

UNIVERSIDAD POLITÉCNICA DE MADRID



**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA
DE SISTEMAS INFORMÁTICOS**

GRADO EN INGENIERÍA DEL SOFTWARE

PROYECTO FIN DE GRADO

**Automatización de la adquisición de
datos en fuentes abiertas (OSINT)**

Rubén Álvarez Elena

Rodrigo Baladrón de Juan

Curso académico 2019/20

PROYECTO FIN DE GRADO

TÍTULO: Automatización de la adquisición de datos en fuentes abiertas (OSINT)

AUTORES: Rubén Álvarez Elena, Rodrigo Baladrón de Juan

TUTOR/DIRECTOR: Jesús Sánchez López

Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

TRIBUNAL:

PRESIDENTE/A: D^a. Jessica Díaz Fernández

VOCAL: D. Jesús Martínez Barbero

SECRETARIO/A: D. Jesús Sánchez López

FECHA DE LECTURA: 5 de mayo de 2020

CALIFICACIÓN: 10

Fdo: El Secretario del Tribunal

Agradecimientos

A nuestro tutor profesional, David, por permitirnos llevar a cabo este proyecto. Y, por supuesto, a nuestros padres, quienes han hecho posible que seamos ingenieros.

Resumen

En el presente Proyecto de Fin de Grado se automatiza la adquisición de datos que se encuentran en fuentes abiertas para la posterior generación de inteligencia (OSINT), almacenando dichos datos en Elasticsearch. Cabe mencionar que se ha realizado la planificación del proyecto siguiendo el PMBOK.

Se ha utilizado el framework Recon-ng, que ya dispone de diversos módulos para adquirir datos de fuentes abiertas, y se ha realizado un desarrollo de nuevos módulos en Python de acuerdo con las necesidades de nuestra empresa. De esta manera, con los módulos desarrollados se adquieren direcciones de correo, documentos, noticias, posts de foros, pastes, servidores de correo en listas negras y dominios similares a partir de unos datos de entrada, tales como dominios y nombres de marca.

Dado que este framework utiliza una base datos SQL (SQLite), se ha realizado una reingeniería para migrarla a NoSQL (Elasticsearch). Así, se ha integrado Elasticsearch con Recon-ng para almacenar los datos obtenidos con los módulos, con la consiguiente adaptación de los componentes del framework que interaccionan con la base de datos.

Por cada workspace de Recon-ng se crea un índice en Elasticsearch, donde se indexan los documentos con los datos recopilados, organizándose dichos documentos según el tipo de datos que contienen. Además, puesto que Elasticsearch no tiene un control de la duplicidad de los documentos que se indexan al utilizar por defecto un _id autogenerado, se ha implementado un método para evitar la redundancia de datos.

Asimismo, los datos recopilados con los módulos desarrollados e indexados en Elasticsearch se visualizan con Kibana. También se ha desarrollado un módulo de reportes para exportar los datos en Excel con el formato de los informes corporativos.

Abstract

The aim of this project is to automate the acquisition of data from publicly available sources for intelligence gathering (OSINT), storing the data in Elasticsearch. It is worth mentioning that a project planning has been done in accordance with the PMBOK.

The Recon-*ng* framework already has several modules to collect data from publicly available sources, and new modules have been developed in Python based on our company needs. In this way, emails addresses, documents, news, forum posts, pastes, blacklisted mail servers and similar domains are collected with the developed modules, using domain and brand names as input.

This framework has a SQL database (SQLite), so a reengineering has been carried out in order to migrate it to NoSQL (Elasticsearch). Elasticsearch has been integrated with Recon-*ng* to store the data gathered through the modules, adapting the framework components that interact with the database.

An index is created in Elasticsearch for each Recon-*ng* workspace, where documents containing the collected data are indexed, organised by type of data. In addition, due to the lack of duplicity control in Elasticsearch as it uses an autogenerated `_id` by default, a method has been implemented to avoid data redundancy.

Furthermore, the data gathered through the developed modules and indexed in Elasticsearch is visualized with Kibana. A reporting module has also been developed, which exports the data to Excel according to the corporate report format.

Contenido

Agradecimientos.....	i
Resumen	iii
Abstract	v
Contenido	vii
Listado de figuras	ix
Listado de tablas	xiii
Acrónimos.....	xv
1 Introducción	1
1.1 Objetivo y motivación	1
1.2 Estructura del documento	2
2 Planificación	3
2.1 Matriz de trazabilidad de requisitos.....	3
2.2 EDT/WBS	4
2.3 Diccionario WBS.....	4
2.4 Actividades	7
2.5 Matriz de asignación de responsabilidades	10
2.6 Diagrama de red del cronograma del proyecto (PERT).....	14
2.6.1 PERT de actividades	14
2.6.2 Duración y holguras.....	14
2.6.3 Camino crítico.....	17
2.7 Cronograma del proyecto.....	18
2.8 Plan de gestión de costos	19
2.8.1 Estimación de costos.....	19
2.9 Presupuesto del proyecto.....	22
3 Estado del arte	23
3.1 Inteligencia	23
3.2 OSINT.....	24
3.2.1 Evolución de OSINT	25
3.2.2 OSINT en la era de Internet	26
3.3 Metodología OSINT.....	28
3.3.1 Primera fase: Adquisición.....	28
3.3.2 Segunda fase: Procesamiento	29
3.3.3 Tercera fase: Análisis	29
3.3.4 Cuarta fase: Producción	32
3.4 Herramientas OSINT	32
3.4.1 FOCA.....	33
3.4.2 Maltego.....	34
3.4.3 Recon-ng	35
3.4.4 Shodan	35
3.4.5 Spiderfoot	36
3.4.6 The Harvester.....	37

3.5	Comparación de herramientas OSINT	37
3.6	Bases de datos SQL y NoSQL.....	38
3.7	Herramientas para el análisis de datos recopilados en fuentes abiertas.....	39
3.8	Elasticsearch	40
3.8.1	¿Cómo funciona Elasticsearch?.....	40
3.8.2	¿Qué es un índice de Elasticsearch?.....	40
3.8.3	¿Por qué usar Elasticsearch?	41
3.8.4	Kibana.....	42
4	Reingeniería	43
4.1	Ingeniería inversa.....	43
4.2	Plan de migración.....	47
4.3	Migración del modelo de datos	48
4.3.1	Correspondencia entre SQL y Elasticsearch	48
4.3.2	Estructura de los documentos de Elasticsearch	49
4.4	Integración de Elasticsearch en Recon-ng	51
4.4.1	Creación de índices y operaciones CRUD.....	51
4.4.2	Obtención de los datos de entrada de los módulos	54
4.4.3	Inserción de los datos adquiridos con los módulos.....	55
5	Documentación del desarrollo.....	56
5.1	Módulo de adquisición de Dominios similares	56
5.2	Módulo de adquisición de Direcciones de correo electrónico	65
5.3	Módulo de adquisición de Direcciones de correo en Hunter	69
5.4	Módulo de adquisición de Noticias.....	73
5.5	Módulo de adquisición de Documentos	76
5.6	Módulo de búsqueda de coincidencias en servicios de compartición de texto online 78	78
5.6.1	Pastebin.....	78
5.6.2	GitHub Gist	84
5.7	Módulo de búsqueda de menciones en foros	87
5.7.1	ElOtroLado	88
5.7.2	Reddit	92
5.7.3	Forocoches.....	95
5.8	Módulo de comprobación de dominios en listas negras	100
5.9	Módulo de exportación de los datos adquiridos a Excel.....	103
5.9.1	Extracción de los datos indexados en Elasticsearch	104
5.9.2	Método básico de creación de tablas (reputación, pastes y foros)	106
5.9.3	Método de creación de tablas agrupadas por dominios (emails, documentos y dominios similares).....	110
5.9.4	Método de creación de tablas con agrupación por filas con datos en común (fuente de localización de emails, listas negras).....	118
5.9.5	Método de creación de tablas con combinación de columnas (metadatos) .	120
6	Guía de uso	124
6.1	Recon-vd	124
6.1.1	Instalación	124
6.1.2	API Keys	126
6.1.3	Workspace.....	127
6.1.4	Módulos	127
6.1.5	Ejecución de comandos desde un fichero.....	131
6.2	Instalación de Elasticsearch y Kibana.....	132
7	Conclusiones	136
8	Impacto social y legal	138
9	Bibliografía.....	139

Listado de figuras

Ilustración 1 - EDT/WBS4
Ilustración 2 - PERT de actividades	14
Ilustración 3 - Camino crítico	17
Ilustración 4 - Cronograma del proyecto	18
Ilustración 5 - Diagrama Entidad-Relación.....	43
Ilustración 6 - Método <code>_init_workspace()</code> de la clase Recon	44
Ilustración 7 - Ejecución de consultas SQL en el método <code>_query()</code> de la clase Framework.....	44
Ilustración 8 - Métodos “insert” de la clase Framework.....	45
Ilustración 9 - Método <code>insert()</code> de la clase Framework	45
Ilustración 10 - Consulta SQL para obtener el input en los módulos	46
Ilustración 11 - Método <code>_get_source()</code> de la clase <code>BaseModule</code>	46
Ilustración 12 - Tabla de la base de datos SQL	48
Ilustración 13 - Documentos de Elasticsearch	48
Ilustración 14 - Método <code>connect_ES()</code> agregado a la clase Framework.....	51
Ilustración 15 - Método <code>create_index_ES()</code> agregado a la clase Recon	51
Ilustración 16 - Llamada al método <code>self.create_index_ES()</code> agregada en el método <code>_init_workspace()</code> de la clase Recon	51
Ilustración 17 - Métodos agregados a la clase Framework para realizar las operaciones CRUD en Elasticsearch.....	52
Ilustración 18 - Método <code>create_id()</code> agregado a la clase Framework	53
Ilustración 19 - Consulta para obtener el input de Elasticsearch en los módulos	54
Ilustración 20 - Método <code>_get_source()</code> modificado	54
Ilustración 21 - Método “insert” modificado	55
Ilustración 22 - Información del módulo de adquisición de Dominios similares	56
Ilustración 23 - Primer fragmento de código del método <code>module_run()</code> de adquisición de Dominios similares	57
Ilustración 24 - Segundo fragmento de código del método <code>module_run()</code> de adquisición de Dominios similares.....	57
Ilustración 25 - Segundo fragmento de código del método <code>module_run()</code> de adquisición de Dominios similares.....	57
Ilustración 26 - Documento JSON con las permutaciones del dominio en dnstwister.....	58
Ilustración 27 - Tercer fragmento de código del método <code>module_run()</code> de adquisición de Dominios similares	58
Ilustración 28 - Cuarto fragmento de código del método <code>module_run()</code> de adquisición de Dominios similares	59
Ilustración 29 - Quinto fragmento de código del método <code>module_run()</code> de adquisición de Dominios similares	59
Ilustración 30 - Sexto fragmento de código del método <code>module_run()</code> de adquisición de Dominios similares	59
Ilustración 31 - Primer fragmento de código del método <code>module_threat()</code> de adquisición de Dominios similares.....	60
Ilustración 32 - Segundo fragmento de código del método <code>module_threat()</code> de adquisición de Dominios similares.....	60
Ilustración 33 - Tercer fragmento de código del método <code>module_threat()</code> de adquisición de Dominios similares.....	61
Ilustración 34 - Primer fragmento de código del método <code>virustotalScan()</code> de adquisición de Dominios similares.....	61
Ilustración 35 - Segundo fragmento de código del método <code>virustotalScan()</code> de adquisición de Dominios similares.....	62
Ilustración 36 - Tercer fragmento de código del método <code>virustotalScan()</code> de adquisición de Dominios similares.....	62
Ilustración 37 - Documento JSON con el resultado del análisis en VirusTotal.....	63

Ilustración 38 - Cuarto fragmento de código del método virustotalScan() de adquisición de Dominios similares.....	63
Ilustración 39 - Primer fragmento de código del método archiveSave() de adquisición de Dominios similares.....	64
Ilustración 40 - Segundo fragmento de código del método archiveSave() de adquisición de Dominios similares.....	64
Ilustración 41 - Tercer fragmento de código del método archiveSave() de adquisición de Dominios similares.....	64
Ilustración 42 - Información del módulo de adquisición de Direcciones de correo electrónico	65
Ilustración 43 - Primer fragmento de código del método module_run() de adquisición de Direcciones de correo electrónico	65
Ilustración 44 - Segundo fragmento de código del método module_run() de adquisición de Direcciones de correo electrónico	66
Ilustración 45 - Tercer fragmento de código del método module_run() de adquisición de Direcciones de correo electrónico	66
Ilustración 46 - Cuarto fragmento de código del método module_run() de adquisición de Direcciones de correo electrónico	67
Ilustración 47 - Quinto fragmento de código del método module_run() de adquisición de Direcciones de correo electrónico	67
Ilustración 48 - Sexto fragmento de código del método module_run() de adquisición de Direcciones de correo electrónico	67
Ilustración 49 - Séptimo fragmento de código del método module_run() de adquisición de Direcciones de correo electrónico	68
Ilustración 50 - Octavo fragmento de código del método module_run() de adquisición de Direcciones de correo electrónico	68
Ilustración 51 - Información del módulo de adquisición de Direcciones de correo en Hunter.....	69
Ilustración 52 - Primer fragmento de código del método module_run() de adquisición de Direcciones de correo en Hunter.....	69
Ilustración 53 - Documento JSON con los datos de las direcciones de correo obtenidas con la API de Hunter	70
Ilustración 54 - Segundo fragmento de código del método module_run() de adquisición de Direcciones de correo en Hunter	71
Ilustración 55 - Tercer fragmento de código del método module_run() de adquisición de Direcciones de correo en Hunter	71
Ilustración 56 - Cuarto fragmento de código del método module_run() de adquisición de Direcciones de correo en Hunter.....	71
Ilustración 57 - Quinto fragmento de código del método module_run() de adquisición de Direcciones de correo en Hunter.....	72
Ilustración 58 - Información del módulo de adquisición de Noticias.....	73
Ilustración 59 - Primer fragmento de código del método module_run() de adquisición de Noticias	73
Ilustración 60 - Segundo fragmento de código del método module_run() de adquisición de Noticias	74
Ilustración 61 - Tercer fragmento de código del método module_run() de adquisición de Noticias	74
Ilustración 62 - Cuarto fragmento de código del método module_run() de adquisición de Noticias	75
Ilustración 63 - Información del módulo de adquisición de Documentos	76
Ilustración 64 - Primer fragmento de código del método module_run() de adquisición de Documentos.....	76
Ilustración 65 - Segundo fragmento de código del método module_run() de adquisición de Documentos.....	77
Ilustración 66 - Tercer fragmento de código del método module_run() de adquisición de Documentos.....	77
Ilustración 67 - Cuarto fragmento de código del método module_run() de adquisición de Documentos	77
Ilustración 68 - Información del módulo de adquisición de Pastes.....	78
Ilustración 69 - Código del método module_run() del módulo de adquisición de Pastes	78
Ilustración 70 - Primer fragmento de código del método pastebin() de adquisición de Pastes	79
Ilustración 71 - Segundo fragmento de código del método pastebin() de adquisición de Pastes	80
Ilustración 72 - Elemento HTML con el número de página	80
Ilustración 73 - Tercer fragmento de código del método pastebin() de adquisición de Pastes	80
Ilustración 74 - Elemento HTML que contiene la URL de cada resultado.....	81
Ilustración 75 - Cuarto fragmento de código del método pastebin() de adquisición de Pastes.....	81
Ilustración 76 - Quinto fragmento de código del método pastebin() de adquisición de Pastes	81
Ilustración 77 - Sexto fragmento de código del método pastebin() de adquisición de Pastes	82
Ilustración 78 - Séptimo fragmento de código del método pastebin() de adquisición de Pastes	82
Ilustración 79 - Elemento HTML que contiene la fecha del resultado	82
Ilustración 80 - Octavo fragmento de código del método pastebin() de adquisición de Pastes	83
Ilustración 81 - Elemento HTML con el contenido raw	83
Ilustración 82 - Noveno fragmento de código del método pastebin() de adquisición de Pastes	83
Ilustración 83 - Décimo fragmento de código del método pastebin() de adquisición de Pastes	84
Ilustración 84 - Primer fragmento de código del método gistGithub() de adquisición de Pastes	84
Ilustración 85 - Segundo fragmento de código del método gistGithub() de adquisición de Pastes	84
Ilustración 86 - Elemento HTML que contiene la URL de los resultados	85

Ilustración 87 - Tercer fragmento de código del método gistGitHub() de adquisición de Pastes.....	85
Ilustración 88 - Cuarto fragmento de código del método gistGitHub() de adquisición de Pastes	85
Ilustración 89 - Quinto fragmento de código del método gistGitHub() de adquisición de Pastes	86
Ilustración 90 - Sexto fragmento de código del método gistGitHub() de adquisición de Pastes	86
Ilustración 91 - Elemento HTML con la URL de la página siguiente	86
Ilustración 92 - Séptimo fragmento de código del método gistGitHub() de adquisición de Pastes	86
Ilustración 93 - Información del módulo de adquisición de Posts.....	87
Ilustración 94 - Código del método module_run() de adquisición de Posts	88
Ilustración 95 - Primer fragmento de código del método elOtroLado() de adquisición de Posts	88
Ilustración 96 - Segundo fragmento de código del método elOtroLado() de adquisición de Posts	89
Ilustración 97 - Tercer fragmento de código del método elOtroLado() de adquisición de Posts	89
Ilustración 98 - Elemento HTML que contiene la URL del resultado	89
Ilustración 99 - Cuarto fragmento de código del método elOtroLado() de adquisición de Posts	90
Ilustración 100 - Quinto fragmento de código del metodo elOtroLado() de adquisición de Posts	90
Ilustración 101 - Elemento HTML con la fecha del resultado	90
Ilustración 102 - Sexto fragmento de código del método elOtroLado() de adquisición de Posts	91
Ilustración 103 - Página de error	92
Ilustración 104 - Séptimo fragmento de código del método elOtroLado() de adquisición de Posts	92
Ilustración 105 - Primer fragmento de código del método reddit() de adquisición de Posts.....	92
Ilustración 106 - Segundo fragmento de código del método reddit() de adquisición de Posts	93
Ilustración 107 - Tercer fragmento de código del método reddit() de adquisición de Posts	93
Ilustración 108 - Cuarto fragmento de código del método reddit() de adquisición de Posts	93
Ilustración 109 - Quinto fragmento de código del método reddit() de adquisición de Posts	93
Ilustración 110 - Elemento HTML con la URL de los resultados	94
Ilustración 111 - Sexto fragmento de código del método reddit() de adquisición de Posts	94
Ilustración 112 - Séptimo fragmento de código del método reddit() de adquisición de Posts	94
Ilustración 113 - Octavo fragmento de código del método reddit() de adquisición de Posts.....	95
Ilustración 114 - Primer fragmento de código del método forocoches() de adquisición de Posts	95
Ilustración 115 - Segundo fragmento de código del método forocoches() de adquisición de Posts	95
Ilustración 116 - Elemento HTML con la URL de los resultados	96
Ilustración 117 - Tercer fragmento de código del método forocoches() de adquisición de Posts	96
Ilustración 118 - Elemento HTML con el título del resultado	96
Ilustración 119 - Cuarto fragmento de código del método forocoches() de adquisición de Posts	97
Ilustración 120 - Elemento HTML que contiene la fecha	97
Ilustración 121 - Quinto fragmento de código del método forocoches() de adquisición de Posts	97
Ilustración 122 - Sexto fragmento de código del método forocoches() de adquisición de Posts	98
Ilustración 123 - Séptimo fragmento de código del método forocoches() de adquisición de Posts	98
Ilustración 124 - Elemento HTML que contiene la fecha	98
Ilustración 125 - Octavo fragmento de código del método forocoches() de adquisición de Posts	99
Ilustración 126 - Noveno fragmento de código del método forocoches() de adquisición de Posts	99
Ilustración 127 - Elemento HTML con el enlace a la página siguiente de Google	99
Ilustración 128 - Información del módulo de comprobación de dominios en listas negras	100
Ilustración 129 - Primer fragmento de código del módulo de Listas negras de spam	100
Ilustración 130 - Segundo fragmento de código del módulo de Listas negras de spam	100
Ilustración 131 - Tercer fragmento de código del módulo de Listas negras de spam	100
Ilustración 132 - Cuarto fragmento de código del módulo de Listas negras de spam	101
Ilustración 133 - Quinto fragmento de código del módulo de Listas negras de spam	101
Ilustración 134 - Sexto fragmento de código del módulo de Listas negras de spam	101
Ilustración 135 - Información del módulo de exportación a Excel	103
Ilustración 136 - Primer fragmento de código del método module_run() de exportación a Excel	104
Ilustración 137 - Segundo fragmento de código del método module_run() de exportación a Excel.....	104
Ilustración 138 - Tercer fragmento de código del método module_run() de exportación a Excel	104
Ilustración 139 - Cuarto fragmento de código del método module_run() de exportación a Excel	105
Ilustración 140 - Quinto fragmento de código del método module_run() de exportación a Excel	105
Ilustración 141 - Sexto fragmento de código del método module_run() de exportación a Excel.....	106
Ilustración 142 - Primer fragmento de código del método dictsToTable() de exportación a Excel	106
Ilustración 143 - Segundo fragmento de código del método dictsToTable() de exportación a Excel	107
Ilustración 144 - Tercer fragmento de código del método dictsToTable() de exportación a Excel	107
Ilustración 145 - Cuarto fragmento de código del método dictsToTable() de exportación a Excel	108
Ilustración 146 - Quinto fragmento de código del método dictsToTable() de exportación a Excel	108
Ilustración 147 - Sexto fragmento de código del método dictsToTable() de exportación a Excel	109
Ilustración 148 - Octavo fragmento de código del método dictsToTable() de exportación a Excel	109
Ilustración 149 - Hoja de Reputación	109
Ilustración 150 - Hoja de Pastes	110
Ilustración 151 - Primer fragmento de código del método emailsToTables() de exportación a Excel	110
Ilustración 152 - Segundo fragmento de código del método emailsToTables() de exportación a Excel ...	111
Ilustración 153 - Tercer fragmento de código del método emailsToTables() de exportación a Excel	111

