaName)
        userJson['instagram'] = findInstagram(socialMediaName)

        guest_token = getGuestToken()
        userJson['twitter'] = findTwitter(socialMediaName, guest_token)

        emailsLeak = checkLeakEmail(socialMediaName)

        if 'emails' in emailsLeak:
            for i in emailsLeak['emails']:
                search = informationLeaks(i['value'])
                if search and 'error' not in search:
                    leaks.append({'data': search})

        userJson['emailsLeak'] = emailsLeak
        userJson['leaks'] = leaks

        return userJson

    except:
        raise Exception("Fallo durante la búsqueda en redes sociales")
```

Ilustración 75. Método "socialMedia"

Examinaremos una petición a una red social en este caso Twitter. Hay que tener en cuenta que cada petición y la información que se obtiene es distinta, pero observando una de estas nos podemos hacer una idea de la estructura de las demás.

```python
def findTwitter(username, guest_token):
    dataJson = {}
    try:
        url = "https://twitter.com/i/api/graphql/mCbpQvZAw6zu_4PvuAUVVQ/UserByScreenName?variables=%7B%22screen_name%22%3A%22" + username + "%2

        payload = {}
        headers = {
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:101.0) Gecko/20100101 Firefox/101.0',
            'Accept': '*/*',
            'Accept-Language': 'es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3',
            'Accept-Encoding': 'identity',
            'content-type': 'application/json',
            'authorization': 'Bearer AAAAAAAAAAAAAAAAAAAAANRILgAAAAAAnNwIzUejRCOuH5E6I8xnZz4puTs%3D1Zv7ttfk8LF81IUq16cHjhLTvJu4FA33AGWWjCpTnA',
            'x-guest-token': guest_token,
            'x-twitter-client-language': 'es',
            'x-twitter-active-user': 'yes',
            'DNT': '1',
            'Connection': 'keep-alive',
            'Sec-Fetch-Dest': 'empty',
            'Sec-Fetch-Mode': 'cors',
            'Sec-Fetch-Site': 'same-origin',
            'TE': 'trailers'
        }

        response = requests.request("GET", url, headers=headers, data=payload)
```

Ilustración 76. Búsqueda de información en Twitter parte 1.

Tras realizar la petición con sus determinadas cabeceras se obtiene un JSON con información relacionada del nombre de usuario introducido. Seguidamente esta se almacena y pasa descargarse en este caso la imagen del perfil (en otras peticiones como por ejemplo Instagram se descargan imágenes y videos).

```python
    value = response.json()['data']
    if value and value['user']['result']['id']:

        dataJson['profile'] = 'Verdadero'
        results = value['user']['result']

        dataJson['id'] = results['rest_id']
        dataJson['name'] = results['legacy']['name']
        dataJson['created_at'] = results['legacy']['created_at']
        dataJson['location'] = results['legacy']['location']
        dataJson['followers'] = results['legacy']['followers_count']
        dataJson['following'] = results['legacy']['friends_count']
        dataJson['favourites'] = results['legacy']['favourites_count']
        dataJson['media'] = results['legacy']['media_count']
        dataJson['image_profile'] = re.sub("normal", "400x400", results['legacy']['profile_image_url_https'])
        dataJson['image_path'] = "images/" + username + "_twitter.jpg"
        urllib.request.urlretrieve(re.sub("normal", "400x400", results['legacy']['profile_image_url_https']),"information/static/images/"

        if results['legacy_extended_profile']:
            dataJson['day'] = results['legacy_extended_profile']['birthdate']['day']
            dataJson['month'] = results['legacy_extended_profile']['birthdate']['month']

        dataJson['tweets'] = getTweets(results['rest_id'], guest_token)

    else:
        dataJson['profile'] = 'Falso'

    return dataJson

except:
    dataJson['profile'] = 'No concluyente'
    return dataJson
```

Ilustración 77. Búsqueda de información en Twitter parte 2.

En el método **checkLeakEmail** a partir de un nombre de usuario se buscará si existe un correo que haya sido comprometido con ese mismo nombre, puesto que las personas tenemos tendencia a usar el mismo nombre para distintos sitios (Ej: nombre_usuario= jesus123 → resultado: comprobar jesus123@gmail.com, jesus123@hotmail.com, etc). Si el correo ha sido comprometido, es decir, se ha filtrado información sobre él, obtendremos la fecha de la filtración, sitio donde se filtró y el tipo de información filtrada.