Ilustración 154 - Cuarto fragmento de código del método emailsToTables() de exportación a Excel.....	112
Ilustración 155 - Quinto fragmento de código del método emailsToTables() de exportación a Excel.....	112
Ilustración 156 - Sexto fragmento de código del método emailsToTables() de exportación a Excel	113
Ilustración 157 - Séptimo fragmento de código del método emailsToTables() de exportación a Excel	113
Ilustración 158 - Octavo fragmento de código del método emailsToTables() de exportación a Excel	113
Ilustración 159 - Noveno fragmento de código del método emailsToTables() de exportación a Excel	114
Ilustración 160 - Décimo fragmento de código del método emailsToTables() de exportación a Excel	114
Ilustración 161 - Undécimo fragmento de código del método emailsToTables() de exportación a Excel .	114
Ilustración 162 - Hoja de Emails	115
Ilustración 163 - Hoja de Documentos	115
Ilustración 164 - Primer fragmento de código del método similarDomainsToTables() de exportación a Excel	116
Ilustración 165 - Segundo fragmento de código del método similarDomainsToTables() de exportación a Excel	116
Ilustración 166 - Tercer fragmento de código del método similarDomainsToTables() de exportación a Excel	116
Ilustración 167 - Cuarto fragmento de código del método similarDomainsToTables() de exportación a Excel	117
Ilustración 168 - Hoja de Posible Phishing.....	117
Ilustración 169 - Primer fragmento de código del método emailSourcesToTable() de exportación a Excel	118
Ilustración 170 - Segundo fragmento de código del método emailSourcesToTable() de exportación a Excel	118
Ilustración 171 - Tercer fragmento de código del método emailSourcesToTable() de exportación a Excel	119
Ilustración 172 - Cuarto fragmento de código del método emailSourcesToTable() de exportación a Excel	119
Ilustración 173 - Quinto fragmento de código del método emailSourcesToTable() de exportación a Excel	119
Ilustración 174 - Hoja de Fuentes de localización de email con resultados de UPM	120
Ilustración 175 - Hoja de Fuentes de localización de email con resultados de ETSISI.....	120
Ilustración 176 - Primer fragmento de código del método metadataToTable() de exportación a Excel	121
Ilustración 177 - Segundo fragmento de código del método metadataToTable() de exportación a Excel	122
Ilustración 178 - Cuarto fragmento de código del método metadataToTable() de exportación a Excel	122
Ilustración 179 - Quinto fragmento de código del método metadataToTable() de exportación a Excel	122
Ilustración 180 - Sexto fragmento de código del método metadataToTable() de exportación a Excel.....	123
Ilustración 181 - Séptimo fragmento de código del método metadataToTable() de exportación a Excel..	123
Ilustración 182 - Octavo fragmento de código del método metadataToTable() de exportación a Excel ...	123
Ilustración 183 - Hoja de Metadatos	123
Ilustración 184 - Ejecución de Recon-vd.....	124
Ilustración 185 - Carpeta "modules"	125
Ilustración 186 - Carpeta "data"	125
Ilustración 187 - Ejecución del comando "modules reload"	126
Ilustración 188 - Introducción de API Key	126
Ilustración 189 - Ejecución del comando "workspaces create"	127
Ilustración 190 - Ejecución del comando "modules load"	127
Ilustración 191 - Ejecución del comando "modules search"	128
Ilustración 192 - Ejecución del comando "options list"	128
Ilustración 193 - Ejecución del comando "options set"	128
Ilustración 194 - Ejecución del comando "options unset"	128
Ilustración 195 - Definición del input del módulo de manera manual	129
Ilustración 196 - Ejecución del comando "input"	129
Ilustración 197 - Inserción de un dominio en Elasticsearch utilizando las "Dev Tools" de Kibana	129
Ilustración 198 - Inserción de nombres de marca en Elasticsearch utilizando Kibana.....	130
Ilustración 199 - Ejecución del módulo de Emails.....	130
Ilustración 200 - Ejecución del módulo de Emails.....	131
Ilustración 201 - Ejecución de Elasticsearch.....	132
Ilustración 202 - Ejecución de Kibana.....	132
Ilustración 203 - Ejecución de Kibana en el puerto 5601	132
Ilustración 204 - Pantalla inicial de Kibana	133
Ilustración 205 - Creando un "index pattern" en Kibana (paso 1)	133
Ilustración 206 - Creando un "index pattern" (paso 2)	133
Ilustración 207 - Pestaña Discover de Kibana	134
Ilustración 208 - Workpad realizado con Canvas.....	134
Ilustración 209 - Especificación de los datos a extraer de Elasticsearch para representar en el Workpad	135
Ilustración 210 - Especificación de los datos a mostrar en el Workpad	135

Listado de tablas

Tabla 1 - Matriz de trazabilidad de requisitos	4
Tabla 2 - Diccionario WBS	7
Tabla 3 - Actividades.....	10
Tabla 4 - Matriz de asignación de responsabilidades	13
Tabla 5 - Duración y holguras	16
Tabla 6 - Estimación de costos	21
Tabla 7 - Presupuesto del proyecto	22
Tabla 8 - Correspondencia entre terminología SQL y Elasticsearch	48
Tabla 9 - Estructura de los nuevos tipos de documentos de Elasticsearch	49
Tabla 10 - Estructura de los documentos de Elasticsearch correspondientes a las tablas de la base de datos SQL.....	50

Acrónimos

API	Application Programming Interface
CIA	Central Intelligence Agency
CRUD	Create, Read, Update and Delete
CSV	Comma Separated Value
DB	Database
DNS	Domain Name System
ELINT	Electronic Intelligence
FBIS	Foreign Broadcast Intelligence Service
FOCA	Fingerprinting Organizations with Collected Archives
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HUMINT	Human Intelligence
IMINT	Imagery Intelligence
IP	Internet Protocol
MASINT	Measurement and Signature Intelligence
MX	Mail Exchange
OSINT	Opens Source Intelligence
PGP	Pretty Good Privacy
PMBOK	A Guide to the Project Management Body of Knowledge
PTR	Pointer Records
REST	Representational State Transfer
SIGINT	Signal Intelligence
SQL	Structured Query Language
TELINT	Telemetry Intelligence
UI	User Interface
URL	Uniform Resource Locator

1

Introducción

1.1 Objetivo y motivación

Cuando hablamos de OSINT nos referimos a la búsqueda de información en fuentes abiertas, es decir, disponible de manera pública. La información expuesta de personas u organismos es especialmente sensible, ya que puede ser utilizada con el objetivo de llevar a cabo ciberataques que afecten a la confidencialidad, integridad y disponibilidad de sus activos, por lo que es conveniente tener un control del nivel de exposición al que te enfrentas.

En una era en la que se ha impuesto la automatización, ya no es viable realizar manualmente la recopilación y procesamiento de toda esta información que se encuentra en Internet. Por ello, es necesario el uso de herramientas que permitan llevar a cabo dichos procesos de manera automática, con el objetivo de centrar los esfuerzos en el análisis de esos datos.

Este proyecto se ha realizado para el servicio de Vigilancia Digital de la empresa en la que trabajamos, dentro del departamento de Ciberseguridad. La motivación del proyecto es precisamente automatizar la adquisición de los datos y procesarlos para su posterior análisis, una tarea que hasta ahora se realizaba de manera manual. Esto supone un ahorro de tiempo considerable, además de que permite mejorar los métodos utilizados para obtener una mayor cantidad de información de fuentes públicas.

Para ello desarrollaremos módulos del framework Recon-ng con los que se obtengan datos de diversas fuentes abiertas, utilizando la técnica de web scraping para extraer información de los sitios web y transformarla en datos estructurados. También se realizará un proceso de reingeniería, integrando Elasticsearch en dicho framework para almacenar los datos recopilados. Estos datos se visualizarán en tiempo real mediante el uso de Kibana y se podrán exportar en Excel de manera estructurada, para así facilitar la tarea de análisis.

1.2 Estructura del documento

El presente documento se divide en los siguientes apartados:

- Planificación del proyecto: este apartado incluye la obtención de los requisitos, a partir de los cuales se definen las tareas con sus respectivas actividades. También se incluye la matriz de asignación de responsabilidades junto con los diagramas EDT-WBS, PERT y Gantt. En último lugar se encuentra el presupuesto del proyecto.
- Estado del arte: en este apartado se aborda la definición de Inteligencia y sus tipos, profundizando en la evolución, la metodología y las herramientas de OSINT. Por otra parte, se realiza una breve comparación entre bases de datos SQL y NoSQL. A continuación, se estudian las herramientas para el análisis de datos en fuentes abiertas, centrándonos en Elasticsearch junto con Kibana.
- Reingeniería: en este apartado se describe el proceso de reingeniería, que incluye la ingeniería inversa, el plan de migración y la propia migración.
- Documentación del desarrollo: en este apartado se documenta el código de los módulos desarrollados.
- Guía de uso: este apartado incluye una breve guía del framework Recon-ng, Elasticsearch y Kibana.
- Conclusiones: en este apartado se realiza una evaluación final del proyecto tras haber puesto en funcionamiento el software en nuestra empresa.

2 Planificación

2.1 Matriz de trazabilidad de requisitos

ID	ID de Asociado	Descripción de los Requisitos	EDT
1. Requisitos del proyecto	1.0	En primer lugar, se hará una planificación del proyecto que formará parte de la memoria	1.1
	1.1	La memoria ha de tratar todos los epígrafes de obligado cumplimiento	1.2
	1.2	El proyecto deberá tener la conformidad del tutor profesional y del tutor del proyecto	1.4
2. Requisitos funcionales	2.0	El framework dispondrá de un módulo que adquiera todas las direcciones de correo que se encuentren en los resultados de buscadores para un dominio dado	2.4
	2.1	El framework dispondrá de un módulo que realice una búsqueda de direcciones de correo con el servicio de Hunter	2.4
	2.2	El framework dispondrá de un módulo que adquiera todas las noticias indexadas en buscadores	2.4
	2.3	El framework dispondrá de un módulo que realice una búsqueda de coincidencias en servicios de compartición de texto online	2.4
	2.4	El framework dispondrá de un módulo que realice una búsqueda de menciones en foros	2.4
	2.5	El framework dispondrá de un módulo que compruebe si el servidor de correo correspondiente a un dominio está en listas negras de spam	2.4
	2.6	El framework dispondrá de un módulo que realice una búsqueda de dominios similares con host que puedan ser maliciosos	2.4
	2.7	El framework dispondrá de un módulo que adquiera documentos que se encuentren en buscadores y extraiga sus metadatos	2.4
	2.8	La información obtenida con los módulos del framework será exportable en Excel siguiendo el formato de los informes corporativos	2.6

	2.9	Los datos indexados en Elasticsearch se visualizarán con Kibana	2.7
3. Requisitos no funcionales	3.0	Se hará uso del framework recon-ng para automatizar las tareas de adquisición de datos	2.2
	3.1	La información se obtendrá de fuentes abiertas	2.3
	3.3	Los módulos estarán desarrollados en Python	2.1
	3.4	Los módulos a desarrollar serán compatibles con el framework	2.4
	3.5	Los datos obtenidos con los módulos del framework se guardarán en una base de datos NoSQL con Elasticsearch	2.4 2.5

Tabla 1 - Matriz de trazabilidad de requisitos

2.2 EDT/WBS

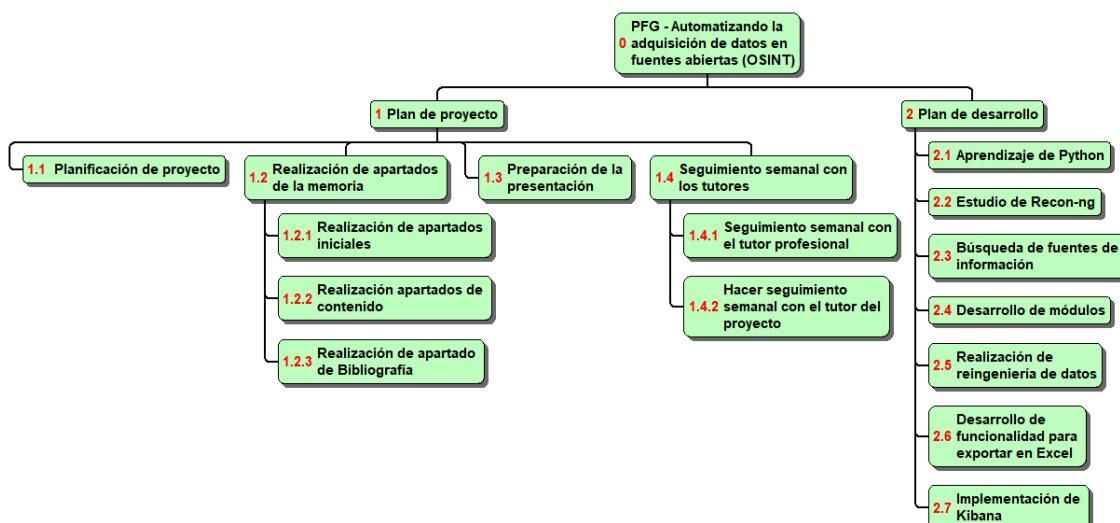


Ilustración 1 - EDT/WBS

2.3 Diccionario WBS

TAREAS		DESCRIPCIÓN
1. Plan de proyecto	1.1. Planificación del proyecto	<ul style="list-style-type: none"> Hacer un brainstorming para sacar los requisitos de negocio, proyecto, funcionales y no funcionales. Descomposición de los requisitos en tareas y actividades. <ul style="list-style-type: none"> PERT Cronograma Presupuesto

		<p>1.2.1. Realización de apartados iniciales</p>	<ul style="list-style-type: none"> • Agradecimientos • Resumen: Destacar los aspectos y resultados más relevantes del trabajo realizado, con una extensión de entre 150 y 300 palabras. • Abstract • Índice general • Índice de figuras • Índice de tablas • Acrónimos
<p>1.2. Realización de apartados de la memoria</p>		<p>1.2.2. Realización de apartados de contenido</p>	<ul style="list-style-type: none"> • Introducción: Objetivo, motivación y estructura del documento. • Estado del arte: OSINT y bases de datos SQL y NoSQL. • Planificación y costes • Desarrollo: Documentar los módulos desarrollados y el proceso de reingeniería. También realizar un manual del software. • Conclusiones • Responsabilidad social y legal
		<p>1.2.3. Realización de apartado de Bibliografía</p>	<p>A partir de las fuentes de consulta realizar la bibliografía con el formato de referencias de IEEE.</p>
<p>2. Plan de desarrollo</p>		<p>1.3. Preparación de la presentación</p>	<p>Preparar las diapositivas y el guion de la presentación.</p>
	<p>1.4. Seguimiento semanal con los tutores</p>	<p>1.4.1. Seguimiento semanal con el tutor profesional</p>	<p>Informar al tutor profesional sobre el avance del proyecto.</p>
		<p>1.4.2. Seguimiento semanal con el tutor del proyecto</p>	<p>Asistir a reuniones con el tutor para revisar el progreso del proyecto.</p>
		<p>2.1. Aprendizaje de Python</p>	<p>Aprender los conceptos necesarios de Python para realizar el desarrollo de los módulos de recon-ng y la reingeniería.</p>
		<p>2.2. Estudio de Recon-ng</p>	<p>Estudiar el funcionamiento del</p>

		framework junto con los módulos disponibles en el marketplace de Recon-ng.
	2.3. Búsqueda de fuentes de información	Buscar fuentes de las que obtener los datos en Internet.
2.4. Desarrollo de módulos	Desarrollo de módulo de adquisición de direcciones de correo en resultados de buscadores	Desarrollar un módulo que adquiera direcciones de correo electrónico en las páginas web y documentos PDF de los resultados de búsqueda en Google para un dominio dado.
	Desarrollo de módulo de búsqueda de direcciones de correo con Hunter	Desarrollar un módulo que adquiera direcciones de correo electrónico usando la API de Hunter.
	Desarrollo de módulo de adquisición de noticias	Desarrollar un módulo que adquiera las noticias que aparezcan en la sección de noticias de Google para un nombre de marca dado.
	Desarrollo de módulo de búsqueda de coincidencias en servicios de compartición de texto online	Desarrollar un módulo que adquiera resultados en Pastebin y GitHub Gist que contengan un nombre de marca.
	Desarrollo de módulo de búsqueda de menciones en foros	Desarrollar un módulo que busque menciones de nombres de marca en los foros ElOtroLado, Reddit y Forocoches.
	Desarrollo de módulo de comprobación de dominios en listas negras	Desarrollar un módulo que dado un dominio comprueba la aparición de sus registros MX en listas negras de spam.
	Desarrollo de módulo de búsqueda de dominios similares que puedan ser maliciosos	Desarrollar un módulo que obtenga dominios similares a un dominio dado realizando permutaciones de este con dnstwister, comprobando que dichos dominios similares tengan página web. En tal caso, analizarlos en VirusTotal y guardar

			una captura en Archive.org.
		Desarrollo de módulo de adquisición de documentos en buscadores y extracción de sus metadatos	Desarrollar un módulo que dado un dominio adquiera distintos tipos de documentos que se encuentren en dicho dominio, extrayendo los metadatos de los PDF.
		2.5. Realización de reingeniería	Realizar una reingeniería para migrar la base de datos SQL (SQLite) de Recon-ng a NoSQL (Elasticsearch), adaptando los componentes del framework que interactúan con la base de datos.
		2.6. Desarrollo de funcionalidad para exportar en Excel	Desarrollar el módulo que extraiga de Elasticsearch los datos adquiridos con los módulos desarrollados, exportándolos a Excel con el formato de los informes corporativos.
		2.7. Implementación de Kibana	Utilizar Kibana para la visualización de los datos indexados en Elasticsearch, diseñando un dashboard.

Tabla 2 - Diccionario WBS

2.4 Actividades

TAREAS	ACTIVIDADES	ID	ORDEN	TIEMPO PROBABLE (h)	TIEMPO OPTIMISTA	TIEMPO PESIMISTA	$t(i, j) = (a + 4m + b) / 6$
1.1 Planificación del proyecto	- Obtención de requisitos	1.1.A01	1.5	10	8,5	11,5	10
	- Descomponer requisitos en tareas y actividades	1.1.A02	1.6	15	12,75	17,25	15
	- Hacer PERT y cronograma	1.1.A03	1.7	20	17	23	20
	- Hacer presupuesto	1.1.A04	1.8	10	8,5	11,5	10
1.2.1	- Realizar apartado de Agradecimientos	1.2.1.A01	1.24.1	0,25	0,2125	0,2875	0,25

Realización de apartados iniciales	- Realizar apartado de Resumen	1.2.1.A02	1.24.2	5	4,25	5,75	5
	- Realizar apartado de Abstract	1.2.1.A03	1.25.1	3	2,55	3,45	3
	- Realizar apartado de Índice general	1.2.1.A04	1.25.2	0,5	0,425	0,575	0,5
	- Realizar apartado de Índice de figuras	1.2.1.A05	1.25.3	0,5	0,425	0,575	0,5
	- Realizar apartado de Índice de tablas	1.2.1.A06	1.25.4	0,5	0,425	0,575	0,5
	- Realizar apartado de Acrónimos	1.2.1.A07	1.25.5	2	1,7	2,3	2
	- Realizar apartado de Introducción	1.2.2.A01	1.20.1	15	12,75	17,25	15
1.2.2 Realización de apartados de contenido	- Realizar apartado de Estado del arte	1.2.2.A02	1.20.2	75	63,75	86,25	75
	- Realizar apartado de Planificación y costes	1.2.2.A03	1.9	6	5,1	6,9	6
	- Realizar apartado de Desarrollo	1.2.2.A04	1.19	60	51	69	60
	- Realizar apartado de Conclusiones	1.2.2.A05	1.21	12	10,2	13,8	12
	- Realizar apartado de Responsabilidad social y legal	1.2.2.A06	1.22	8	6,8	9,2	8
1.2.3 Realización de apartado de Bibliografía	- Realizar apartado de Bibliografía	1.2.3.A01	1.23	5	4,25	5,75	5
1.3 Preparar la presentación	- Preparar las diapositivas	1.3.A01	1.26	25	21,25	28,75	25
	- Preparar la lectura	1.3.A02	1.27	25	21,25	28,75	25
1.4 Seguimiento semanal con el tutor profesional	- Hacer seguimiento semanal con el tutor profesional	1.4.A01	1.1.1	4,75 (15 min. por semana)	4,0375	5,4625	4,75
1.5 Revisiones periódicas con el tutor del proyecto	- Hacer seguimiento semanal con el tutor del proyecto	1.5.A01	1.1.2	4,75 (15 min. por semana)	4,0375	5,4625	4,75
2.1 Aprendizaje de Python	- Estudiar lo esencial de Python	2.1.A01	1.2	15	12,75	17,25	15

	- Estudiar automatización con Python	2.1.A02	1.3	35	29,75	40,25	35
2.2 Estudio de Recon-ng	- Realizar un estudio de Recon-ng	2.2.A01	1.4.1	25	21,25	28,75	25
2.3 Búsqueda de fuentes de información	- Realizar búsqueda de fuentes de información	2.3.A01	1.4.2	5	4,25	5,75	5
2.4 Desarrollo de módulos	- Desarrollar módulo de adquisición de direcciones de correo en resultados de buscadores	2.4.A01	1.10.1	25	21,25	28,75	25
	- Desarrollar módulo de adquisición de direcciones de correo con Hunter	2.4.A02	1.12	15	12,75	17,25	15
	- Desarrollar módulo de adquisición de noticias	2.4.A03	1.11.1	10	8,5	11,5	10
	- Desarrollar módulo de búsqueda de coincidencias en servicios de compartición de texto online	2.4.A04	1.11.2	25	21,25	28,75	25
	- Desarrollar módulo de búsqueda de menciones en foros	2.4.A05	1.13.1	25	21,25	28,75	25
	- Desarrollar módulo de comprobación de dominios en listas negras	2.4.A06	1.10.2	25	21,25	28,75	25
	- Desarrollar módulo de búsqueda de dominios similares con host que puedan ser maliciosos	2.4.A07	1.13.2	30	25,5	34,5	30
	- Desarrollar módulo de adquisición de documentos en buscadores y extracción de sus metadatos	2.4.A08	1.14	10	8,5	11,5	10
2.5 Realización de reingeniería de datos	- Realizar ingeniería inversa de la base de datos	2.5.A01	1.15	10	8,5	11,5	10

	-Implementar Elasticsearch en el framework	2.5.A02	1.16.1	50	42,5	57,5	50
2.6 Desarrollo de funcionalidad para exportar en Excel	- Desarrollar la funcionalidad de exportar en Excel	2.6.A01	1.16.2	40	34	46	40
2.7 Implementación de Kibana	- Instalar y configurar Kibana	2.7.A01	1.17	5	4,25	5,75	5
	- Diseñar dashboard con Kibana	2.7.A02	1.18	15	12,75	17,25	15

Tabla 3 - Actividades

2.5 Matriz de asignación de responsabilidades

Roles RASCI:

R → Responsable A → Aprobador S → Apoyo C → Consultado I → Informado

Actividades	Miembros del Equipo de Proyecto		Otros Stakeholders	
	Rubén	Rodrigo	Tutor Profesional	Tutor del Proyecto
1.1.A01 Obtención de requisitos	R	R	A	A
1.1.A02 Descomponer requisitos en tareas y actividades	R	R	I	A
1.1.A03 Hacer PERT y cronograma	R	R	A	I
1.1.A03 Hacer presupuesto	R	R	A	I
1.2.1.A01 Realizar apartado de Agradecimientos	R	R		A
1.2.1.A02 Realizar apartado de Resumen	R	R		A
1.2.1.A03 Realizar apartado de Abstract	R	R		A
1.2.1.A04 Realizar apartado de Índice general	R	R		A

1.2.1.A05 Realizar apartado de Índice de figuras	R	R		A
1.2.1.A06 Realizar apartado de Índice de tablas	R	R		A
1.2.1.A07 Realizar apartado de Acrónimos	R	R		A
1.2.2.A01 Realizar apartado de Introducción	R	R		A
1.2.2.A02 Realizar apartado de Estado del arte	R	R		A
1.2.2.A03 Realizar apartado de Planificación y costes	R	R		A
1.2.2.A04 Realizar apartado de Desarrollo	R	R		A
1.2.2.A05 Realizar apartado de Conclusiones	R	R		A
1.2.2.A08 Realizar apartado de Responsabilidad social y legal	R	R		A
1.2.3.A01 Realizar apartado de Bibliografía	R	R		A
1.3.A01 Preparar las diapositivas	R	R		A
1.3.A02 Preparar la lectura	R	R		A
1.4.A01 Programar revisiones periódicas con el tutor del proyecto	R	R		A
1.5.A01 Programar revisiones periódicas con el tutor profesional	R	R	A	
2.1.A01	R	R		A

Estudiar lo esencial de Python				
2.1.A02 Estudiar automatización con Python	R	R		A
2.2.A01 Realizar un estudio de Recon-ng	R	R		A
2.3.A01 Realizar investigación de fuentes de información	R	R		A
2.4.A01 Desarrollar módulo de adquisición de direcciones de correo en resultados de buscadores	R		A	I
2.4.A02 Desarrollar módulo de adquisición de direcciones de correo con Hunter	R		A	I
2.4.A03 Desarrollar módulo de adquisición de noticias	R		A	I
2.4.A04 Desarrollar módulo de búsqueda de coincidencias en servicios de compartición de texto online		R	A	I
2.4.A05 Desarrollar módulo de búsqueda de menciones en foros		R	A	I
2.4.A06 Desarrollar módulo de comprobación de dominios en listas negras		R	A	I
2.4.A07 Desarrollar módulo de búsqueda de	R		A	I

dominios similares con host que puedan ser maliciosos				
2.4.A08 Desarrollar módulo de adquisición de documentos en buscadores y extracción de sus metadatos	R		A	I
2.5.A01 Realizar ingeniería inversa de la base de datos	R	R	A	I
2.5.A02 Implementar Elasticsearch en el framework	R	S	A	I
2.6.A01 Desarrollar la funcionalidad de exportar en Excel	S	R	A	I
2.7.A01 Instalar y configurar Kibana	R	R	A	I
2.7.A02 Diseñar dashboard con Kibana	R	R	A	I

Tabla 4 - Matriz de asignación de responsabilidades

2.6 Diagrama de red del cronograma del proyecto (PERT)

2.6.1 PERT de actividades

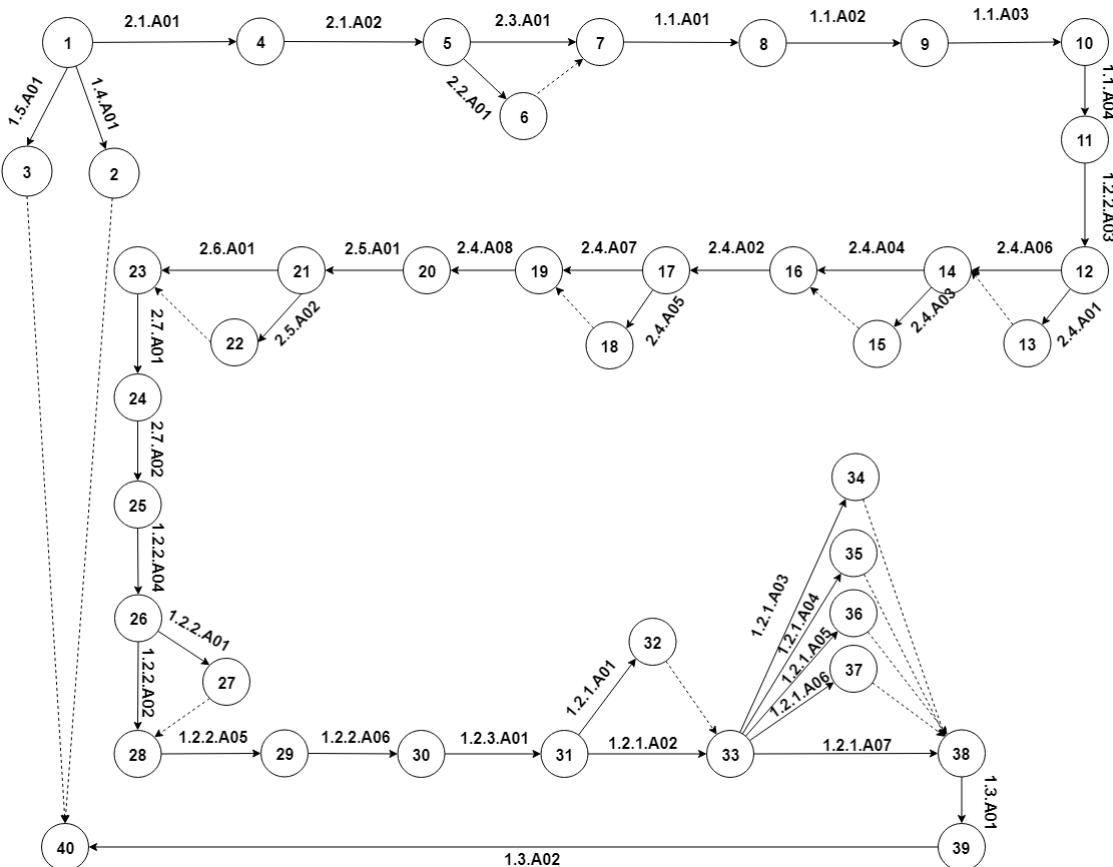


Ilustración 2 - PERT de actividades

2.6.2 Duración y holguras

En esta tabla relacionamos las actividades, con su duración asignada. Se puede apreciar el instante más bajo y alto en el que pueden empezar y acabar, así como su holgura.

ACTIVIDADES			DURACIÓN	TIEMPO MÁS BAJO		TIEMPO MÁS ALTO		HOLGURA		
NODO INICIAL	NODO FINAL	ACTIVIDAD	DURACIÓN t(i,j)	INICIACIÓN ES(i)	TERMINACIÓN EF(j)	INICIACIÓN LS(i)	TERMINACIÓN LF(j)	TOTAL	LIBRE	INDEPENDIENTE
1	2	1.4.A01	4,75	0	4,75	0	499	494,25	0	0
1	3	1.5.A01	4,75	0	4,75	0	499	494,25	0	0
1	4	2.1.A01	15	0	15	0	15	0	0	0

2	40	Fl	0	4,75	539	15	15	10,25	534,25	524
3	40	Fl	0	4,75	539	15	15	10,25	534,25	524
4	5	2.1.A02	35	15	50	15	50	0	0	0
5	6	2.2.A01	25	50	75	50	75	0	0	0
5	7	2.3.A01	5	50	55	50	75	20	0	0
6	7	Fl	0	55	75	75	75	20	20	0
7	8	1.1.A01	10	75	85	75	85	0	0	0
8	9	1.1.A02	15	85	100	85	100	0	0	0
9	10	1.1.A03	20	100	120	100	120	0	0	0
10	11	1.1.A04	10	120	130	120	130	0	0	0
11	12	1.2.2.A03	6	130	136	130	136	0	0	0
12	13	2.4.A01	25	136	161	136	161	0	0	0
12	14	2.4.A06	25	136	161	136	161	0	0	0
13	14	Fl	0	161	161	161	161	0	0	0
14	15	2.4.A03	10	161	171	161	186	15	0	0
14	16	2.4.A04	25	161	186	161	186	0	0	0
15	16	Fl	0	171	186	186	186	15	15	0
16	17	2.4.A02	15	186	201	186	201	0	0	0
17	18	2.4.A05	25	201	226	201	231	5	0	0
17	19	2.4.A07	30	201	231	201	231	0	0	0
18	19	Fl	0	226	231	231	231	5	5	0
19	20	2.4.A08	10	231	241	231	241	0	0	0
20	21	2.5.A01	10	241	251	241	251	0	0	0
21	22	2.5.A02	50	251	301	251	301	0	0	0
21	23	2.6.A01	40	251	291	251	301	10	0	0
22	23	Fl	0	291	301	301	301	10	10	0
23	24	2.7.A01	5	301	306	301	306	0	0	0
24	25	2.7.A02	15	306	321	306	321	0	0	0
25	26	1.2.2.A04	60	321	381	321	381	0	0	0
26	27	1.2.2.A01	15	381	396	381	456	60	0	0

26	28	1.2.2.A02	75	381	456	381	456	0	0	0
27	28	FI	0	396	456	456	456	60	60	0
28	29	1.2.2.A05	12	456	468	456	468	0	0	0
29	30	1.2.2.A06	8	468	476	468	476	0	0	0
30	31	1.2.3.A01	5	476	481	476	481	0	0	0
31	32	1.2.1.A01	0,25	481	481,25	481	486	4,75	0	0
31	33	1.2.1.A02	5	481	486	481	486	0	0	0
32	33	FI	0	481,25	486	486	486	4,75	4,75	0
33	34	1.2.1.A03	3	486	489	486	489	0	0	0
33	35	1.2.1.A04	0,5	486	486,5	486	489	2,5	0	0
33	36	1.2.1.A05	0,5	486	486,5	486	489	2,5	0	0
33	37	1.2.1.A06	0,5	486	486,5	486	489	2,5	0	0
33	38	1.2.1.A07	2	486	488	486	489	1	0	0
34	38	FI	0	486,5	489	489	489	2,5	2,5	0
35	38	FI	0	486,5	489	489	489	2,5	2,5	0
36	38	FI	0	486,5	489	489	489	2,5	2,5	0
37	38	FI	0	489	489	489	489	0	0	0
38	39	1.3.A01	25	489	514	489	514	0	0	0
39	40	1.3.A02	25	514	539	514	539	0	0	0

Tabla 5 - Duración y holguras

2.6.3 Camino crítico

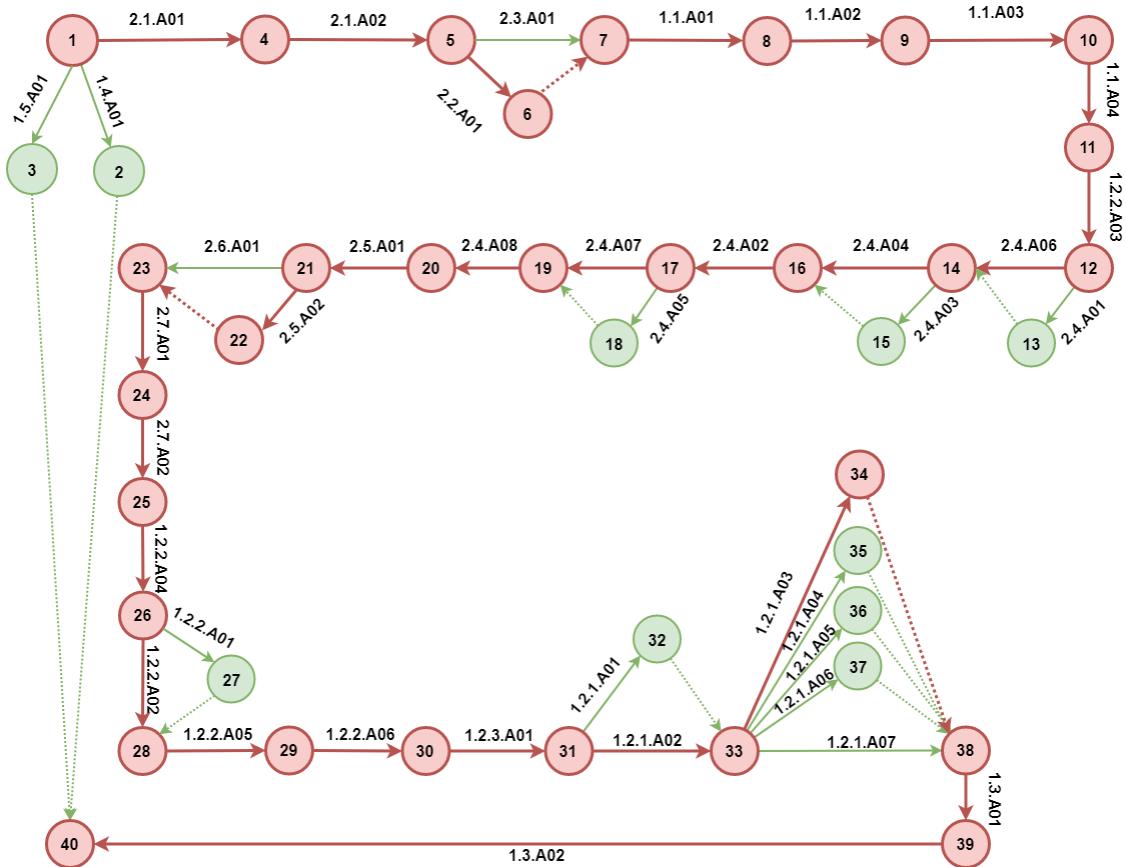


Ilustración 3 - Camino crítico

2.7 Cronograma del proyecto

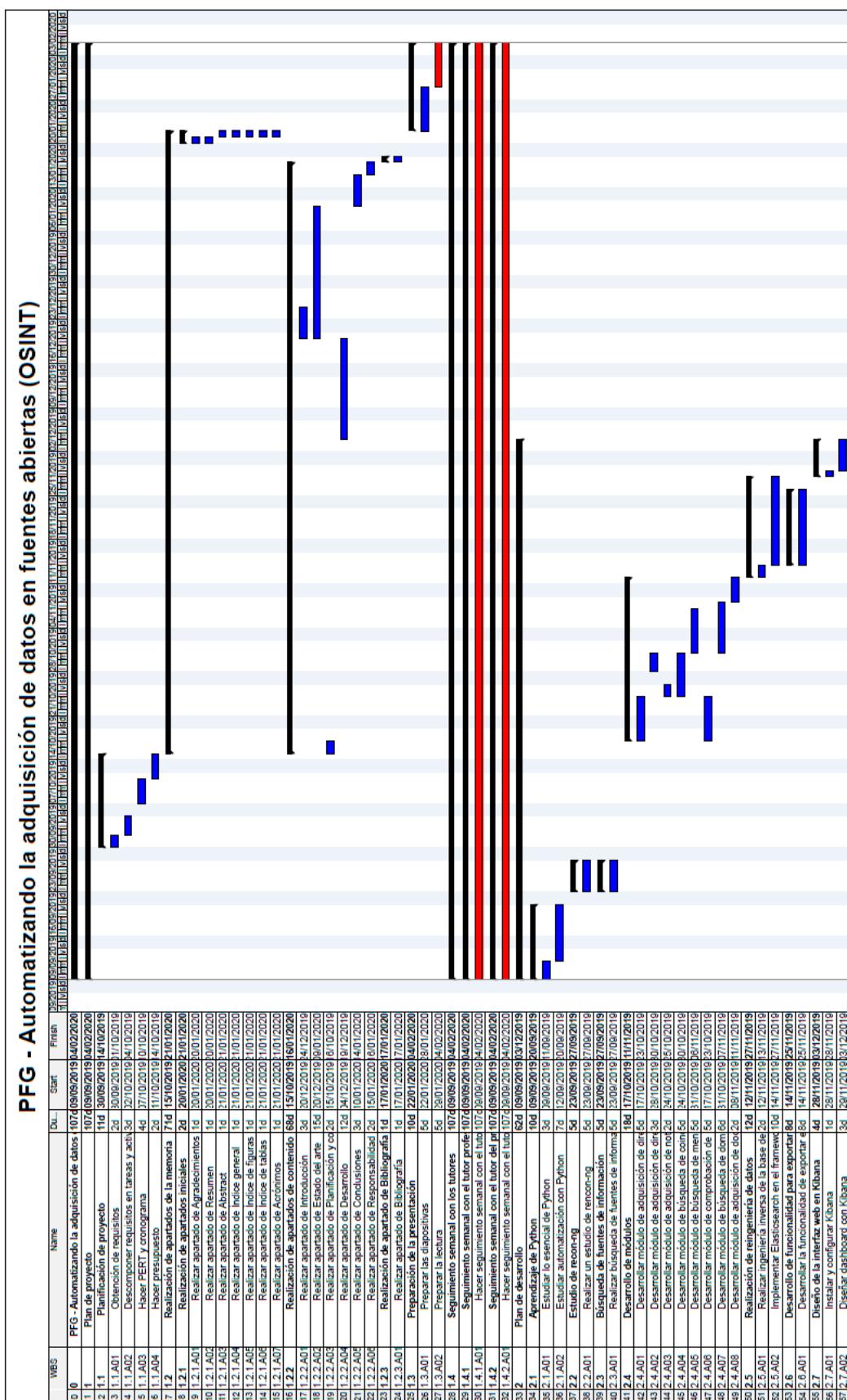


Ilustración 4 - Cronograma del proyecto

2.8 Plan de gestión de costos

En este apartado se definen ciertas pautas en las que nos basamos a la hora de realizar la estimación de los costos del proyecto.

- El nivel de precisión será de $\pm 15\%$ para el costo pesimista y optimista.
- La unidad de medida para los costos será €.
- El costo probable será de 5€/h.
- Se establecerá un control de costes que evaluará el impacto de cualquier posible cambio en el costo.
 - Toda variación dentro del $\pm 15\%$ del presupuesto será considerada como normal.
 - Una solicitud de cambio sobre el costo del proyecto que exceda el $\pm 15\%$ del presupuesto del proyecto deberá ser revisada y aprobada por el tutor profesional.