```
def checkLeakEmail(username):
    try:
        site_emails = ['@gmail.com', '@hotmail.com', '@outlook.com', '@yahoo.com']
        dataJson = {}
        leaks = []

        for i in range(0, len(site_emails)):
            email = username + site_emails[i]
            response = subprocess.check_output('curl "https://haveibeenpwned.com/unifiedsearch/' + email +'" -H (continua..)
            data = response.decode()

            if(data):
                breaches = json.loads(data)
                leaks.append({'value': email ,'breaches': breaches['Breaches']})

        dataJson['emails'] = leaks
        return dataJson

    except:
        dataJson['error'] = 'Error al obtener información sobre emails'
        return dataJson
```

Ilustración 78. Método "checkLeakEmail"

Si queremos buscar fugas de datos usaremos la plataforma Intelx.io, como ya vimos esta permite encontrar bases de datos con distinta información que ha sido filtrada. Utilizaremos los emails que hemos comprobado en el paso anterior, pero solo aquellos que hayan tenido alguna filtración. En una primera instancia, realizaremos una consulta para ver en cuantos almacenes está el email que buscamos y obtener el identificador de dicho almacén.

*Anotación: Es necesario tener creada una cuenta en Intelx.io la cual permite 50 búsquedas diarias, bastará con registrarse.

```python
def informationLeaks(email):
    try:
        curl1 = subprocess.check_output('curl "https://2.intelx.io/intelligent/search" -X POST -H "User-Agent: Mozilla/5.0 (Windows
        data = curl1.decode()
        leaks = []

        url = "https://2.intelx.io/intelligent/search/result?id=" + data.split('"')[3] + "&limit=10&statistics=1&previewlines=8"

        payload={}
        headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:101.0) Gecko/20100101 Firefox/101.0',
        'Accept': '*/*',
        'Accept-Language': 'es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3',
        'Accept-Encoding': 'gzip, deflate, br',
        'x-key': INTELX_API_KEY,
        'Origin': 'https://intelx.io',
        'Connection': 'keep-alive',
        'Referer': 'https://intelx.io/',
        'Sec-Fetch-Dest': 'empty',
        'Sec-Fetch-Mode': 'cors',
        'Sec-Fetch-Site': 'same-site',
        'TE': 'trailers'
        }

        response = requests.request("GET", url, headers=headers, data=payload)

        res = response.json()
```

Ilustración 79. Método "informationLeaks" parte 1

Tras obtener los identificadores de donde hay una coincidencia con el email que se ha introducido procedemos a su descarga y limpieza para guardar la información en crudo.

```python
        for i in res['records']:

            bucket = re.sub(" ","",i['bucketh'])

            url = "https://2.intelx.io/file/view?f=16&storageid=" + i['storageid'] + "&bucket=" + re.sub("»",".",bucket).lower() +

            payload={}
            headers = {
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:101.0) Gecko/20100101 Firefox/101.0',
            'Accept': '*/*',
            'Accept-Language': 'es-ES,es;q=0.8,en-US;q=0.5,en;q=0.3',
            'Accept-Encoding': 'gzip, deflate, br',
            'Origin': 'https://intelx.io',
            'DNT': '1',
            'Connection': 'keep-alive',
            'Referer': 'https://intelx.io/',
            'Sec-Fetch-Dest': 'empty',
            'Sec-Fetch-Mode': 'cors',
            'Sec-Fetch-Site': 'same-site',
            'TE': 'trailers'
            }

            response = requests.request("GET", url, headers=headers, data=payload)
            text = response.text.split("\n")
            for t in text:
                if email in t:
                    leaks.append({'leak':t})