2.8.1 Estimación de costos

TAREAS	ACTIVIDADES	ID	COSTO PROBABLE (€)	COSTO OPTIMISTA	COSTO PESIMISTA	$cE = (cO + 4cM + cP) / 6$
1.1 Planificación del proyecto	- Obtención de requisitos	1.1.A01	50	42,5	57,5	50
	- Descomponer requisitos en tareas y actividades	1.1.A02	75	63,75	86,25	75
	- Hacer PERT y cronograma	1.1.A03	100	85	115	100
	- Hacer presupuesto	1.1.A04	50	42,5	57,5	50
1.2.1 Realización de apartados iniciales	- Realizar apartado de Agradecimientos	1.2.1.A01	1,25	1,0625	1,4375	1,25
	- Realizar apartado de Resumen	1.2.1.A02	25	21,25	28,75	25
	- Realizar apartado de Abstract	1.2.1.A03	15	12,75	17,25	15
	- Realizar apartado de Índice general	1.2.1.A04	2,5	2,125	2,875	2,5
	- Realizar apartado de Índice de figuras	1.2.1.A05	2,5	2,125	2,875	2,5
	- Realizar apartado de Índice de tablas	1.2.1.A06	2,5	2,125	2,875	2,5
	- Realizar apartado de Acrónimos	1.2.1.A07	10	8,5	11,5	10
1.2.2 Realización de apartados de contenido	- Realizar apartado de Introducción	1.2.2.A01	75	63,75	86,25	75
	- Realizar apartado de Estado del arte	1.2.2.A02	375	318,75	431,25	375

	- Realizar apartado de Planificación y costes	1.2.2.A03	30	25,5	34,5	30
	- Realizar apartado de Desarrollo	1.2.2.A04	300	255	345	300
	- Realizar apartado de Conclusiones	1.2.2.A05	60	51	69	60
	- Realizar apartado de Responsabilidad social y legal	1.2.2.A06	40	34	46	40
1.2.3 Realización de apartado de Bibliografía	- Realizar apartado de Bibliografía	1.2.3.A01	25	21,25	28,75	25
1.3 Preparar la presentación	- Preparar las diapositivas	1.3.A01	125	106,25	143,75	125
	- Preparar la lectura	1.3.A02	125	106,25	143,75	125
1.4 Seguimiento semanal con el tutor profesional	- Hacer seguimiento semanal con el tutor profesional	1.4.A01	23,75	20,1875	27,3125	23,75
1.5 Revisiones periódicas con el tutor del proyecto	- Hacer seguimiento semanal con el tutor del proyecto	1.5.A01	23,75	20,1875	27,3125	23,75
2.1 Aprendizaje de Python	- Estudiar lo esencial de Python	2.1.A01	75	63,75	86,25	75
	- Estudiar automatización con Python	2.1.A02	175	148,75	201,25	175
2.2 Estudio de Recon-ng	- Realizar un estudio de recon-ng	2.2.A01	125	106,25	143,75	125
2.3 Búsqueda de fuentes de información	- Realizar búsqueda de fuentes de información	2.3.A01	25	21,25	28,75	25
2.4 Desarrollo de módulos	- Desarrollar módulo de adquisición de direcciones de correo en resultados de buscadores	2.4.A01	125	106,25	143,75	125
	- Desarrollar módulo de adquisición de direcciones de correo con Hunter	2.4.A02	75	63,75	86,25	75
	- Desarrollar módulo de adquisición de noticias	2.4.A03	50	42,5	57,5	50
	- Desarrollar módulo de búsqueda de coincidencias en servicios de compartición de texto online	2.4.A04	125	106,25	143,75	125

	- Desarrollar módulo de búsqueda de menciones en foros	2.4.A05	125	106,25	143,75	125
	- Desarrollar módulo de comprobación de dominios en listas negras	2.4.A06	125	106,25	143,75	125
	- Desarrollar módulo de búsqueda de dominios similares con host que puedan ser maliciosos	2.4.A07	150	127,5	172,5	150
	- Desarrollar módulo de adquisición de documentos en buscadores y extracción de sus metadatos	2.4.A08	50	42,5	57,5	50
2.5 Realización de reingeniería de datos	- Realizar ingeniería inversa de la base de datos	2.5.A01	50	42,5	57,5	50
	- Implementar Elasticsearch en el framework	2.5.A02	250	212,5	287,5	250
2.6 Desarrollo de funcionalidad para exportar en Excel	- Desarrollar la funcionalidad de exportar en Excel	2.6.A01	200	170	230	200
2.7 Implementación de Kibana	- Instalar y configurar Kibana	2.7.A01	25	21,25	28,75	25
	- Diseñar dashboard con Kibana	2.7.A02	75	63,75	86,25	75

Tabla 6 - Estimación de costos

2.9 Presupuesto del proyecto

FASE	ENTREGABLE	Coste Total	COSTE FASE
1. Plan de proyecto	1.1 Planificación del proyecto	275	1536,25
	1.2.1 Realización de apartados iniciales	58,75	
	1.2.2 Realización de apartados de contenido	880	
	1.2.3 Realización de apartado de Bibliografía	25	
	1.3 Preparar la presentación	250	
	1.4 Seguimiento semanal con el tutor profesional	23,75	
	1.5 Revisiones periódicas con el tutor del proyecto	23,75	
2. Plan de desarrollo	2.1 Aprendizaje de Python	250	1875
	2.2 Estudio de Recon-ng	125	
	2.3 Búsqueda de fuentes de información	25	
	2.4 Desarrollo de módulos	875	
	2.5 Realización de reingeniería de datos	300	
	2.6 Desarrollo de funcionalidad para exportar en Excel	200	
	2.7 Implementación de Kibana	100	
TOTAL FASES			3411,25
Reserva de Gestión			341,13
PRESUPUESTO DEL PROYECTO			3752,38

Tabla 7 - Presupuesto del proyecto

3 Estado del arte

3.1 Inteligencia

La Inteligencia se define como producto que resulta de la evaluación, la integración, el análisis y la interpretación de la información reunida por un servicio de inteligencia. Su elaboración es objeto del proceso conocido como ciclo de inteligencia. Esto quiere decir que el tratamiento de un dato aporte valor sobre algo concreto, siendo transformado a través del Ciclo de Inteligencia para generar así el producto de inteligencia. [1]

Los tipos principales de inteligencia se dividen en función de la naturaleza de la fuente de la que se obtienen:

- **HUMINT (Human Intelligence / Inteligencia humana):** la información es extraída de los humanos y sus relaciones. La recolección puede hacerse abiertamente, como cuando un organismo público de seguridad entrevista a testigos o sospechosos, o puede hacerse por medios clandestinos o encubiertos espionaje.
 - **SIGINT (Signals Intelligence / Inteligencia de Señales):** extrae la información de transmisiones electrónicas que pueden ser recogidas por barcos, aviones, instalaciones terrestres o satélites.
 - **IMINT (Image Intelligence / Inteligencia de Imágenes):** también conocida como inteligencia fotográfica (PHOTINT), puede ser obtenida por barcos, aviones, instalaciones terrestres o satélites.
 - **GEOINT (Geoespacial Intelligence / Inteligencia Geoespacial):** es el análisis y la representación visual de las actividades relacionadas con la seguridad en la tierra. Se produce mediante la integración de imágenes, inteligencia de imágenes e información geoespacial.
- MASINT (Measurement and Signature Intelligence / Inteligencia de medición y firmas):** es una disciplina de recopilación relativamente poco

conocida que se refiere a las capacidades armamentísticas y las actividades industriales. MASINT incluye el procesamiento y uso avanzado de los datos recogidos de los sistemas de recopilación de IMINT y SIGINT aéreos. También se apoya en la Inteligencia Telemétrica (TELINT) que se utiliza a veces para indicar los datos transmitidos por las armas durante las pruebas, y la Inteligencia Electrónica (ELINT) que muestra las emisiones electrónicas recogidas de las armas modernas y los sistemas de seguimiento. Tanto TELINT como ELINT pueden ser tipos de SIGINT y contribuir a MASINT. [2]

- **OSINT (Open Source Intelligence / Inteligencia de fuentes abiertas):** hace referencia al conocimiento recopilado a partir de fuentes de acceso público. [3]

3.2 OSINT

En 2001 el Allied Joint Intelligence de la OTAN, documento AJP-2.0.2001 definía OSINT como: “La inteligencia derivada de una amplia gama de recursos abiertos, como la radio, la televisión, los periódicos, los libros; a los que el público tiene acceso”.

El National Defense Authorization Act for Fiscal Year 2006, Public Law 109-163, sección 931 de Estados Unidos definía OSINT como “[...] la inteligencia que se produce a través de información disponible para el público, que es obtenida, explotada y diseminada en el tiempo y para la audiencia apropiada, a fin de satisfacer una petición de inteligencia concreta”

En el Handbook of Intelligence Studies, publicado en 2006 se definía así: “[...] información que no está clasificada, que ha sido intencionadamente descubierta, separada, tamizada y diseminada a una audiencia seleccionada a fin de responder a una pregunta específica.”

OSINT es una ramificación de la inteligencia que no está tan delimitada como las mostradas anteriormente y por eso su definición no es tan precisa. En lugar de clasificarse por su medio o técnica de obtención, es clasificada por su posibilidad de acceso. Por ello en determinados casos, HUMINT, SIGINT o IMINT puede ser también OSINT. [4]

Una ventaja de OSINT es su accesibilidad, aunque la gran cantidad de información disponible puede hacer difícil saber qué es lo que tiene valor. Determinar la fuente de los datos y su fiabilidad también puede ser complicado. Por lo tanto, los datos de OSINT aún requieren revisión y análisis para ser útiles a los políticos. [5]

El proceso para generar este tipo de inteligencia incluye la búsqueda, selección y adquisición de la información, así como un posterior procesado y análisis de la misma con el fin de obtener conocimiento útil y aplicable en distintos ámbitos. [3]

3.2.1 Evolución de OSINT

Si nos centramos en su esencia, OSINT lleva existiendo más tiempo del que puede parecer. En la época de los romanos ya había foros y publicaciones en las plazas en las que uno podía enterarse del panorama general. [4]

Sin embargo, este tipo de OSINT tenía una motivación muy diferente a la del OSINT moderno que conocemos.

Como muchos de los avances que tenemos hoy en día, su motivación fue meramente militar. Los gobiernos buscaban generar inteligencia que pudiera suponer una ventaja significativa en la guerra (número de tropas enemigas, puntos débiles, momento de la próxima ofensiva...), por ello se invirtieron recursos para la creación de organismos especializados.

El 26 de febrero de 1941 se creó el Foreign Broadcast Monitoring Service por parte de los Estados Unidos durante la segunda guerra mundial. El fin de este organismo era monitorizar y analizar los medios extranjeros, más concretamente los medios de las potencias del eje (Roma, Berlín, Tokio) [6]. Tras el ataque a Pearl Harbor este departamento tuvo un gran impulso.

En 1947 fue rebautizado como Foreign Broadcast Intelligence Service (FBIS) bajo la dirección de la recién creada CIA.

Años después, durante la guerra fría países pertenecientes a ambos bandos desarrollaron sus propios equipos de OSINT, a menudo bajo el mando de los servicios secretos. Las fuentes abiertas suponían la mayor parte de la inteligencia, según el analista de la CIA Stephen Mercado, convirtiéndose en la principal fuente de información sobre los enemigos.

Hasta la llegada de Internet, la mayor cantidad de recursos se destinaba a la extracción y el procesamiento de los datos. Ya que prácticamente la totalidad de la información se extraía de libros revistas y periódicos con el esfuerzo logístico y monetario que eso conllevaba. Por ejemplo, el Ministerio de Seguridad del Estado de la Alemania Oriental (MfS, conocido como la "Stasi") analizó 1.000 revistas occidentales y 100 libros al mes, a la vez que resumía más de 100 periódicos y 12 horas de emisiones diarias de radio y televisión de la Alemania occidental. [7]

Fue a lo largo de los años 90 cuando se fraguó la gran revolución que hemos experimentado en el siglo XXI. La aparición y la popularización de las computadoras, unidas a esa conexión global llamada “Internet”, abrieron un nuevo mundo de posibilidades al OSINT. Los documentos empezaban a digitalizarse y a poder consultarse fácil y rápidamente. Paralelamente se empezaron a crear foros que tenían un importante soporte en Internet: desde foros y redes sociales que facilitaban el acceso a opiniones de auténticos expertos, pasando por las viejas radios, televisiones o diarios que también eran consultables. [4]

Internet supuso un cambio en el paradigma, y ya el problema no era tanto la obtención sino la fiabilidad y legitimidad de las fuentes. La mayoría de las comunidades de inteligencia tardaron en apreciar el valor de Internet por dos razones:

- (1) Los organismos de inteligencia buscan una ventaja informativa mediante el trato de información secreta obtenida de manera clandestina. Confiar en la información de fuentes abiertas y en las restricciones de copyright iba en contra de estas ideas.
- (2) En la mayoría de los casos es más difícil, arriesgado y costoso aplicar métodos clandestinos para adquirir fuentes secretas, dando así la impresión de que esas fuentes deben ser de mayor valor que las fuentes abiertas, confundiendo el método con el producto o confundiendo el secreto con el conocimiento. [7] [8]

El ejército de los Estados Unidos acuñó el término OSINT a finales del decenio de 1980, argumentando que era necesario reformar la inteligencia para hacer frente a la naturaleza dinámica de los requisitos de información, especialmente a nivel táctico en el campo de batalla. En 1992, la Ley de Reorganización de la Inteligencia definió los objetivos de la recopilación de información como "proporcionar inteligencia oportuna, objetiva, libre de prejuicios, basada en todas las fuentes disponibles para la comunidad de inteligencia de los EE.UU., públicas y no públicas". [5]

Si lo comparamos con el OSINT usado actualmente el enfoque es similar. Aunque ahora el fin ya no es meramente militar, sino que también es demandado por compañías y particulares, a los que les interesa tener conocimiento sobre posibles agentes de amenaza, fugas de información, etc.

3.2.2 OSINT en la era de Internet

La llegada de Internet impone nuevas exigencias, especialmente a la capacidad de procesamiento de información de los analistas. Los motores de búsqueda ayudan a

mitigar estas exigencias, aunque solo parcialmente, ya que se está produciendo un cambio cualitativo en los tipos de información.

Uno de los elementos que limitaron el impacto inicial de Internet fue la falta de estructuras organizativas fiables para el flujo constante de información (Stross, 2009). Internet ya ha pasado por una serie de etapas evolutivas, en las cuales han cambiado las formas en que se organiza la información en función del uso y el reconocimiento, incluido el desarrollo de la tecnología de los motores de búsqueda y los sistemas de calificación de blogs y wikis. Actualmente las estructuras organizativas más avanzadas han sido desarrolladas por Google.

Mediante un sistema al que Google se refiere como Page rankings (basado en un algoritmo desarrollado por el cofundador de la empresa Larry Page) la información se clasifica en función de los enlaces a la página (Stross, 2009).

De esta manera, mediante los rankings generados se puede definir la calidad de la información. Los usuarios suelen utilizarlos como parte de su estrategia de búsqueda, focalizándose en la primera página de resultados o incluso en el primer resultado. Sin embargo, los primeros resultados no se basan en una comprensión exhaustiva de los términos de búsqueda. Los rankings en la práctica son muy dinámicos y estas variaciones no siempre tienen que ver con la calidad y la afinidad entre la información mostrada y el atributo de búsqueda.

Al hacer clic en la URL de uno de los resultados, el usuario influye activamente en la organización de la información. Una de las dificultades de este tipo de sistema de organización es que una vez que un nodo establece una reputación de "fuente fiable" obtendrá a su vez más clics, creando una reputación más sólida, lo que a su vez dará lugar a más clics y a una posición más alta en las listas de resultados (es un círculo vicioso).

Además, las clasificaciones de las páginas pueden ser manipuladas por otras actividades externas de maneras que nada tienen que ver con la calidad o la relevancia de la información. Técnicas como el "Google bomb" pueden influir fácilmente en la categorización mediante la creación artificial de enlaces.

Por tanto, la responsabilidad y el papel del individuo en la organización de la información no harán sino aumentar, hacerse más complejos y exigir mayores obligaciones sociales a medida que evolucionen las relaciones entre los seres humanos e Internet. [8]

3.3 Metodología OSINT

Hay varios modelos que describen la metodología de inteligencia. El ciclo de inteligencia de la CIA describe este proceso como planificación y dirección, adquisición, procesamiento, análisis y producción, y difusión. El “Handbook of Intelligence Studies” describe estas fases como una colección, procesamiento, análisis y producción, clasificación y difusión.

En el contexto de OSINT, nos centramos en cuatro fases: adquisición, procesamiento, análisis y producción.

Estas fases consisten en adquirir la información, validar esa información, identificar el valor de la información y proporcionar la información a los clientes. En los siguientes párrafos explicamos con más detalle cada una de estas fases. [6]

3.3.1 Primera fase: Adquisición

En primer lugar, en la fase de adquisición, la información disponible públicamente del objetivo se obtiene de fuentes abiertas relevantes. El proceso de adquisición es particularmente relevante porque a partir de esta fase se activa todo el proceso de generación de inteligencia. [9]

Internet es el recurso por excelencia debido al volumen de material existente y la facilidad de acceso. Hay que tener presente que el volumen de información disponible en Internet es prácticamente inabordable, por lo que se deben identificar y concretar las fuentes de información con el fin de optimizar el proceso de adquisición. [3]

Existen multitud de fuentes abiertas a partir de las cuales se puede obtener información relevante, entre las que destacan:

- Motores de búsqueda, foros, redes sociales, blogs, wikis, etc.
- Medios de comunicación: revistas, periódicos, radio, etc.
- Conferencias, simposios, «papers», bibliotecas online, etc.
- Información pública de fuentes gubernamentales.

En esta fase, se supone que, al menos, hay disponible alguno de los datos sobre el objetivo. En particular, se consideran el nombre de usuario, nombre real, redes sociales, dirección de correo electrónico, sitio web, dirección IP y ubicación.

A partir de esta información, el investigador aplica las técnicas OSINT más adecuadas para expandir el conjunto de datos sobre el objetivo. Para ello, el output obtenido con una técnica específica se utiliza como el input de otras técnicas.

3.3.2 Segunda fase: Procesamiento

La fase de procesamiento consiste en validar y dar formato a toda la información recopilada de manera que posteriormente pueda ser analizada.

Identificamos dos componentes del procesamiento: traducción y agregación. Estos componentes no necesariamente tienen que ocurrir en un orden dado, aunque en ciertos casos uno podría ayudar al otro.

El procesamiento incluye la traducción del idioma original de la información y la transformación de videos o fotografías en inteligencia utilizable. Muchas de las tareas del procesamiento ahora pueden realizarse más fácilmente y a menor costo mediante el uso de software. A su vez, hoy en día hay una gran cantidad de información disponible en un formato menos estructurado, lo que hace que el procesamiento sea mucho más complicado.

La agregación, que generalmente no es necesaria para la información proveniente de medios de comunicación, papers e información pública de fuentes gubernamentales, es un paso crítico para el análisis de muchos tipos de contenido de redes sociales, en particular el contenido de las redes sociales de formato corto como Facebook, Twitter y LinkedIn.

La agregación también puede implicar la reducción o integración de un conjunto de datos para convertirlos en un formato utilizable. Muchas empresas prestan servicios de agregación de datos que evitan la necesidad de recopilarlos al sector de inteligencia. Si bien estos agregadores de datos pueden minimizar la recopilación y el procesamiento de la información, es posible que no proporcionen datos de múltiples plataformas ni datos completos; pudiendo además ser difícil saber exactamente qué datos se han incluido en el conjunto de datos, lo que complica la capacidad de verificarlos y ponerlos en un contexto apropiado. [6]

3.3.3 Tercera fase: Análisis

En la fase de análisis se genera inteligencia a partir de los datos recopilados y procesados. El objetivo es relacionar la información de distintos orígenes buscando patrones que permitan llegar a alguna conclusión significativa. Los datos en sí mismos

no son útiles, por lo que deben interpretarse para obtener las primeras conclusiones a partir de un análisis en profundidad.

Hay una cantidad cada vez mayor de técnicas de análisis en la literatura para llevar a cabo esta tarea de análisis, destacando a continuación los procedimientos que son aplicables en nuestro escenario:

- Análisis léxico: Los datos sin procesar deben examinarse para extraer entidades y relaciones del texto. Es esencial filtrar lo que no agregue valor.
- Análisis semántico: Con el propósito de comprender el significado de los datos, actualmente se utilizan algoritmos de procesamiento del lenguaje natural. Además, las técnicas de análisis sentimental permiten contextualizar publicaciones u opiniones subjetivas para clasificar el estado emocional del autor (por ejemplo, positivo, negativo o neutral). Finalmente, los procedimientos de descubrimiento de la verdad abordan el reto de resolver conflictos en datos provenientes de múltiples fuentes que representan posiciones opuestas sobre el mismo tema.
- Análisis geoespacial: Es útil analizar los datos recogidos de redes sociales, eventos, sensores o direcciones IP desde una perspectiva basada en la ubicación. En este sentido, el uso de mapas o gráficos facilita la representación y comprensión de los datos, además de extraer conexiones significativas entre incidentes o personas.
- Análisis de redes sociales: Las redes sociales permiten a los investigadores llevar a cabo un análisis exhaustivo de los usuarios. En este escenario, el análisis de datos sociales permite la creación de una red de contactos, interacciones, lugares, comportamientos y gustos en torno al sujeto.

Los resultados de aplicar las técnicas mencionadas anteriormente se consideran información de salida, clasificándose principalmente en tres grupos:

- La información personal comprende los detalles de identidad de la persona, que se obtienen principalmente a partir del nombre real, dirección de correo electrónico, nombre de usuario, redes sociales y motores de búsqueda.
- La información organizacional está formada por aspectos de un equipo o empresa compuesta por individuos. Se recopila esencialmente mediante redes sociales, motores de búsqueda, ubicación, nombre de dominio y dirección IP.
- La información de la red cubre datos técnicos de los sistemas y las topologías de comunicación, que generalmente se obtienen a través de la ubicación, dominio y dirección IP.

Estos tres bloques de información se pueden ampliar con más elementos, y además una sola investigación puede tener diferentes tipos de información de salida que se complementan entre sí.

La extracción de inteligencia a partir de la información recopilada hasta el momento consiste en el tratamiento de dicha información de salida haciendo uso de minería de datos y técnicas de inteligencia artificial. A continuación mencionamos algunas técnicas para esta etapa:

- Correlación: La detección de relaciones entre personas, eventos o datos en general. Las características fuertemente relacionadas son especialmente valiosas para revelar asociaciones no explícitas existentes en el conjunto de datos.
- Clasificación: Los datos se pueden dividir en grupos según categorías predefinidas (aprendizaje supervisado). Esta técnica permite la organización de grandes cantidades de información para una extracción de conocimiento más efectiva.
- Detección de valores atípicos: Este procedimiento analiza los datos y detecta anomalías en ellos. Son particularmente interesantes para la observación de agentes malignos, cuyo comportamiento o acciones difieren de la población general.
- Agrupación: Asigna datos en grupos, pudiendo considerar una gran cantidad de condiciones o heurísticas. Esto podría revelar, por ejemplo, diferentes formas de comportamiento en la red, varios tipos de perfiles en línea o categorizar tipos de ataques a individuos, organizaciones o infraestructuras sin conocer de antemano la existencia de esa diversidad (aprendizaje no supervisado).
- Regresión: El objetivo principal de esta técnica es predecir valores numéricos o hechos. Por ejemplo, una regresión lineal devuelve un valor que atiende a una función lineal, una red neuronal es una estructura que esquematiza combinaciones complejas de entradas en una salida, o una arquitectura de aprendizaje profundo (deep learning) que se compone de varias capas que se combinan y realizan operaciones con la entrada.
- Patrones de seguimiento: A diferencia de la detección de anomalías, el reconocimiento de patrones es un proceso para detectar regularidades en los datos. Los métodos mencionados anteriormente pueden incluirse en este concepto amplio de descubrimiento de conocimiento. De hecho, cualquier técnica de inteligencia artificial es adecuada para la extracción de conocimiento de datos abiertos.

Estas técnicas permiten inferir aspectos abstractos, complejos y sustanciosos sobre el objetivo que no están publicados explícitamente en Internet. Sin embargo, este proceso plantea varios desafíos, principalmente en cuanto a la investigación y el desarrollo de este proceso de extracción de inteligencia para identificar, retratar o monitorear criminales, reconocer y explorar organizaciones maliciosas o descubrir y atribuir ciberincidentes.

Además, surgen varias consideraciones en cuanto a la privacidad debido a las poderosas inferencias que se pueden lograr. El conocimiento extraído sobre una persona, empresa u organización puede ser especialmente sensible y su manipulación conduce indirectamente a problemas éticos y legales. De hecho, nunca deberíamos perder de vista el hecho de que estas técnicas podrían usarse incluso para dañar directamente a personas o grupos. [9]

3.3.4 Cuarta fase: Producción

En la fase final, producción, la información se proporciona al cliente de forma utilizable. El análisis OSINT se difunde con mayor frecuencia en forma de informe escrito. Sin embargo, los productos también pueden tratarse de resúmenes informativos o visualizaciones gráficas.

El medio utilizado para la difusión suele ser, desafortunadamente, un mecanismo de distribución más sencillo que efectivo. El video, el audio o los gráficos interactivos pueden ser más efectivos que los informes escritos para transmitir información particular.

Los analistas de inteligencia de todas las fuentes generalmente extraen sus informes de inteligencia de una base de datos basada en texto. Del mismo modo, los consumidores de inteligencia a menudo reciben productos de inteligencia en un libro informativo impreso. Sin embargo, el potencial de los portales OSINT y la transición a un formato iPad están facilitando métodos más creativos para transmitir la información, como visualizaciones de datos y archivos dinámicos. [6]

3.4 Herramientas OSINT

Un uso manual de algunas técnicas sería suficiente para búsquedas básicas. Desafortunadamente, el uso de algunos servicios podría no ser efectivo para investigaciones complejas. En este sentido, el potencial de OSINT radica en utilizar tantos servicios como sea posible de forma conjunta, ampliando la información

disponible para juntar todas las piezas del rompecabezas. Sin embargo, no es práctico para el usuario final combinar manualmente varias técnicas OSINT, pues una tarea tan tediosa implicaría largos procesos de investigación.

Para este propósito, los investigadores y desarrolladores han implementado herramientas más precisas para aplicar técnicas OSINT automáticamente y recopilar información de mejor calidad de muchas fuentes diferentes, implementar varios trabajos internamente y, como consecuencia, obtener información más gratificante y mejores inferencias.

A continuación, se describen las características principales de las herramientas OSINT más populares y relevantes en la actualidad. Sin embargo, hay muchas otras recopiladas en OSINT Framework. [9]

3.4.1 FOCA

La principal contribución de FOCA (Fingerprinting Organizations with Collected Archives), diseñada por ElevenPaths, es la extracción y el análisis de los metadatos de documentos. Esta aplicación se puede utilizar tanto para archivos locales como para archivos que se encuentren en páginas web, usando tres motores de búsqueda diferentes (Google, Bing y DuckDuckGo) para buscar documentos.

FOCA soporta una amplia variedad de formatos, tales como Microsoft Office, PDF, Open Office, Adobe InDesign, archivos SVG, etc.

Esta aplicación extrae la información oculta de los archivos y los procesa para mostrar al usuario aspectos relevantes. Algunos datos que se descubren con este procedimiento son el nombre de los ordenadores relacionadas con los documentos, la ubicación donde se crearon los documentos, los sistemas operativos utilizados, los nombres reales y sus direcciones de correo electrónico de dichos usuarios, información acerca de los servidores, la fecha de creación de los documentos, el rango de direcciones IP de redes internas, etc. Como resultado, se puede dibujar un mapa de red basado en los metadatos extraídos para identificar el objetivo.

FOCA incluye además un módulo de descubrimiento de servidor para complementar el análisis de metadatos de los documentos. Algunas técnicas utilizadas en esta herramienta son las siguientes:

- Búsqueda web para buscar hosts y nombres de dominio a través de URL asociadas al dominio dado.

- Búsqueda DNS para descubrir nuevos hosts y nombres de dominio a través de los servidores NS, MX y SPF.
- Resolución IP para obtener las direcciones IP de los hosts encontrados a través de los registros DNS.
- Escaneo PTR para encontrar servidores más en un segmento de red descubierto.
- Bing IP extrae nuevos nombres de dominio asociados a direcciones IP encontradas.

Esta herramienta se usa generalmente en el sector de la seguridad, ya que permite auditar la red y servicios de una empresa. De hecho, puede conseguir muy buenos resultados porque los empleados no suelen quitar los metadatos de los archivos que se suben a Internet. [9] [10]

3.4.2 Maltego

Maltego es una aplicación reconocida que busca información pública sobre un determinado objetivo en diferentes fuentes (registros DNS, registros Whois, motores de búsqueda, redes sociales, APIs, metadatos, etc.). Las relaciones entre los elementos relevantes encontrados se representan para su análisis en forma de gráfico dirigido. Esta herramienta combina cuatro conceptos principales:

- Entidad: es un nodo del gráfico que representa el dato descubierto. Algunas entidades predeterminadas son el nombre real, dirección de correo electrónico, nombre de usuario, red social, empresa, organización, sitio web, documento, asociación, dominio, servidor DNS, dirección IP, etc. Además, también podríamos definir entidades personalizadas para nuestra investigación.
- Transformación: es un fragmento de código que se aplica a una entidad para descubrir una nueva entidad vinculada. Por ejemplo, la transformación “A Dirección IP”, que resuelve un nombre de servidor DNS a una dirección IP, podría aplicarse a una entidad de dominio para crear una nueva entidad con la dirección IP correspondiente. Así, continuaríamos aplicando más transformaciones recurrentemente para extender el proceso de búsqueda. Además de las transformaciones predeterminadas, también es posible implementar e incluir transformaciones personalizadas para fines más específicos.
- Máquina: es un conjunto de transformaciones que se definen de manera conjunta para automatizar y concatenar largos procesos de búsqueda.

- Hub Item: es un conjunto de transformaciones y tipos de entidad que pueden utilizar los usuarios de la comunidad. De manera predeterminada, Maltego implementa el *hub item* denominado “Paterva CTAS” que contiene las entidades, transformaciones y máquinas mantenidas por los desarrolladores oficiales. Además, es posible crear e instalar *hub items* de terceros. [9]

3.4.3 Recon-ng

Recon-ng es un framework de reconocimiento web con una interfaz de línea de comandos similar a Metasploit. Se trabaja en un espacio de trabajo (workspace), seleccionando el módulo a usar entre los disponibles, y si es necesario se pueden establecer las opciones antes de ejecutar el módulo.

Esta herramienta incluye varios módulos independientes que implementan diferentes funcionalidades. Por ejemplo, los módulos Bing Domain Web y Google Site Web buscan hosts conectados a los dominios del espacio de trabajo en los motores de búsqueda Bing y Google; PGP Search escanea los dominios de entrada para las direcciones de correo electrónico asociadas con claves PGP públicas; Full Contact recopila usuarios y sus correspondientes perfiles de redes sociales a partir de los contactos almacenados en la base de datos; o Profiler busca servicios en línea adicionales que posean cuentas con los mismos nombres de usuario que los del área de trabajo.

Recon-ng almacena toda la información obtenida en una base de datos local, de manera que los datos de salida obtenidos con un determinado módulo pueden ser utilizados por otros módulos como datos de entrada. Así, el usuario dirige la investigación seleccionando el módulo indicado y la herramienta automatiza la generación de conocimiento, progresando notablemente para investigaciones complejas. [9]

3.4.4 Shodan

Shodan es un motor de búsqueda que proporciona información pública de los dispositivos conectados a Internet, incluidos los dispositivos IoT. Esto incluye servidores, enrutadores, dispositivos de almacenamiento en línea, cámaras de vigilancia, cámaras web o sistemas VoIP, entre otros. La recopilación de datos se realiza a través de servicios como HTTP o SSH, permitiendo buscar por dirección IP, compañía, ciudad o país.

Esta herramienta se utiliza principalmente para la seguridad de la red (para buscar dispositivos expuestos o para detectar vulnerabilidades de los servicios disponibles públicamente), Internet de las cosas (para controlar el uso creciente de dispositivos inteligentes y su ubicación en la geografía mundial) y rastrear ransomware (para medir la infección provocada por este tipo de ataque). Permite descargar los resultados en formato JSON, CSV o XML, así como generar informes intuitivos.

Además de la funcionalidad mencionada, tiene dos servicios premium: Shodan Maps (maps.shodan.io), que permite investigaciones basadas en ubicaciones, y Shodan Images (images.shodan.io), que muestra imágenes recopiladas de dispositivos públicos. [9]

3.4.5 Spiderfoot

Spiderfoot es otra herramienta de reconocimiento que recorre multitud de fuentes de datos públicas para recopilar información. Nuestra entrada podría ser una dirección IP, subred, dominio, dirección de correo electrónico, nombre de host, nombre real o número de teléfono. Los resultados se representan en un gráfico de nodos con todas las entidades y relaciones encontradas.

Dependiendo del tipo de entrada introducida, la herramienta selecciona de forma autónoma los módulos (equivalentes a las transformaciones de Maltego) para proporcionar un reconocimiento más efectivo. Además, también considera el nivel de búsqueda seleccionado por el usuario, ofreciendo cuatro tipos de escaneos:

- Passive: recolecta la mayor cantidad de información posible sin tocar al objetivo, evitando ser descubierto por el mismo.
- Investigate: realiza un escaneo básico para descubrir si el objetivo es malicioso.
- Footprint: identifica la topología de red del objetivo y recopila información rastreando webs y usando motores de búsqueda, suficiente para investigaciones estándar.
- All: aconsejable para investigaciones detalladas, a pesar de tardar mucho tiempo en completarse, ya que consulta absolutamente todos los recursos posibles relacionados con el objetivo.

Esta herramienta puede utilizarse para recopilar información sobre lo que un individuo o empresa podría tener expuesto en Internet o para realizar el reconocimiento del objetivo en un pentest. Además, vale la pena señalar que es posible desarrollar módulos de Spiderfoot. [9]

3.4.6 The Harvester

The Harvester permite recopilar información pública relacionada con un dominio o nombre de empresa a través de motores de búsqueda. En particular, es capaz de enumerar correos electrónicos y nombres de host de la empresa, así como subdominios, direcciones IP y URLs relacionadas con el dominio. También permite exportar los resultados en HTML o XML.

Esta aplicación de consola se utiliza en las primeras fases de un pentest, ayudando a determinar el escenario de amenazas externas de una empresa en Internet. [9]

3.5 Comparación de herramientas OSINT

Dependiendo de las necesidades del usuario, algunas herramientas serán más adecuadas que otras dependiendo para una tarea determinada.

Si lo que buscas es extraer información oculta de archivos, FOCA y Maltego son las herramientas más adecuadas. En particular, FOCA está específicamente diseñada para este fin. Presenta funcionalidades adicionales, aparte del análisis de metadatos, para proporcionar información adicional que complementa a dicha información oculta. De esta manera, es capaz de generar más conocimiento sobre el objetivo.

Sin embargo, si lo que queremos es buscar información de redes, Shodan, Spiderfoot y The Harvester son las opciones recomendadas. Por una parte, se recomienda Spiderfoot para analizar la topología de la red y obtener información interna (pero pública) sobre la compañía objetivo. Por otra parte, estos resultados se completan con Shodan, que incluye información específica sobre dispositivos de IoT, cámaras de vigilancia, cámaras web, sistemas VoIP, o servicios inteligentes en general.

Por último, si el objetivo de la búsqueda es reunir la mayor cantidad de información posible para una entrada determinada, las herramientas Recon-ng y Maltego son las más completas y extraerán diversos datos y relaciones entre ellos.

Recon-ng contiene muchos módulos que interactúan con una base de datos local, la cual va escalando durante la investigación, siendo un framework ideal para recopilar información de un individuo o empresa expuesta en Internet, llevar a cabo pentests, prevenir phishings y ataques de ingeniería social, e incluso elaborar perfiles de personas.

Si se quiere evitar la línea de comandos y optar por una interfaz más intuitiva, Maltego es una buena alternativa a Recon-ng. Maltego implementa procesos

automatizados de inferencia con transformaciones que aumentan el alcance de la búsqueda original. Además, admite procedimientos de búsqueda personalizados.

A pesar de que la comparación descrita anteriormente se ha realizado de acuerdo con la salida deseada, en la práctica el usuario estará restringido por la entrada disponible y el tipo de datos aceptado por las herramientas OSINT elegidas.

Por último, nótese que estas herramientas son complementarias y no exclusivas, lo que significa en una investigación profunda y exhaustiva de OSINT podría ser de utilidad combinar varias de ellas. Aunque algunas pueden producir resultados similares para una búsqueda determinada, siempre pueden encontrarse datos extraídos por una herramienta que no son obtenidos por otras. [9]

3.6 Bases de datos SQL y NoSQL

En general, al comparar bases de datos SQL (relacionales) con NoSQL (no relacionales), estas últimas tienen un mejor rendimiento en tiempo de ejecución para inserciones, actualizaciones y consultas simples. Por otro lado, SQL rinde mejor al actualizar y consultar atributos que no son clave, así como en consultas con funciones agregadas.

Una base de datos NoSQL podría ser una buena solución para conjuntos de datos más grandes en los que su esquema cambia constantemente o en el caso de que las consultas realizadas sean menos complejas. Dado que en NoSQL realmente no hay un esquema definido, al contrario de SQL que requiere un esquema definido rigurosamente, soportaría fácilmente un esquema dinámico, como un sistema de gestión de documentos con varios campos dinámicos y solo unos pocos campos indexados que sean conocidos.

En conclusión, NoSQL es una buena solución cuando se necesita una estructura de base de datos más flexible, con conjuntos de datos grandes en los que el esquema cambie constantemente o en el caso de que las consultas que se vayan a realizar no sean demasiado complejas, debido a su bajo rendimiento para funciones agregadas y consultas basadas en valores que no sean clave.

3.7 Herramientas para el análisis de datos recopilados en fuentes abiertas

El crecimiento exponencial de los datos en Internet plantea un desafío importante en el proceso de obtención de un conjunto de datos representativos que pueda traducirse en resultados tangibles.

El preprocesamiento en tiempo real añade otra capa de complejidad, especialmente cuando los datos son textuales y no estructurados o de multitud de fuentes.

Actualmente, las tres principales herramientas utilizadas para el análisis de grandes bases de datos son Elasticsearch, Hadoop y Spark.

Elasticsearch es un motor analítico y de búsqueda distribuida que permite transformaciones de datos en tiempo real, consultas de búsqueda, procesamiento de flujo de documentos e indexación a una velocidad relativamente alta. Además, Elasticsearch puede indexar números, coordenadas geográficas, fechas y casi cualquier tipo de datos, y es compatible con múltiples idiomas (es decir, Python, Java, Ruby). La velocidad del motor de Elasticsearch se basa en su capacidad de realizar la agregación, la búsqueda y el procesamiento del índice de los datos.

Hadoop es una plataforma de computación distribuida por lotes, que utiliza el algoritmo MapReduce, que incluye capacidades de extracción y transformación de datos. Si bien la plataforma se basa en la tecnología NoSQL que facilita la carga de datos no estructurados, su procesamiento de consultas HBASE no tiene capacidades de búsqueda analítica avanzada como Elasticsearch.

Elasticsearch dispone también de un plugin de visualización para el análisis en tiempo real con una licencia de código abierto. Añadir que Elasticsearch dispone de plugins para Hadoop y Spark para reducir la distancia entre las dos tecnologías diferentes y permite implementar un sistema híbrido.

Entre las herramientas que apoyan la gestión de grandes conjuntos de datos y la obtención de datos en tiempo real se encuentran las herramientas relacionales (MySQL, Oracle Database, SQLite), las herramientas gráficas (Neo4j, Oracle Spatial) y las herramientas NoSQL (MongoDB, IBM Domino, Apache CouchDB).

Entre los factores limitantes relacionados con todos los tipos de bases de datos cabe mencionar la falta de apoyo a las búsquedas de texto completo en tiempo real. Si bien NoSQL es funcional para las búsquedas de texto completo, carece de fiabilidad cuando

se compara con los modelos de bases de datos relacionales. Las bases de datos tradicionales requieren que los datos se carguen primero y luego el administrador debe decidir activamente qué datos deben indexarse, lo que añade una capa más de procesamiento que lo hace inviable para el análisis en tiempo real.

Elasticsearch ofrece una solución a estos factores limitantes al proporcionar un sistema altamente eficiente de búsqueda de datos y análisis en tiempo real que:

- Realiza una preindexación antes de almacenar los datos para evitar la necesidad de buscar y consultar datos específicos en tiempo real.
- Requiere recursos y potencia de cálculo limitados en relación con las soluciones tradicionales.
- Proporciona un sistema distribuido y fácil de escalar. [11]

3.8 Elasticsearch

Elasticsearch se inició en el año 2004 como un proyecto de código abierto llamado *compass*, basado en Apache Lucene. Elasticsearch es un motor de búsqueda y análisis RESTful para todos los tipos de datos, incluidos textuales, numéricos, geoespaciales, estructurados y desestructurados. Está desarrollado en Java y es open source. Estas características, combinadas con la flexibilidad y las opciones de fácil expansión, son útiles para el análisis de grandes cantidades de datos en tiempo real.

Conocido por sus API REST simples, naturaleza distribuida, velocidad y escalabilidad, Elasticsearch es el componente principal del Elastic Stack, un conjunto de herramientas open source para la ingestión, el enriquecimiento, el almacenamiento, el análisis y la visualización de datos. [12]

3.8.1 ¿Cómo funciona Elasticsearch?

La ingestión de datos es el proceso mediante el cual los datos son parseados, normalizados y enriquecidos antes de su indexación en Elasticsearch. Una vez indexados en Elasticsearch, los usuarios pueden ejecutar consultas complejas sobre sus datos y usar agregaciones para recuperar resúmenes complejos de sus datos. [12]

3.8.2 ¿Qué es un índice de Elasticsearch?

Un índice de Elasticsearch es una colección de documentos relacionados entre sí. Elasticsearch almacena datos como documentos JSON. Cada documento correlaciona

un conjunto de claves (nombres de campos o propiedades) con sus valores correspondientes (textos, números, Booleanos, fechas, variedades de valores, geolocalizaciones u otros tipos de datos).

Elasticsearch usa una estructura de datos llamada índice invertido, que está diseñada para permitir búsquedas de texto completo muy rápidas. Un índice invertido hace una lista de cada palabra única que aparece en cualquier documento e identifica todos los documentos en que ocurre cada palabra.

Durante el proceso de indexación, Elasticsearch almacena documentos y construye un índice invertido para poder buscar datos en el documento casi en tiempo real. [12]

3.8.3 ¿Por qué usar Elasticsearch?