        return leaks

    except:
        return {'error': 'Error al obtener leaks de emails'}
```

Ilustración 80. Método "informationLeaks" parte 2

3.6 Pruebas

Puesto que las pruebas son algo que ha sido necesario durante todo el desarrollo, nos centraremos en aquellas que han sido más importantes:

Como hemos visto se realizan peticiones a diferentes sitios, por eso la realización de pruebas ha sido imprescindible desde el primer momento, dado que para comenzar se ha tenido que realizar un análisis de las peticiones de red dentro del dominio en cuestión y tras esto es necesario adaptar dicha petición al lenguaje de programación que estemos usando, en este caso Python. Es por todo esto que con cada red social que se ha implementado la búsqueda, ha sido necesario su evaluación individual comprobando el correcto funcionamiento y haciendo una selección de la información más relevante que se puede obtener.

Tras haber comprobado que todas las peticiones funcionan correctamente de manera individual, se procede a desarrollar una interfaz bastante simple donde poder mostrar la unión de todas las peticiones dentro de un mismo método. A partir de este momento también comienza el desarrollo paralelo de la interfaz y la realización de pruebas sobre esta, hasta llegar al resultado final.

Seguidamente, se prueban las peticiones para comprobar si un email con el nombre de usuario que se busca ha sido comprometido y si encontramos información sobre este dentro de la plataforma Intelx.io, dado que estas peticiones están enlazadas. Posteriormente se integrarán y probarán su funcionamiento dentro del método donde se encuentran las peticiones a las redes sociales.

Finalmente, se realizan las últimas pruebas donde se examina el correcto funcionamiento de todos los componentes que intervienen en el resultado final del prototipo.

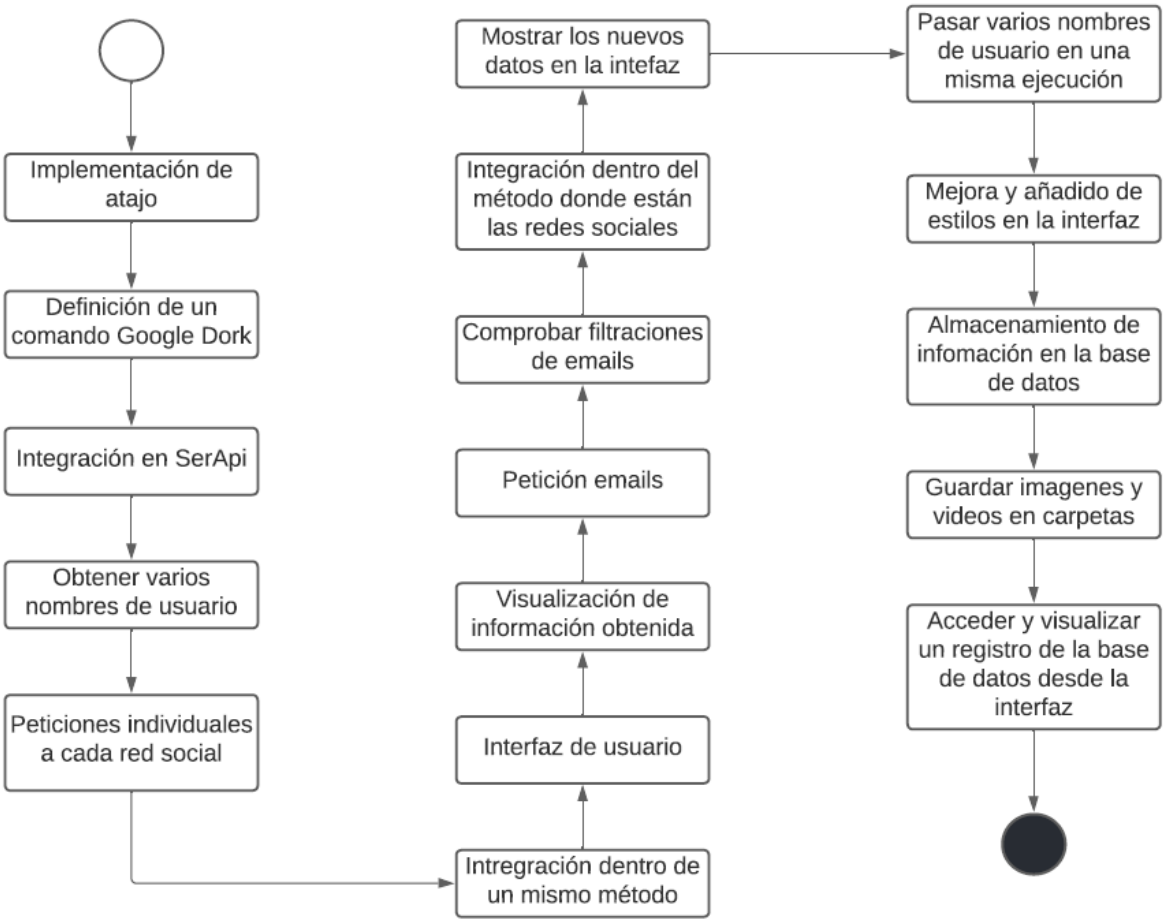


Ilustración 81. Secuencia de pruebas.

3.7 Mantenimiento y posibles mejoras

El mantenimiento es una parte esencial cuando se despliega un software. Este incluye mejoras en rasgos generales, mejora de rendimiento, corrección de errores y problemas que no hayan sido detectados o pasados de vista.