- Elasticsearch es rápido. Como Elasticsearch está desarrollado sobre Lucene, es excelente en la búsqueda de texto completo. Elasticsearch también es una plataforma de búsqueda prácticamente en tiempo real, lo que implica que la latencia entre el momento en que se indexa un documento hasta el momento en que se puede buscar en él es muy breve: típicamente, un segundo. Como resultado, Elasticsearch está bien preparado para casos de uso con restricciones de tiempo como analítica de seguridad y monitoreo de infraestructura.
- Elasticsearch es distribuido por naturaleza. Los documentos almacenados en Elasticsearch se distribuyen en distintos contenedores conocidos como *shards*, que están duplicados para brindar copias redundantes de los datos en caso de que falle el hardware. La naturaleza distribuida de Elasticsearch le permite escalar horizontalmente a cientos (o incluso miles) de servidores y gestionar petabytes de datos.
- Elasticsearch viene con un amplio conjunto de características. Además de su velocidad, la escalabilidad y la resistencia, Elasticsearch tiene una cantidad de características integradas poderosas que contribuyen a que el almacenamiento y la búsqueda de datos sean incluso más eficientes, como data rollup y gestión de ciclo de vida del índice.
- Elastic Stack simplifica la ingesta de datos, la visualización y el reporte. La integración con Beats y Logstash facilita el proceso de datos antes de indexarlos en Elasticsearch. Y Kibana provee visualización en tiempo real de los datos de Elasticsearch así como UI para acceder rápidamente al monitoreo de rendimiento de aplicaciones (APM), los logs y los datos de métricas de infraestructura. [12]

3.8.4 Kibana

Además de que Elasticsearch es eficiente para el análisis en tiempo real, los plugins como Kibana lo hacen conveniente para las representaciones funcionales de big data en tiempo real. [11]

Kibana brinda histogramas, gráficos circulares y mapas. Kibana también incluye aplicaciones avanzadas, como Canvas, que permite a los usuarios crear infografías dinámicas personalizadas con base en sus datos, y Elastic Maps para visualizar los datos geoespaciales. [12]

También tiene disponibles múltiples visualizaciones estándar por defecto y simplifica el proceso de desarrollo de visualizaciones para los usuarios finales gracias a la funcionalidad de arrastrar y soltar. Como Kibana está respaldado por la arquitectura de Elasticsearch, funciona rápidamente y es lo suficientemente eficiente para el análisis en tiempo real. Por último, ofrece interacción gráfica en el proceso de construcción y manejo de consultas con una visualización accesible. [11]

4 Reingeniería

El framework Recon-ng almacena la información adquirida con sus módulos en una base de datos SQL. Dado que se requiere de una estructura más flexible para almacenar grandes conjuntos de datos obtenidos de diversas fuentes, es conveniente utilizar una base de datos NoSQL, por lo cual hemos decidido realizar la migración a Elasticsearch.

4.1 Ingeniería inversa

En primer lugar, para analizar el esquema de la base de datos SQL se ha obtenido el diagrama Entidad-Relación.

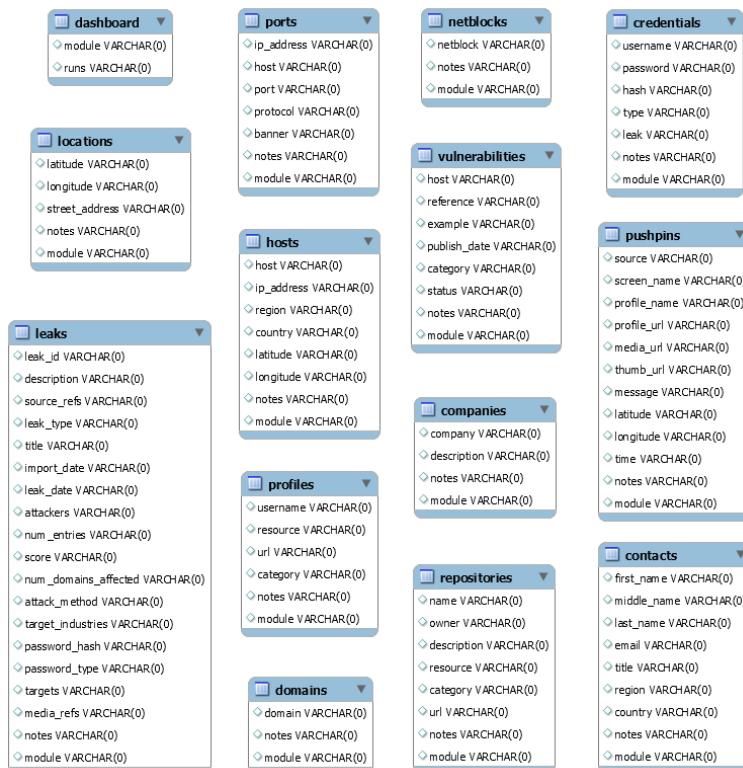


Ilustración 5 - Diagrama Entidad-Relación

Como se puede observar, las tablas no tienen relaciones y por tanto es un motivo más para utilizar una base de datos NoSQL.

Tras llevar a cabo la ingeniería inversa de la base de datos, se ha realizado un análisis de los componentes del framework que interactúan con esta.

Un workspace, ya sea por defecto (default) o definido por el usuario, se inicializa con el método “`_init_workspace()`” de la clase base denominada “Recon”, creándose el directorio correspondiente donde se encontrará la base de datos, la cual se crea mediante el método “`_create_db()`”:

```
recon > core > base.py > Recon > _init_workspace
193     def _init_workspace(self, workspace):
194         if not workspace:
195             return
196         path = os.path.join(self.spaces_path, workspace)
197         self.workspace = framework.Framework.workspace = path
198         if not os.path.exists(path):
199             os.makedirs(path)
200         self._create_db()
201     else:
202         self._migrate_db()
203     # set workspace prompt
204     self.prompt = self._prompt_template.format(self._base_prompt[:-3], self.workspace.split('/')[-1])
205     # load workspace configuration
206     self._load_config()
207     # reload modules after config to populate options
208     self._load_modules()
209     return True
```

Ilustración 6 - Método `_init_workspace()` de la clase Recon

De esta manera, se crean las tablas de la base de datos haciendo uso del método “`query()`” de la clase “Framework”, el cual llama al método “`_query()`”, donde se ejecuta la consulta SQL pasada por parámetro:

```
recon > core > framework.py > Framework > _query
411     def query(self, *args, **kwargs):
412         path = os.path.join(self.workspace, 'data.db')
413         return self._query(path, *args, **kwargs)
414
415     def _query(self, path, query, values=(), include_header=False):
416         '''Queries the database and returns the results as a list.'''
417         self.debug(f"DATABASE => {path}")
418         self.debug(f"QUERY => {query}")
419         with sqlite3.connect(path) as conn:
420             with closing(conn.cursor()) as cur:
421                 if values:
422                     self.debug(f"VALUES => {repr(values)}")
423                     cur.execute(query, values)
424                 else:
425                     cur.execute(query)
426                     # a rowcount of -1 typically refers to a select statement
427                     if cur.rowcount == -1:
428                         rows = []
429                         if include_header:
430                             rows.append(tuple([x[0] for x in cur.description]))
431                             rows.extend(cur.fetchall())
432                         results = rows
433                         # a rowcount of 1 == success and 0 == failure
434                     else:
435                         conn.commit()
436                         results = cur.rowcount
437
438         return results
```

Ilustración 7 - Ejecución de consultas SQL en el método `_query()` de la clase Framework

Para almacenar los datos adquiridos con los módulos del framework, en los métodos “insert” de la clase “Framework” se llama al método “self.insert()”, que inserta el contenido del diccionario “data” en la base de datos SQL.

```
recon > core > framework.py > Framework
551     def insert_domains(self, domain=None, notes=None, mute=False):
552         '''Adds a domain to the database and returns the affected row count.'''
553         data = dict(
554             domain = domain,
555             notes = notes
556         )
557         rowcount = self.insert('domains', data.copy(), data.keys())
558         if not mute: self._display(data, rowcount, '[domain] %s', data.keys())
559         return rowcount
560
561     def insert_netblocks(self, netblock=None, notes=None, mute=False):
562         '''Adds a netblock to the database and returns the affected row count.'''
563         data = dict(
564             netblock = netblock,
565             notes = notes
566         )
567         rowcount = self.insert('netblocks', data.copy(), data.keys())
568         if not mute: self._display(data, rowcount, '[netblock] %s', data.keys())
569         return rowcount
570
571     ...
572
```

Ilustración 8 - Métodos “insert” de la clase Framework

Los parámetros del método “insert()” son la tabla donde insertar los datos, el diccionario con dichos datos y una lista de nombres de columnas que se utilizan para comprobar que la información a insertar no sea duplicada. Así, se construye la consulta SQL y se ejecuta con el método “query()”.

```
recon > core > framework.py > insert
755     def insert(self, table, data, unique_columns=[]):
756         '''Inserts items into database and returns the affected row count.'''
757         # set module to the calling module unless the do_add command was used
758         data['module'] = 'user_defined' if '_do_db_insert' in [x[3] for x in inspect.stack()]
759         else self._modulename.split('/')[-1]
760         # sanitize the inputs to remove NoneTypes, blank strings, and zeros
761         columns = [x for x in data.keys() if data[x]]
762         # make sure that module is not seen as a unique column
763         unique_columns = [x for x in unique_columns if x in columns and x != 'module']
764         # exit if there is nothing left to insert
765         if not columns:
766             return 0
767         # convert any type to unicode (str) for external processing
768         for column in columns:
769             data[column] = self.to_unicode_str(data[column])
770
771         # build the insert query
772         columns_str = ', '.join(columns)
773         placeholder_str = ', '.join('?' * len(columns))
774         unique_columns_str = ' AND '.join(['{}=?'.format(column) for column in unique_columns])
775         if not unique_columns:
776             query = "INSERT INTO `{}({})` VALUES ({})".format(table, columns_str, placeholder_str)
777         else:
778             query = "INSERT INTO `{}({})` SELECT {} WHERE NOT EXISTS(SELECT * FROM `{}` WHERE {})".format(
779                 table, columns_str, placeholder_str, table, unique_columns_str)
780             values = tuple([data[column] for column in columns] + [data[column] for column in unique_columns])
781
782         # query the database
783         rowcount = self.query(query, values)
784
785         # increment summary tracker
786         if table not in self._summary_counts:
787             self._summary_counts[table] = {'count': 0, 'new': 0}
788             self._summary_counts[table]['new'] += rowcount
789             self._summary_counts[table]['count'] += 1
790
791         return rowcount
```

Ilustración 9 - Método insert() de la clase Framework

Por otra parte, para obtener el input de la base de datos al ejecutar los módulos se realiza la siguiente consulta a la base de datos SQLite:

```
'query': 'SELECT DISTINCT domain FROM domains WHERE domain IS NOT NULL'
```

Ilustración 10 - Consulta SQL para obtener el input en los módulos

Dicha consulta se lleva a cabo en el método “_get_source()” de la clase BaseModule.

```
recon > core > module.py > BaseModule > _get_source
137     def _get_source(self, params, query=None):
138         prefix = params.split()[0].lower()
139         if prefix in ['query', 'default']:
140             query = ''.join(params.split()[1:]) if prefix == 'query' else query
141             try:
142                 results = self.query(query)
143             except sqlite3.OperationalError as e:
144                 raise framework.FrameworkException(f"Invalid source query. {type(e).__name__} {e}")
145             if not results:
146                 sources = []
147             elif len(results[0]) > 1:
148                 sources = [x[:len(x)] for x in results]
149                 #raise framework.FrameworkException('Too many columns of data as source input.')
150             else:
151                 sources = [x[0] for x in results]
152             elif os.path.exists(params):
153                 sources = open(params).read().split()
154             else:
155                 sources = [params]
156             if not sources:
157                 raise framework.FrameworkException('Source contains no input.')
158         return sources
```

Ilustración 11 - Método _get_source() de la clase BaseModule

4.2 Plan de migración

En la migración que se va a realizar la funcionalidad básica del framework Recon-ng se mantiene, migrando el esquema de su base de datos relacional (SQLite) a una base de datos NoSQL (Elasticsearch), con la consiguiente adaptación de los componentes del sistema para interaccionar con la base de datos.

El proceso de migración tendrá un enfoque incremental, es decir el proceso de reingeniería será gradual, de tal forma que cada paso se acerca al sistema objetivo. Aplicamos dicho enfoque incremental debido a que se requiere realizar una transición gradual del sistema legado al nuevo sistema con los recursos técnicos disponibles y con el menor impacto posible.

La metodología consiste en realizar la migración de la base de datos relacional a Elasticsearch, basándose en una serie de etapas incrementales en funcionalidad. En cada etapa se realizará la migración y sincronización de un grupo de entidades hasta llegar a la etapa final donde la aplicación legada dejará de funcionar.

A continuación, definimos las etapas de la migración:

- Etapa 1:

Definición de la estructura de los documentos de Elasticsearch.

- Etapa 2:

Implementación de la funcionalidad para realizar la conexión con Elasticsearch.

Implementación de la funcionalidad para crear un índice en Elasticsearch por cada workspace.

Creación de los métodos CRUD.

- Etapa 3:

Adaptación de la funcionalidad para la obtención de los datos de entrada de los módulos de Recon-ng.

Adaptación de los métodos “Insert” del framework utilizados para almacenar los datos de salida de los módulos.

4.3 Migración del modelo de datos

4.3.1 Correspondencia entre SQL y Elasticsearch

En la siguiente tabla se muestra la correspondencia entre los conceptos de SQL y Elasticsearch:

Terminología SQL	Terminología Elasticsearch
Base de datos	Índice
Tabla	Tipo de documento
Fila	Documento
Columna	Campo

Tabla 8 - Correspondencia entre terminología SQL y Elasticsearch

La representación de las tablas en la base de datos SQL es la siguiente:

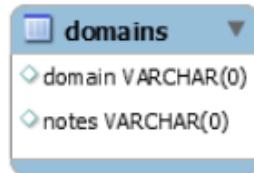


Ilustración 12 - Tabla de la base de datos SQL

A continuación, se muestra el modelo de documentos de Elasticsearch:

```
{
  "_index": "upm",
  "_type": "_doc",
  "_id": "J4hcQHEBUGe4AQWMr8A5",
  "_score": 4.526792,
  "_source": {
    "type": "domains",
    "domain": "etsisi.upm.es",
    "notes": null
  }
},
{
  "_index": "upm",
  "_type": "_doc",
  "_id": "JohcQHEBUGe4AQWMr8A5",
  "_score": 4.526792,
  "_source": {
    "type": "domains",
    "domain": "upm.es",
    "notes": null
  }
}
```

Ilustración 13 - Documentos de Elasticsearch

Como se puede apreciar, el campo “_source” contiene los datos de los documentos JSON indexados. Para poder agrupar los documentos por tipo, es necesario añadir tu propio campo “type”, ya que el campo “_type” de los documentos está deprecado a partir de la versión 7.0 de Elasticsearch, siendo su valor por defecto “_doc”.

4.3.2 Estructura de los documentos de Elasticsearch

Con los módulos de Recon-*ng* que vamos a desarrollar se insertarán distintos tipos de datos en Elasticsearch, algunos de los cuales no existían en la base de datos SQL. Por ello, se definen los siguientes nuevos tipos de documentos con sus campos correspondientes:

Tipo de documento	Campos de datos
news	title, url, date
emails	email, source
documents	domain, url, metadata
similarDomains	original_domain, similar_domain, ips, vt_positives, snapshot, screenshot
pastes	title, url, date, content
posts	title, url, date
spamMailServers	domain, mail_server, ip, blacklist

Tabla 9 - Estructura de los nuevos tipos de documentos de Elasticsearch

También se definen los tipos de documentos correspondientes a las tablas existentes en la base de datos SQL:

Tipo de documento	Campos de datos
domains	domain
companies	company, description
netblocks	netblock
locations	latitude, longitude, street_address
vulnerabilities	host, reference, example, publish_date, category, status
ports	ip_address, host, port, protocol, banner

hosts	host, ip_address, region, country, latitude, longitude
contacts	first_name, middle_name, last_name, email, title, region, country
credentials	username, password, _hash, _type, leak
leaks	leak_id, description, source_refs, leak_type, title, import_date, leak_date, attackers, num_entries, score, num_domains_affected, attack_method, target_industries, password_hash, password_type, targets y media_refs
pushpins	screen_name, profile_name, profile_url, media_url, thumb_url, message, latitude, longitude y time
profiles	username, resource, url, category
repositories	name, owner, description, resource, category, url

Tabla 10 - Estructura de los documentos de Elasticsearch correspondientes a las tablas de la base de datos SQL

Además de los campos especificados, todos los documentos tienen el campo “type”, donde se indica el tipo de documento, así como el campo “timestamp”, en el cual se indica la hora de inserción.

4.4 Integración de Elasticsearch en Recon-ng

4.4.1 Creación de índices y operaciones CRUD

Para hacer las consultas de Elasticsearch es necesario sustituir el método “query()” de la clase “Framework”. Además, dado que en NoSQL cada documento puede tener una estructura diferente, se puede prescindir del método donde se realiza la creación de tablas.

En primer lugar, se necesita un método con el que realizar la conexión a Elasticsearch cuando sea necesario hacer una consulta a la base de datos, por lo que lo agregamos en la clase “Framework”:

```
recon > core > framework.py > Framework > connect_ES
363     def connect_ES(self):
364         es = Elasticsearch([{'host': 'localhost', 'port': 9200}])
365         return es
```

Ilustración 14 - Método connect_ES() agregado a la clase Framework

Al inicializar el workspace tan solo se tendrá que crear un índice en Elasticsearch, que será la colección de documentos correspondientes al workspace. Para ello, en la clase “Recon” añadimos el siguiente método:

```
recon > core > base.py > Recon > create_index_ES
237     def create_index_ES(self, index):
238         es = self.connect_ES()
239         try:
240             if not es.indices.exists(index):
241                 es.indices.create(index=index)
242                 self.alert('Created Index ' + index)
243             except Exception as e:
244                 self.error(e)
```

Ilustración 15 - Método create_index_ES() agregado a la clase Recon

Al cual se llama desde el método “_init_workspace()”, con el que se inicializa el workspace:

```
recon > core > base.py > Recon > _init_workspace
193     def _init_workspace(self, workspace):
194         if not workspace:
195             return
196         path = os.path.join(self.spaces_path, workspace)
197         self.workspace = framework.Framework.workspace = path
198         if not os.path.exists(path):
199             os.makedirs(path)
200             self._create_db()
201             self.create_index_ES(workspace)
202         else:
203             self._migrate_db()
204             # set workspace prompt
205             self.prompt = self._prompt_template.format(self._base_prompt[:-3], self.workspace.split('/')[-1])
206             # load workspace configuration
207             self._load_config()
208             # reload modules after config to populate options
209             self._load_modules()
210             return True
```

Ilustración 16 - Llamada al método self.create_index_ES() agregada en el método _init_workspace() de la clase Recon

Una vez se tienen los métodos necesarios para conectarse a Elasticsearch y crear un índice por cada workspace, faltaría poder realizar operaciones de creación, lectura, actualización y eliminación (CRUD) de documentos, reemplazando el método “_query()” utilizado en el framework para la realización de consultas SQL. De esta manera, en la clase “Framework” codificamos un método para cada operación:

```

recon > core > framework.py
367     def create_doc_ES(self, index, body, mute=False):
368         es = self.connect_ES()
369         _id = self.create_id(body.copy())
370         try:
371             res = es.index(index=index, id=_id, body=body, op_type='create', refresh='wait_for')
372             if not mute: self.verbose(f"Document created:\n{res}")
373         except Exception as e:
374             if not mute: self.error(e)
375
376     def update_doc_ES(self, index, _id, body, mute=False):
377         es = self.connect_ES()
378         try:
379             res = es.index(index=index, id=_id, body=body, op_type='index', refresh='wait_for')
380             if not mute: self.verbose(f"Document updated:\n{res}")
381         except Exception as e:
382             if not mute: self.error(e)
383
384     def read_doc_ES(self, index, body, mute=False):
385         es = self.connect_ES()
386         try:
387             res = es.search(index=index, body=body, size=10000)
388             hits = res['hits']['hits']
389             if not mute: self.verbose(f"Document(s) read:\n{hits}")
390         except Exception as e:
391             if not mute: self.error(e)
392         #return res
393         return hits
394
395     def delete_doc_ES(self, index, _id, mute=False):
396         es = self.connect_ES()
397         try:
398             res = es.delete(index=index, id=_id)
399             if not mute: self.verbose(f"Document deleted:\n{res}")
400         except Exception as e:
401             if not mute: self.error(e)

```

Ilustración 17 - Métodos agregados a la clase Framework para realizar las operaciones CRUD en Elasticsearch

En cuanto a la creación y actualización de documentos, con el parámetro “refresh='wait_for'” hasta que el documento haya sido indexado, Elasticsearch no responderá a otras operaciones. Así, se evita que las consultas que se realicen inmediatamente después de insertar el documento devuelvan una cadena vacía mientras no esté disponible aún para ser consultado.

Cabe destacar que nos encontramos con el problema de que Elasticsearch no gestiona la duplicidad de documentos, pues el _id que se genera por defecto al insertar un documento es único. Por tanto, pueden insertarse documentos iguales a los que ya se encuentren indexados, y entonces estarán duplicados.

Para evitar esto, se podría realizar una búsqueda previa a la inserción para comprobar si el documento ya existe, lo cual según el tamaño de la base de datos puede que no sea eficiente. Otra opción es que `_id` del documento sea el hash de sus campos, de manera que dicho hash será el mismo para documentos iguales, evitando así la inserción de documentos duplicados.

Puesto que consideramos óptima esta última opción, hemos codificado un método para realizar la generación del hash que se utilizará como `_id` al crear un documento. Para ello, concatenamos los valores de los campos del documento, exceptuando el campo “`timestamp`”. También se pasan a minúscula, para así evitar que se inserten documentos duplicados, y por último se genera el hash SHA1 del string con los valores de los campos concatenados:

```
recon > core > framework.py > Framework > create_id
403     def create_id(self, data):
404         combined_key = ""
405         data.pop('timestamp', None)
406         for key in data.keys():
407             combined_key += (str(data[key])).lower()
408         _id = hashlib.sha1(combined_key.encode('utf-8')).hexdigest()
409         return _id
```

Ilustración 18 - Método `create_id()` agregado a la clase `Framework`

4.4.2 Obtención de los datos de entrada de los módulos

La nueva consulta para obtener el input de Elasticsearch al ejecutar los módulos de Recon-ng es la siguiente:

```
'query': {'_source': ['domain'], 'query': {'match': {'type': 'domains'}}}
```

Ilustración 19 - Consulta para obtener el input de Elasticsearch en los módulos

Para llevar a cabo dicha consulta se ha realizado la modificación que se muestra a continuación en el método “`_get_source()`” de la clase `BaseModule`:

```
recon > core > module.py > BaseModule > _get_source
137     def _get_source(self, params, query=None):
138         prefix = params.split()[0].lower()
139         if prefix in ['query', 'default']:
140             query = ''.join(params.split()[1:]) if prefix == 'query' else query
141             '''try:
142                 results = self.query(query)
143             except sqlite3.OperationalError as e:
144                 raise framework.FrameworkException(f"Invalid source query. {type(e).__name__} {e}")
145             if not results:
146                 sources = []
147             elif len(results[0]) > 1:
148                 sources = [x[:len(x)] for x in results]
149                 #raise framework.FrameworkException('Too many columns of data as source input.')
150             else:
151                 sources = [x[0] for x in results]'''
152             docs = self.read_doc_ES(os.path.basename(self.workspace), query)
153             sources = []
154             for doc in docs:
155                 for value in doc['_source'].values():
156                     sources.append(value)
157             elif os.path.exists(params):
158                 sources = open(params).read().split()
159             else:
160                 sources = [params]
161             if not sources:
162                 raise framework.FrameworkException('Source contains no input.')
163         return sources
```

Ilustración 20 - Método `_get_source()` modificado

Mediante el método “`self.read_doc_ES()`” se ejecuta la consulta, y por cada documento encontrado en Elasticsearch se obtiene el valor del campo “`_source`”, que contiene el documento JSON indexado.

4.4.3 Inserción de los datos adquiridos con los módulos

Para insertar en Elasticsearch los datos adquiridos con los módulos de Recon-ng se ha realizado una modificación de los métodos “insert” de la clase “Framework”. De esta manera, en el diccionario “data” se añaden los campos “type” y “timestamp”. Por último, se sustituye el método “self.insert()” por el método “self.create_doc_ES()”, con el cual se inserta el diccionario en Elasticsearch.

```
def insert_domains(self, domain=None, notes=None, mute=False):
    '''Adds a domain to the database and returns the affected row count.'''
    data = dict(
        type = 'domains',
        timestamp = self.date(),
        domain = domain,
        notes = notes
    )
    self.create_doc_ES(os.path.basename(self.workspace), data, mute)
```

Ilustración 21 - Método “insert” modificado

Además, se añaden los métodos “insert” correspondientes a los nuevos tipos de documentos que se han incorporado en la base de datos.

5 Documentación del desarrollo

5.1 Módulo de adquisición de Dominios similares

Con este módulo se realiza una búsqueda de dominios similares con diferentes TLDs, permutando el nombre del dominio mediante dnstwister. Por cada dominio similar se comprueba que tenga página web, y en tal caso se lleva a cabo un análisis en VirusTotal y se realiza una captura en Archive.org.

```
meta = {
    'name': 'Phishing',
    'author': 'Rubén Álvarez',
    'version': '',
    'description': '',
    'dependencies': [],
    'required_keys': [],
    'comments': (),
    'query': {'_source': ['domain'], 'query': {'match': {'type': 'domains'}}},
    'options': (
        ('TLDs', os.path.join(BaseModule.data_path, 'TLDs.txt'), False,
         'file containing a list of TLDs'),
    ),
    'files': ['TLDs.txt'],
}
```

Ilustración 22 - Información del módulo de adquisición de Dominios similares

En primer lugar, se almacenan en una lista los TLDs (dominios de primer nivel) que contenga el fichero TLDs.txt, en caso de que exista.

```
def module_run(self, domains):
    tldsFile = self.options['TLDs']
    tlds = []
    if os.path.isfile(tldsFile):
        with open(tldsFile) as f:
            tlds = [x.strip() for x in f.read().splitlines()]
    resolver = self.get_resolver()
    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/53
7.36'}
```

Ilustración 23 - Primer fragmento de código del método module_run() de adquisición de Dominios similares

Para separar la extensión del dominio y así poder concatenar estos TLDs se utiliza el módulo tldextract, que hace uso de la “Public Suffix List”, una lista con todas las extensiones públicas conocidas.

```
for domain in domains:
    domainLabels = tldextract.extract(domain)
    if domainLabels.subdomain:
        domainName = '.'.join(domainLabels[:2])
    else:
        domainName = domainLabels.domain
    domainExt = '.' + domainLabels.suffix
    if not tlds or domainExt not in tlds:
        tlds.append(domainExt)
```

Ilustración 24 - Segundo fragmento de código del método module_run() de adquisición de Dominios similares

De esta manera, una vez separadas las distintas partes o etiquetas del dominio, se concatena el nombre de dominio registrado y los subdominios que tenga delimitados por puntos, sin la extensión. Si la lista de TLDs está vacía o la extensión del dominio no está en dicha lista, entonces su extensión se añade a la misma.

Por cada TLD de la lista este se concatena al resto de etiquetas del dominio, y el string correspondiente al dominio resultante se convierte a hexadecimal pasándoselo codificado en UTF-8 al método binascii.hexlify().

```
for tld in tlds:
    domainTld = domainName + tld
    self.verbose(domainTld)
    domainHex = binascii.hexlify(domainTld.encode())
```

Ilustración 25 - Segundo fragmento de código del método module_run() de adquisición de Dominios similares

Así, se realiza una petición HTTP GET a la API de dnstwister, pasándole el dominio en hexadecimal con el nuevo TLD. El contenido de la respuesta a dicha petición es un documento JSON, con la siguiente estructura:

```
{
  "domain": "upm.es",
  "domain_as_hexadecimal": "75706d2e6573",
  "fuzzy_domains": [
    {
      "domain": "upm.es",
      "domain_as_hexadecimal": "75706d2e6573",
      "fuzz_url": "http://dnstwister.report/api/fuzz/75706d2e6573",
      "fuzzer": "Original*",
      "parked_score_url": "http://dnstwister.report/api/parked/75706d2e6573",
      "resolve_ip_url": "http://dnstwister.report/api/ip/75706d2e6573"
    },
    {
      "domain": "upma.es",
      "domain_as_hexadecimal": "75706d612e6573",
      "fuzz_url": "http://dnstwister.report/api/fuzz/75706d612e6573",
      "fuzzer": "Addition",
      "parked_score_url": "http://dnstwister.report/api/parked/75706d612e6573",
      "resolve_ip_url": "http://dnstwister.report/api/ip/75706d612e6573"
    },
    ...
  ],
  "parked_score_url": "http://dnstwister.report/api/parked/75706d2e6573",
  "resolve_ip_url": "http://dnstwister.report/api/ip/75706d2e6573",
  "url": "http://dnstwister.report/api/fuzz/75706d2e6573"
}
```

Ilustración 26 - Documento JSON con las permutaciones del dominio en dnstwister

Dicho objeto JSON se convierte en un diccionario de Python mediante la función `json.loads()` y se almacena en la variable `jsonData`.

```
url = 'http://dnstwister.report/api/fuzz/' + str(domainHex, 'ascii')
try:
    res = requests.get(url, timeout=5)
    res.raise_for_status()
    jsonData = json.loads(res.text)
```

Ilustración 27 - Tercer fragmento de código del método `module_run()` de adquisición de Dominios similares

El valor asociado a la clave “fuzzy_domains” del diccionario raíz es una lista de diccionarios con la información de los dominios similares generados, que se recorre para obtener el valor de la clave “domain”.

```
similarDomainsDictList = jsonData['fuzzy_domains']
similarDomains = []
for i in range(len(similarDomainsDictList)):
    similarDomain = similarDomainsDictList[i]['domain']
```

Ilustración 28 - Cuarto fragmento de código del método module_run() de adquisición de Dominios similares

Además, por cada dominio similar se comprueba que no sea ninguno de los dominios originales, y se le pasa como parámetro al método tldextract.extract() para separar sus etiquetas. Así se puede comprobar si el dominio registrado del dominio original, en caso de que tenga algún subdominio, se corresponde con el dominio registrado del dominio similar. En tal caso no es de interés al ser legítimo.

```
if similarDomain not in domains:
    similarDomainLabels = tldextract.extract(similarDomain)
    if not (domainLabels.subdomain and similarDomainLabels.domain == domainLabels.domain):
        if similarDomainLabels.subdomain:
            similarDomains.append({'similarDomain': similarDomain, 'isSubdomain': True})
        else:
            similarDomains.append({'similarDomain': similarDomain, 'isSubdomain': False})
```

Ilustración 29 - Quinto fragmento de código del método module_run() de adquisición de Dominios similares

Si no se cumple la anterior condición, el dominio similar se añade a una lista de diccionarios con dominios similares, diferenciando si se trata o no de un subdominio.

Para comprobar si los dominios similares tienen página web, se hace de manera concurrente pasando la lista de dominios similares y el dominio original.

```
self.thread(similarDomains, domain, resolver, headers)
```

Ilustración 30 - Sexto fragmento de código del método module_run() de adquisición de Dominios similares

De esta manera, en cada hilo se comprueba la existencia de registros A correspondientes a cada dominio similar mediante el método resolver.query(). Si existen uno o más registros A, entonces quiere decir que el dominio similar tiene hosting, y en tal caso se almacenan todas las direcciones IP en una lista.

```
def module_thread(self, similarDomainDict, domain, resolver, headers):
    similarDomain = similarDomainDict['similarDomain']
    try:
        answers = resolver.query(similarDomain, 'A')
        ips = []
        for rdata in answers:
            ips.append(rdata.address)
```

Ilustración 31 - Primer fragmento de código del método module_thread() de adquisición de Dominios similares

Debido a que hay dominios que en sus registros DNS tienen un registro A wildcard (*.ejemplo.com), cualquier subdominio de dicho dominio apunta a una dirección IP, por lo que se deben descartar los subdominios similares que sean wildcard.

Para ello, si el dominio similar se trata de un subdominio, se comprueba la existencia del registro A wildcard, (realizando dos intentos para evitar resoluciones fallidas). En caso de que exista, se compara la dirección IP de este registro con las direcciones IP correspondientes a los registros A del dominio registrado. Si hay alguna coincidencia, entonces se descarta al tratarse de un subdominio wildcard.

```
wildcardRecordMatch = False
if similarDomainDict['isSubdomain']:
    wildcardSubdomain = '*' + similarDomain
    attempt = 0
    maxAttempts = 2
    while attempt < maxAttempts:
        try:
            answers = resolver.query(wildcardSubdomain, 'A')
            if answers[0].address in ips:
                wildcardRecordMatch = True
                break
        except Exception:
            attempt += 1
            pass
```

Ilustración 32 - Segundo fragmento de código del método module_thread() de adquisición de Dominios similares

Tras verificar que el dominio similar tiene hosting, se comprueba si tiene página web, ya que es cuando podría tratarse de un phishing. Si tiene página además se comprueba que el código de la respuesta no sea 301 (movido permanentemente) o 302 (movido temporalmente), es decir que no tenga redirección.

```

if not wildcardRecordMatch:
    similarDomainUrl = 'http://' + similarDomain + '/'
    res = requests.head(similarDomainUrl, headers=headers, timeout=5)
    res.raise_for_status()
    if res.status_code not in {301, 302}:
        try:
            positives = self.virustotalScan(similarDomain, headers)
            snapshotURL, screenshotURL = self.archiveSave(similarDomain,
                headers)
            self.alert(similarDomain + '\n' + str(ips) + '\n' +
            'Positives: ' + str(positives) + '\n' + snapshotURL + '\n' +
            + screenshotURL)
            self.insert_similarDomains(original_domain=domain,
                similar_domain=similarDomain, ips=ips, vt_positives=positives
                , snapshot=snapshotURL, screenshot=screenshotURL)

```

Ilustración 33 - Tercer fragmento de código del método module_threat() de adquisición de Dominios similares

Hechas todas las comprobaciones anteriores, se llama a los métodos para analizar la página web en VirusTotal y guardar una captura de la misma en Archive.org.

Así, en el método virustotalScan() se analiza la página del dominio similar en VirusTotal. Para ello, se realiza una petición HTTP POST a “<https://www.virustotal.com/ui/urls/>” junto con el hash SHA-256 de la URL correspondiente al dominio similar, utilizando el módulo hashlib para generar dicho hash.

```

def virustotalScan(self, domain):
    domainUrl = 'http://' + domain + '/'
    domainEncoded = domainUrl.encode('utf-8')
    url = 'https://www.virustotal.com/ui/urls/' + hashlib.sha256(domainEn
    coded).hexdigest() + '/analyse'
    res = requests.post(url, timeout=5)
    res.raise_for_status()

```

Ilustración 34 - Primer fragmento de código del método virustotalScan() de adquisición de Dominios similares

El contenido de la respuesta es un objeto JSON, que se convierte en un diccionario de Python mediante la función res.json() y se almacena en la variable jsonData. En la clave “data” del diccionario raíz hay otro diccionario anidado, donde se encuentra la clave “id”, que se concatena a la URL “<https://www.virustotal.com/ui/analyses/>” para obtener el resultado del análisis del dominio similar.

```

jsonData = res.json()
url = 'https://www.virustotal.com/ui/analyses/' + jsonData['data']['id']
time.sleep(10)
res = requests.get(url, timeout=5)
res.raise_for_status()

```

Ilustración 35 - Segundo fragmento de código del método virustotalScan() de adquisición de Dominios similares

Con el fin de no realizar demasiadas peticiones a VirusTotal, se espera 10 segundos, que es el tiempo medio que tarda en completarse el análisis de una URL. Cuando el análisis ha finalizado, el resultado devuelto en formato JSON contiene “completed” en el clave “status”. Hasta entonces el valor de dicha clave es “queued”, por lo que se comprueba si ha finalizado un máximo de 5 veces, esperando 1 segundo entre cada intento.

```

analysisData = res.json()
analysisStatus = analysisData['data']['attributes']['status']
attempt = 0
maxAttempts = 5
while 'queued' in analysisStatus and attempt < maxAttempts:
    time.sleep(1)
    res = requests.get(url, timeout=5)
    res.raise_for_status()
    analysisData = res.json()
    analysisStatus = analysisData['data']['attributes']['status']
    attempt += 1

```

Ilustración 36 - Tercer fragmento de código del método virustotalScan() de adquisición de Dominios similares

La estructura del objeto JSON que contiene el resultado del análisis es la siguiente:

```
{
  "data": {
    "attributes": {
      "date": 1586020267,
      "results": {
        "ADMINUSLabs": {
          "category": "harmless",
          "engine_name": "ADMINUSLabs",
          "method": "blacklist",
          "result": "clean"
        },
        ...
      },
      "stats": {
        "harmless": 71,
        "malicious": 0,
      }
    }
  }
}
```

```

        "suspicious": 0,
        "timeout": 0,
        "undetected": 6
    },
    "status": "completed"
},
"id": "u-
85a5e968462a23e74af03dcfb45f27fe73456902e57f590e7e7dc88e02bb2d67-
1586020267",
"type": "analysis"
},
"meta": {
    "url_info": {
        "id": "85a5e968462a23e74af03dcfb45f27fe73456902e57f590e7e7dc8
8e02bb2d67",
        "url": "http://upm.es/"
    }
}
}

```

Ilustración 37 - Documento JSON con el resultado del análisis en VirusTotal

Una vez está disponible el resultado del análisis, que se guarda como diccionario de Python en la variable analysisData, se accede al diccionario asociado a la clave “stats”. En dicho diccionario se encuentran las estadísticas del análisis, del cual se obtienen los valores de las claves “malicious” y “suspicious”, cuya suma es el número de positivos resultantes.

```

analysisStats = analysisData['data']['attributes']['stats']
positives = int(analysisStats['malicious']) + int(analysisStats['suspicio
us'])
return positives

```

Ilustración 38 - Cuarto fragmento de código del método virustotalScan() de adquisición de Dominios similares

Por otra parte, en el método archiveSave() se guarda una captura en Archive.org de la página web del dominio similar. Para ello, con el método requests.post() se realiza una petición HTTP POST a “<https://web.archive.org/save/>”, pasándole por parámetro el diccionario “data”. Este diccionario contiene los datos a enviar, correspondientes al dominio similar y la opción de hacer una captura de la página web activada.

```
def archiveSave(self, domain):
    data = {
        'url': domain,
        'capture_screenshot': 'on'
    }
    res = requests.post('https://web.archive.org/save/', data=data, timeout=5)
    res.raise_for_status()
```

Ilustración 39 - Primer fragmento de código del método archiveSave() de adquisición de Dominios similares

La URL donde se archiva la página del dominio similar tiene el siguiente formato:

`https://web.archive.org/web/<fecha y hora>/<dominio>`

La fecha y hora se representan de manera conjunta en el formato YYYYMMDDHHMMSS, siendo la hora en UTC.

Como se sabe la fecha en la que se ha archivado la página, pero se desconoce la hora, se realiza una petición HTTP HEAD con la URL anterior sin que contenga la hora. De esta manera, se obtiene la URL a la que redirige, que es la URL completa incluyendo la hora.

```
today = datetime.datetime.utcnow().strftime('%Y%m%d')
url = 'https://web.archive.org/web/' + today + '/' + domain
time.sleep(25)
res = requests.head(url, timeout=5, allow_redirects=True)
res.raise_for_status()
snapshotURL = res.url
```

Ilustración 40 - Segundo fragmento de código del método archiveSave() de adquisición de Dominios similares

Para obtener la URL correspondiente a la captura de la página del dominio similar se procede de la misma manera, pues dicha URL tiene el siguiente formato:

`https://web.archive.org/web/<fecha y hora>if_/http://web.archive.org/screenshot/<dominio>`

```
url = 'https://web.archive.org/web/' + today + 'if_/http://web.archive.org/screenshot/' + domain
res = requests.head(url, timeout=5, allow_redirects=True)
res.raise_for_status()
snapshotURL = res.url
return (snapshotURL, screenshotURL)
```

Ilustración 41 - Tercer fragmento de código del método archiveSave() de adquisición de Dominios similares

5.2 Módulo de adquisición de Direcciones de correo electrónico

```
meta = {
    'name': 'Emails',
    'author': 'Rubén Álvarez',
    'version': '',
    'description': '',
    'dependencies': [],
    'files': [],
    'required_keys': [],
    'comments': (),
    'query': {'_source': ['domain'], 'query': {'match': {'type': 'domains'}}},
    'options': (
        ('date', None, False, 'Last hour, day, week, month or year: h, d, w, m, y'),
    ),
}
}
```

Ilustración 42 - Información del módulo de adquisición de Direcciones de correo electrónico

Con este módulo se realiza una búsqueda de un dominio en Google para adquirir las direcciones de correo electrónico que se encuentren en los resultados de la búsqueda, ya sea en una página web o en un documento PDF, siendo el input dicho dominio.

En primer lugar, con la función `requests.get()` se realiza una petición HTTP GET de la URL correspondiente a la búsqueda en Google, pasando por parámetro los headers.

```
def module_run(self, domains):
    date = self.options['date']
    headers = {'User-Agent':
               'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
               (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36'}
    for domain in domains:
        emailRegex = re.compile(r'[a-zA-ZñÑáéíóúÁÉÍÓÚ0-9._%-]+@'
                               + re.escape(domain))
        url = 'https://www.google.es/search?q=' + '"' + '@' + domain + '"' +
              '&filter=0'
        if date:
            url += '&tbs=qdr:' + date
        while url:
            try:
                res = requests.get(url, headers=headers, timeout=5)
                res.raise_for_status()
```

Ilustración 43 - Primer fragmento de código del método `module_run()` de adquisición de Direcciones de correo electrónico

Los parámetros del query string de la URL son los siguientes:

- “@ejemplo.com” – La dirección de correo electrónico sin la parte local, es decir solo el carácter arroba y el dominio. Además, entre comillas para hacer una búsqueda exacta.
- source=Int&tbs=qdr:m – Para mostrar solo los resultados del último mes.
- filter=0 – Para que muestren todas las entradas sin que se omitan resultados de la búsqueda.

Con el módulo Beautiful Soup se parsea el código HTML de una página web, en nuestro caso necesitamos extraer los enlaces de la página correspondiente a la búsqueda realizada en Google. Para ello, se pasan como parámetros a bs4.BeautifulSoup() el atributo text de la respuesta del request.get(), que contiene el HTML, y el parser HTML que se quiera utilizar. El objeto BeautifulSoup que devuelve se guarda en la variable “soup”.

```
soup = bs4.BeautifulSoup(res.text, features='lxml')
```

Ilustración 44 - Segundo fragmento de código del método module_run() de adquisición de Direcciones de correo electrónico

Hemos decidido utilizar el parser lxml debido a que es más rápido que el incluido en la librería estándar de Python. Para hacer uso de este es necesario instalarlo con el comando pip install lxml.

Como necesitamos extraer los enlaces de los resultados de la búsqueda, hay que llamar al método soup.select() pasándole el selector CSS “.r > a:first-of-type”, comprobando que se haya encontrado la búsqueda exacta con las comillas.

```
if not notFound:
    results = soup.select('.r > a:first-of-type')
    ...
else:
    self.verbose('Not Found.')
    break
```

Ilustración 45 - Tercer fragmento de código del método module_run() de adquisición de Direcciones de correo electrónico

Con este selector CSS se busca el primer elemento “<a>” que esté directamente dentro de los elementos que usen la clase CSS denominada “r”.