Como tarea de mantenimiento periódica debe realizarse una exhaustiva revisión de las peticiones a los distintos sitios, dado que estos no tienen por qué mantener su estructura, ni la información que proporcionan a lo largo del tiempo. Cada vez que hagamos un mantenimiento de estas, deberían de hacerse las siguientes preguntas:

1. ¿La petición sigue siendo válida?
2. ¿La estructura de la web o API sigue siendo la misma?
3. ¿Se necesita añadir/suprimir cookies o información a las cabeceras de la petición?

4. ¿La información que nos proporciona sigue siendo la misma o podemos llegar a obtener más información útil?

Una posible mejora sería cambiar la búsqueda inicial por la que se obtienen los nombres de usuario, llegando a desarrollar una búsqueda propia o mediante el uso de librerías que no requieran el uso de una API externa, además de una posible mejora o añadido de Google Dorks para obtener un mayor alcance.

Otra sería la creación de módulos para descargar los datos almacenados en diferentes formatos, pudiendo llegar a presentar informes detallados capaces de ser útiles para la realización de análisis.

También se deberá hacer un mantenimiento y mejora de la interfaz comprobando el correcto funcionamiento de todos los elementos que la componen y de todo aquello que se vaya añadiendo con cada actualización.

CAPÍTULO 4: Conclusiones

Actualmente se genera constantemente una gran cantidad de datos en internet, los cuales dejan una huella. Cumpliendo con los objetivos del proyecto hemos visto los tipos de fuentes abiertas que se pueden encontrar y como mediante el proceso de búsqueda, selección, recopilación de la información y posteriormente un procesamiento, se puede llegar a convertir la información obtenida en conocimiento útil.

Existen una gran cantidad de herramientas gratuitas, tanto por web como por línea de comandos, capaces de conseguir diferentes tipos de información, que no requieren, en algunos casos una gran cantidad de conocimiento sobre estas y que están al alcance de todo el mundo. Algo tan común como navegar por Google, que se usa diariamente, brinda un mayor alcance en sus búsquedas mediante el uso de comandos, pudiendo obtener datos que serían difícil de encontrar en una búsqueda común.

Ya que es posible obtener información de distintas fuentes datos y cada herramienta realiza consultas a distintas fuentes o la forma en la cual indexa estos. Por eso, otra parte importante es cotejar los datos, permitiendo así identificar aquellos que no son útiles para dar más valor a aquellos que sí lo son. De igual manera tras el estudio y análisis en profundidad de estas fuentes abiertas, podemos desarrollar nuevas herramientas para la extracción de información, dado que hay una gran cantidad de recursos que se pueden aprovechar para el desarrollo de una herramienta funcional.

Por último, considero que este trabajo ha conseguido de manera satisfactoria los objetivos propuestos, además espero que pueda ser útil y dé una visión de la gran cantidad de información expuesta en internet. Puesto que es una tecnología “reciente” y en poco tiempo el número de usuarios ha ido creciendo exponencialmente, es importante la concienciación de que casi todo lo que hacemos en internet deja una huella que en la mayoría de los casos no pude ser borrada.

Bibliografía

https://wearesocial.com/
https://marketing4ecommerce.net/historia-de-internet/
https://marketing4ecommerce.net/que-pasa-en-internet-en-un-minuto-infografia/
https://es.wikipedia.org/wiki/Internet_profunda
https://www.incibe-cert.es/blog/osint-la-informacion-es-poder
https://rockcontent.com/es/blog/motores-de-busqueda/
https://support.google.com/websearch/answer/2466433?hl=es
https://www.seguridadinternacional.es/
https://www.exploit-db.com/google-hacking-database
https://osintframework.com/
https://github.com/laramies/theHarvester
https://crt.sh/
https://www.incibe.es/protege-tu-empresa/blog/certificado-digital-ssl-sitio-web-seleccionar-uno
https://dnsdumpster.com/footprinting-reconnaissance/
https://www.docusign.mx/blog/tipos-de-servidores
https://developer.shodan.io/api
https://archive.org/web/
https://github.com/ElevenPaths/FOCA
https://intelx.io/about
https://docs.maltego.com/support/home
https://www.namecheckr.com/help
https://github.com/sherlock-project/sherlock
https://github.com/Datalux/Osintgram
https://socialbearing.com/
https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/
https://code.visualstudio.com/docs
https://docs.python.org/3/
https://docs.djangoproject.com/en/4.1/

Anexo: Instalación

Consideraremos el proyecto como una aplicación portable, en la cual se podrán seguir desarrollando mejoras. Para la instalación será necesario seguir los siguientes pasos:

1. Extraer la carpeta que contiene el proyecto.
2. Descargar Python 3.8.10, eligiendo su versión según la arquitectura del sistema.

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		83d71c304acab6c678e86e239b42fa7e	24720640	SIG
XZ compressed source tarball	Source release		d9eee4b20155553830a2025e4dcaa7b3	18433456	SIG
macOS 64-bit Intel installer	macOS	for macOS 10.9 and later	690ddb1be403a7efb202e93f3a994a49	29896827	SIG
macOS 64-bit universal2 installer	macOS	experimental, for macOS 11 Big Sur and later; recommended on Apple Silicon	ae8a1ae082074b260381c058d0336d05	37300939	SIG
Windows embeddable package (32-bit)	Windows		659adf421e90fba0f56a9631f79e70fb	7348969	SIG
Windows embeddable package (64-bit)	Windows		3acb1d7d9bde5a79f840167b166bb633	8211403	SIG
Windows help file	Windows		a06af1ff933a13f6901a75e59247cf95	8597086	SIG
Windows installer (32-bit)	Windows		b355cfc84b681ace8908ae50908e8761	27204536	SIG
Windows installer (64-bit)	Windows	Recommended	62cf1a12a5276b0259e8761d4cf4fe42	28296784	SIG

Ilustración 82. Descarga de Python.

3. Ejecutar el instalador y seleccionar “Install Now”.

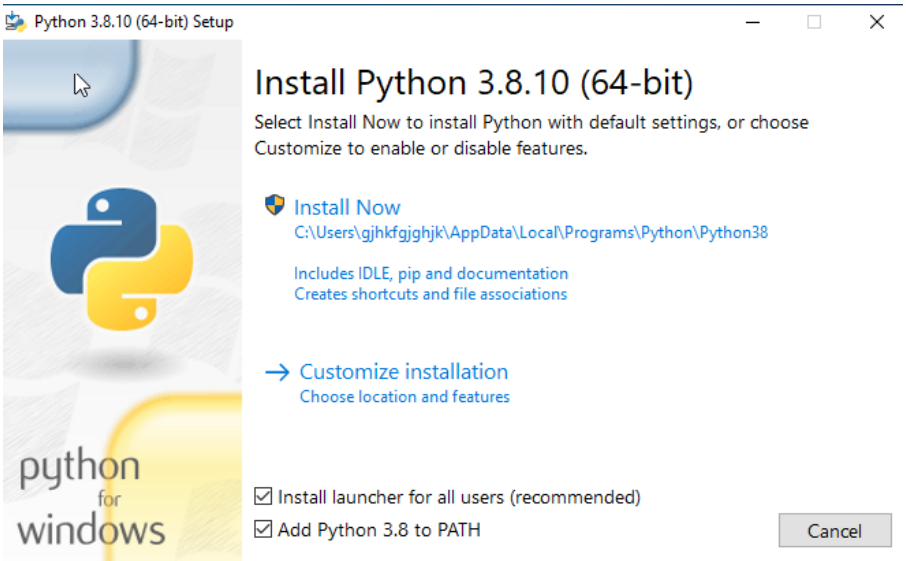


Ilustración 83. Instalación.

4. Cuando termine el proceso de instalación, abriremos un terminal de Windows y comprobaremos que Python está instalado correctamente mediante el comando “python --version”. Tras comprobar que aparece la versión 3.8.10, procedemos a instalar Django mediante el comando “pip install Django==3.2”.

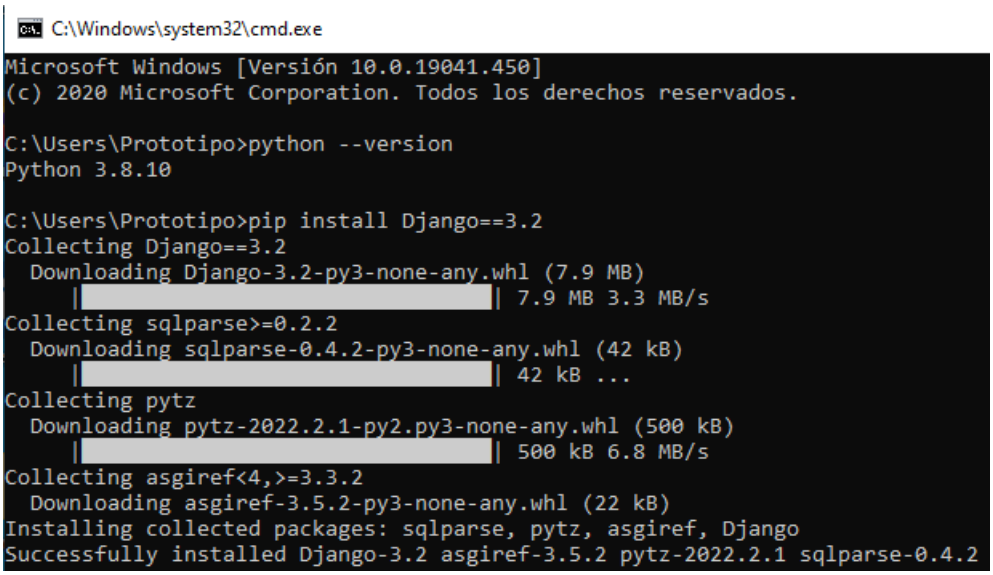


Ilustración 84. Instalación de Django.

5. Sera necesario la instalación de librerías de Python para el correcto funcionamiento. Desde el terminal de Windows nos moveremos a la dentro de la carpeta donde está alojado el proyecto e introduciremos el comando “pip install –r requirements.txt”.

```
C:\Users\Prototipo\Desktop\App\data_from_internet>pip install -r requirements.txt
```

Ilustración 85. Instalación de librerías.

6. Finalmente, mediante el comando “python manage.py runserver” se ejecutará la aplicación y será posible acceder introduciendo “http://127.0.0.1:8000” en el navegador web.

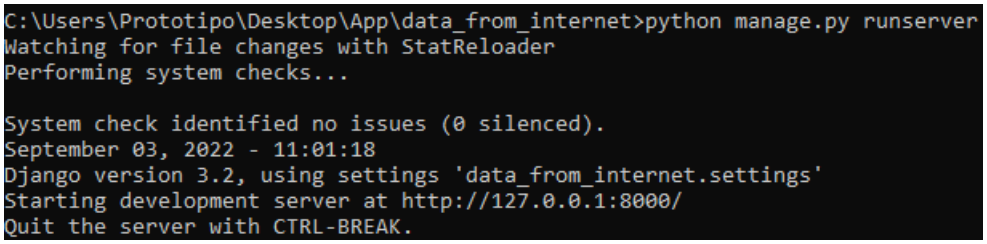


Ilustración 86. Ejecución de la aplicación.

Ilustración 87. Página inicial.