De esta manera, soup.select() devuelve una lista de elementos que coinciden con el selector, de los cuales se obtiene el atributo “href”, que contiene la URL correspondiente al resultado de la búsqueda.

```

for x in range(len(results)):
    try:
        resultURL = results[x].get('href')
        res = requests.get(resultURL, headers=headers, timeout=5)
        res.raise_for_status()

```

Ilustración 46 - Cuarto fragmento de código del método module_run() de adquisición de Direcciones de correo electrónico

Entre los resultados de la búsqueda puede haber tanto páginas web como documentos PDF, de los cuales queremos obtener los emails que puedan contener, por lo que es necesario diferenciarlos para parsearlos de la manera adecuada.

Para identificar si se trata de un documento PDF, simplemente se busca el substring “.pdf” en el string correspondiente a la URL, convirtiéndolo además a minúsculas por si el substring estuviera en mayúsculas.

```

if '.pdf' in resultURL.lower():
    f = io.BytesIO(res.content)
    pdfReader = PyPDF2.PdfFileReader(f)

```

Ilustración 47 - Quinto fragmento de código del método module_run() de adquisición de Direcciones de correo electrónico

En el caso de que se trate de un documento PDF, se utiliza el módulo PyPDF2 para extraer el texto de este. Para no tener que almacenar el documento en disco, se guarda en memoria creando un stream binario con `io.BytesIO()` y se representa mediante un objeto `PdfFileReader()`.

Después se obtiene el número de páginas, extrayendo el texto por cada página, y se buscan emails con la expresión regular “[a-zA-ZñÑáéíóúÁÉÍÓÚ0-9._%-]+@”. Para evitar duplicados, cada email se guarda en “emailList” tras comprobar que no esté ya en la lista.

```

numPages = pdfReader.getNumPages()
emailList = []
for y in range(numPages):
    pageObj = pdfReader.getPage(y)
    page = pageObj.extractText()
    emails = emailRegex.findall(page)
    if emails is not None:
        for email in emails:
            if email not in emailList:
                self.alert(email + '\n' + resultURL)
                self.insert_emails(email=email, source=resultURL)
                emailList.append(email)

```

Ilustración 48 - Sexto fragmento de código del método module_run() de adquisición de Direcciones de correo electrónico

Por otra parte, si el resultado de la búsqueda de Google no se trata de un documento PDF sino de una página web, se buscan emails con regex en el contenido de la página web. Además, mediante el método `requests.utils.unquote()` se elimina la codificación de las URLs, denominada código porcento.

```
else:
    emails = emailRegex.findall(requests.utils.unquote(res.text))
    if emails is not None:
        emailList = []
        for email in emails:
            if email not in emailList:
                self.alert(email + '\n' + resultURL)
                self.insert_emails(email=email, source=resultURL)
                emailList.append(email)
```

Ilustración 49 - Séptimo fragmento de código del método `module_run()` de adquisición de Direcciones de correo electrónico

Por último, se busca mediante BeautifulSoup el elemento “`<a>`” cuyo id es “`pnnext`”, que se corresponde con el enlace a la página siguiente:

```
url = soup.find('a', id='pnnext')
if url:
    url = 'https://www.google.es' + url['href']
else:
    self.verbose('The End.')
    break
```

Ilustración 50 - Octavo fragmento de código del método `module_run()` de adquisición de Direcciones de correo electrónico

Si existe dicho elemento, entonces significa hay siguiente página, y por tanto se obtiene el valor del atributo “`href`” del elemento. Como no contiene la parte de la URL “`https://www.google.es/`”, hay que concatenarla para formar la URL completa. Por otro lado, en caso de que no exista dicho elemento significa que no hay página siguiente, y se sale del loop.

5.3 Módulo de adquisición de Direcciones de correo en Hunter

```
meta = {
    'name': 'emailsHunter',
    'author': 'Rubén Álvarez',
    'version': '',
    'description': '',
    'dependencies': [],
    'files': [],
    'required_keys': ['hunter_api'],
    'comments': (),
    'query': {'_source': ['domain'], 'query': {'match': {'type': 'domains'}}},
    'options': (),
}
```

Ilustración 51 - Información del módulo de adquisición de Direcciones de correo en Hunter

Con este módulo se realiza una búsqueda de direcciones de correo electrónico a partir de un dominio utilizando la API de Hunter. Este módulo se complementa con el de adquisición de direcciones de correo en buscadores debido a que Hunter lleva tiempo indexando direcciones de correo, mientras que los buscadores muestran un número limitado de resultados, por lo que no se pueden obtener todos los correos existentes.

En primer lugar, se realiza una petición HTTP GET a la API de Hunter y la respuesta se almacena en la variable res. Con res.raise_for_status() se lanza una excepción si la petición HTTP es errónea.

```
def module_run(self, domains):
    apiKey = self.keys['hunter_api']
    for domain in domains:
        url = 'https://api.hunter.io/v2/domain-search?domain='
        + domain + '&api_key=' + apiKey + '&limit=100'
        try:
            res = requests.get(url, timeout=5)
            res.raise_for_status()
```

Ilustración 52 - Primer fragmento de código del método module_run() de adquisición de Direcciones de correo en Hunter

Los parámetros del query string de la URL son los siguientes:

- domain=ejemplo.com – El dominio del cual se quieren buscar las direcciones de correo.
- api_key=X – La API key de hunter.
- &limit=100 – El número máximo de direcciones de correo a devolver, por defecto es 10 y el límite es 100 por cada consulta.

El atributo “res.text” es un string que contiene los datos de la respuesta en formato JSON, con la siguiente estructura:

```
{  
    "data": {  
        "domain": "upm.es",  
        "disposable": false,  
        "webmail": false,  
        "accept_all": false,  
        "pattern": "{first}.{last}",  
        "organization": "Universidad Politécnica De Madrid",  
        "country": null,  
        "state": null,  
        "emails": [  
            {  
                "value": "jesus.sanchez1@upm.es",  
                "type": "personal",  
                "confidence": 94,  
                "sources": [  
                    {  
                        "domain": "ocw.upm.es",  
                        "uri": "http://ocw.upm.es/course/teleformacion-2012",  
                        "extracted_on": "2020-04-04",  
                        "last_seen_on": "2020-04-04",  
                        "still_on_page": true  
                    },  
                    ...  
                ],  
                "first_name": "Jesús",  
                "last_name": "Sánchez Lopez",  
                "position": null,  
                "seniority": null,  
                "department": null,  
                "linkedin": null,  
                "twitter": null,  
                "phone_number": null  
            },  
            ...  
        ]  
    },  
    "meta": {  
        "results": 3474,  
        "limit": 100,  
        "offset": 0,  
        "params": {  
            "domain": "upm.es",  
            "company": null,  
            "type": null,  
            "seniority": null,  
            "department": null  
        }  
    }  
}
```

Ilustración 53 - Documento JSON con los datos de las direcciones de correo obtenidas con la API de Hunter

Mediante la función `json.loads()` se realiza la conversión del string JSON en un diccionario de Python, que se guarda en la variable `jsonData`.

```
jsonData = json.loads(res.text)
```

Ilustración 54 - Segundo fragmento de código del método `module_run()` de adquisición de Direcciones de correo en Hunter

Del diccionario asociado a la clave “meta” se obtiene el valor de la clave “results”, que se corresponde al número total de direcciones de correo encontradas. Dado que por cada consulta se devuelven un máximo de 100 direcciones de correo, se harán las consultas necesarias hasta que se hayan obtenido todas las encontradas.

En la clave “data” del diccionario raíz hay otro diccionario anidado, donde se encuentra la clave “emails”, que a su vez contiene una lista de diccionarios con la información de las direcciones de correo electrónico encontradas.

```
numEmails = 0
while numEmails < jsonData['meta']['results']:
    emails = jsonData['data']['emails']
    for i in range(len(emails)):
        email = emails[i]['value']
        sources = emails[i]['sources']
```

Ilustración 55 - Tercer fragmento de código del método `module_run()` de adquisición de Direcciones de correo en Hunter

De esta manera, por cada diccionario de la lista se obtienen los valores de las claves “value” y “sources”. La primera se corresponde con la dirección de correo, mientras que la segunda contiene otra lista con la información de las fuentes de localización de dicha dirección. Esta lista se recorre para obtener el valor asociado a la clave “uri”, que es la URL de la fuente.

```
for j in range(len(sources)):
    sourceURL = sources[j]['uri']
    self.alert(email + '\n' + sourceURL)
    self.insert_emails(email=email, source=sourceURL)
```

Ilustración 56 - Cuarto fragmento de código del método `module_run()` de adquisición de Direcciones de correo en Hunter

El número de direcciones de correo obtenidas se suman al contador almacenado en la variable `numEmails`, y si dicho contador es menor que el número total de direcciones de correo encontradas, se realiza la petición HTTP GET con el parámetro “offset” en el query string de la URL.

El parámetro “offset” establece el número de direcciones de correo a saltarse del total encontradas, y por defecto es 0. Por lo tanto, para obtener las direcciones restantes se le asigna el contador (numEmails) de las que ya se han obtenido.

```
numEmails += len(emails)
if numEmails < jsonData['meta']['results']:
    url += '&offset=' + str(numEmails)
    res = requests.get(url, timeout=5)
    res.raise_for_status()
    jsonData = json.loads(res.text)
```

Ilustración 57 - Quinto fragmento de código del método module_run() de adquisición de Direcciones de correo en Hunter

5.4 Módulo de adquisición de Noticias

Con este módulo se realiza una búsqueda de un nombre de marca en Google para adquirir las noticias que se encuentren en el apartado Noticias, siendo el input dicho término de búsqueda.

```
meta = {
    'name': 'News',
    'author': 'Rubén Álvarez',
    'version': '',
    'description': '',
    'dependencies': [],
    'files': [],
    'required_keys': [],
    'comments': (),
    'query': {'_source': ['name'], 'query': {'match': {'type': 'brands'}}},
    'options': (
        ('date', None, False, 'Last hour, day, week, month or year: h, d, w, m, y'),
        ('country', False, False, 'ES'),
    ),
}
```

Ilustración 58 - Información del módulo de adquisición de Noticias

En primer lugar, con la función `requests.get()` se realiza una petición HTTP GET de la URL correspondiente a la búsqueda en Google, pasando por parámetro los headers.

```
def module_run(self, brands):
    date = self.options['date']
    country = self.options['country']
    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36'}
    for brand in brands:
        url = 'https://www.google.es/search?q=' + ' ' + brand + ' ' +
              '&tbs=nws'
        if date and country:
            url += '&tbs=qdr:' + date + ',ctr:countryES&cr=countryES'
        elif date:
            url += '&tbs=qdr:' + date
        elif country:
            url += '&tbs=ctr:countryES&cr=countryES'
```

Ilustración 59 - Primer fragmento de código del método `module_run()` de adquisición de Noticias

Los parámetros del query string de la URL son los siguientes:

- “Nombre de marca” - Nombre de marca entre comillas para buscar el término exacto.
- tbm=nws – Para buscar en el apartado Noticias.
- source=Int&tbs=qdr:m – Para mostrar solo los resultados del último mes.
- ctr:countryES&source=Int&cr=countryES – Para mostrar solo páginas de España.

Como es necesario extraer los enlaces y la fecha de los resultados de la búsqueda, se llama al método soup.select() pasándole el selector CSS de la clase “.l” y el de la clase “.dhIWPD”

```
while url:
    try:
        res = requests.get(url, headers=headers, timeout=5)
        res.raise_for_status()
        soup = bs4.BeautifulSoup(res.text, features='lxml')
        notFound = soup.select('.obp')
        if not notFound:
            news = soup.select('.l')
            newsDates = soup.select('.dhIWPD')
```

Ilustración 60 - Segundo fragmento de código del método module_run() de adquisición de Noticias

De cada elemento de la lista de elementos coincidentes con dicho selector se obtiene el atributo “href”, cuyo valor es la URL de la noticia, realizando una petición GET de dicha URL.

```
for i in range(len(news)):
    try:
        newsURL = news[i].get('href')
        newsDate = re.sub(r'.*-', '', newsDates[i].getText())
        dateDMY = dateparser.parse(newsDate).strftime("%d/%m/%Y")
        res = requests.get(newsURL, headers=headers, timeout=5)
        res.raise_for_status()
```

Ilustración 61 - Tercer fragmento de código del método module_run() de adquisición de Noticias

Para obtener la fecha de las noticias, del string donde se encuentra es necesario eliminar cualquier carácter, correspondiente a la fuente de la noticia, hasta encontrar un guion. Para ello, se pasa la expresión regular al método re.sub(). Después, con el método dateparser.parse() se parsea la fecha y con el método strftime() se convierte a formato DMY.

El texto de la respuesta se parsea con BeautifulSoup para buscar el título de la noticia en el documento HTML. De esta manera, mediante soupWeb.find() se localiza el elemento “<title>” pasándole el selector CSS correspondiente a dicha etiqueta:

```
soupWeb = bs4.BeautifulSoup(res.text, features='lxml')
title = soupWeb.find('title')
if title is not None:
    if title.getText() != '':
        newsTitle = title.getText().strip()
    else:
        newsTitle = news[i].getText()
else:
    newsTitle = news[i].getText()
self.alert(newsTitle + '\n' + newsURL + '\n' + dateDMY)
self.insert_news(title=newsTitle, url=newsURL, date=dateDMY)
```

Ilustración 62 - Cuarto fragmento de código del método module_run() de adquisición de Noticias

Si el elemento “<title>” existe y no está vacío, es decir la página web tiene un título definido, entonces se obtiene el texto de dicho elemento eliminando los espacios que pueda haber con la función strip(). En caso contrario, se obtiene el título de la noticia de los resultados de la búsqueda en Google. Cabe destacar que es preferible obtener el título de la propia página web en vez de los resultados de la búsqueda, pues si la longitud de este supera un límite de caracteres Google no lo muestra completo.

Por último, se comprueba si hay siguiente página en la búsqueda realizada en Google de la misma manera que en el módulo de adquisición de direcciones de correo electrónico.

5.5 Módulo de adquisición de Documentos

Con este módulo se realiza una búsqueda de documentos en Google y se extraen sus metadatos.

```
meta = {
    'name': 'Docs',
    'author': 'Rubén Álvarez',
    'version': '',
    'description': '',
    'dependencies': [],
    'files': [],
    'required_keys': [],
    'comments': (),
    'query': {'_source': ['domain'], 'query': {'match': {'type': 'domains'}}},
    'options': (
        ('date', None, False, 'Last hour, day, week, month or year: h, d, w, m, y'),
    ),
}
}
```

Ilustración 63 - Información del módulo de adquisición de Documentos

El funcionamiento es similar al módulo de adquisición de Emails explicado anteriormente. En este caso, los parámetros del query string de la URL correspondiente a la búsqueda en Google son los siguientes:

- site:ejemplo.com – Para limitar la búsqueda al dominio dado.
- ext:pdf OR ext:txt OR [...] – Para filtrar por extensiones de documento.
- filter=0 – Para que muestren todas las entradas sin que se omitan resultados de la búsqueda.

```
def module_run(self, domains):
    date = self.options['date']
    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36'}
    for domain in domains:
        url = 'https://www.google.es/search?q=' + 'site:' + domain +
              ' ext:pdf OR ext:txt OR ext:rtf OR ext:xml OR ext:csv OR ext:doc
              OR ext:docx OR ext:xls OR ext:xlsx OR ext:ppt OR ext:pptx OR ext:
              pps OR ext:ppsx OR ext:odt OR ext:ods OR ext:odp' + '&filter=0'
        if date:
            url += '&tbs=qdr:' + date
```

Ilustración 64 - Primer fragmento de código del método module_run() de adquisición de Documentos

Las URLs de los resultados de la búsqueda correspondientes a documentos se extraen de la misma manera que en el módulo de adquisición de Emails, realizando después una petición HTTP GET de cada URL.

Para identificar si se trata de un documento PDF, simplemente se busca el substring “pdf” en la cabecera “Content-Type”. En dicho caso, se guarda en memoria creando un stream binario con io.BytesIO() y mediante el método getDocumentInfo() del módulo PyPDF2 se extraen los metadatos del documento PDF.

```
if 'pdf' in res.headers['Content-Type']:
    f = io.BytesIO(res.content)
    pdfReader = PyPDF2.PdfFileReader(f)
    pdfInfo = pdfReader.getDocumentInfo()
```

Ilustración 65 - Segundo fragmento de código del método module_run() de adquisición de Documentos

La información se devuelve en un objeto de tipo DocumentInformation, cuyos atributos se recorren guardando cada metadato en un diccionario, si el valor del metadato no está vacío.

```
metadata = []
for meta in pdfInfo:
    if pdfInfo[meta]:
        metadata[meta.strip('/')] = pdfInfo[meta]
```

Ilustración 66 - Tercer fragmento de código del método module_run() de adquisición de Documentos

Los datos obtenidos se indexan en Elasticsearch mediante el método insert_documents(). En el caso de que el documento sea un PDF, se incluye el dominio de input, la URL del documento y los metadatos, mientras que para cualquier otro tipo de documento se incluyen solo los dos primeros.

```
if 'pdf' in res.headers['Content-Type']:
    ...
    self.alert(docURL + '\n' + str(metadata))
    self.insert_documents(domain=domain, url=docURL, metadata=metadata)
else:
    self.alert(docURL)
    self.insert_documents(domain=domain, url=docURL)
```

Ilustración 67 - Cuarto fragmento de código del método module_run() de adquisición de Documentos

Por último, se comprueba si hay siguiente página en la búsqueda realizada en Google de igual forma que en el módulo de adquisición de direcciones de correo electrónico.

5.6 Módulo de búsqueda de coincidencias en servicios de compartición de texto online

Este módulo busca la aparición del nombre de marca en pastes compartidos en los servicios online de compartición de texto Pastebin y GitHub Gist. De estas dos webs se obtienen tres datos: el título, la fecha y el contenido de los pastes.

```
meta = {
    'name': 'Posts',
    'author': 'Rodrigo Baladrón',
    'version': '',
    'description': '',
    'dependencies': [],
    'files': ['geckodriver.exe'],
    'required_keys': [],
    'comments': (),
    'query': {'_source': ['name'], 'query': {'match': {'type': 'brands'}}},
    'options': (
        ('driver', os.path.join(BaseModule.data_path, 'geckodriver.exe'),
         True, 'path to selenium driver'),
    ),
}
```

Ilustración 68 - Información del módulo de adquisición de Pastes

Para llevar a cabo esta búsqueda mediante técnicas de web scraping se hace uso de los módulos BeautifulSoup y Selenium. A los métodos que buscan en las webs de Pastebin y Gist se les pasa el nombre de marca y los headers.

```
def module_run(self, brands):
    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/53
7.36'}
    for brand in brands:
        self.pastebin(brand, headers)
        self.gistGithub(brand, headers)
```

Ilustración 69 - Código del método module_run() del módulo de adquisición de Pastes

5.6.1 Pastebin

El módulo comienza buscando en Pastebin, para ello es necesario usar el módulo Selenium, ya que los resultados de la búsqueda se están generando dinámicamente a través de JavaScript, y por lo tanto no se encuentran en el código HTML.

Usando Selenium se solventa este problema ya que, en vez de descargar la web, la renderiza. Por otra parte, es un inconveniente, ya que no se puede empezar a parsear la web hasta que no se ha abierto el navegador y se ha renderizado la web, por lo que tarda más que usando BeautifulSoup.

```
def pasteбин(self, brand, headers):
    driverPath = self.options['driver']
    options = Options()
    options.add_argument('--headless')
    driver = webdriver.Firefox(executable_path=driverPath, options=options)
    url = 'https://paste-bin.com/search?q=' + ' "' + brand + '"'
```

Ilustración 70 - Primer fragmento de código del método pasteбин() de adquisición de Pastes

Se importa el submódulo webdriver de Selenium y se usa el navegador Firefox para renderizar la web. Con la opción “--headless” se consigue que se ejecute el navegador en segundo plano.

La URL que se le pasa al navegador para que haga la petición HTTP GET tiene unos parámetros determinados. Con el parámetro “/search?q=” se realiza una búsqueda en la web. Por otro lado, en la variable “brand” se recibe el nombre de marca que se desee incluir en la búsqueda. Es importante que el nombre de marca vaya entre comillas, sobre todo si tiene más de una palabra. Con las comillas se realiza una búsqueda exacta. Esta variable se usará tanto para PasteBin como para Gist.

Una vez abierta la web con el método de Selenium “driver.get(url)”, se procede a adquirir los datos que nos interesan de la búsqueda. En este caso se pretenden obtener todos los títulos, URLs, fecha y contenido raw de los resultados de búsqueda. Para ello se analiza el código de la web con Selenium, teniendo en cuenta que PasteBin, como otras muchas webs, no ofrece todos los resultados de búsqueda en una sola página, sino que muestra 10 resultados por página como máximo.

Para solucionar esto, se debe conseguir avanzar página a página. Se podría hacer una petición HTTP GET de la página siguiente, pero no es posible conocer esta URL a partir del código de la página actual por el modo en que está desarrollada la web. Por lo tanto, esta opción no es viable y no queda más remedio que ir a la página siguiente haciendo clic en el número siguiente. El módulo de Selenium permite esta interacción con la web.

Para hacer clic en la página siguiente es necesario saber cuál es la página actual y cuál es la siguiente, ya que PasteBin no tiene botón siguiente, solo números de páginas.

Se ha optado por buscar todos los botones de página y recorrerlos para clicar sobre ellos después.

Los botones correspondientes a las páginas se obtienen buscando los elementos HTML cuya clase css es “gsc-cursor-page”. Esta búsqueda se realiza con el método “find_elements_by_class_name” de Selenium.

```
try:
    driver.get(url)
    pages = driver.find_elements_by_class_name('gsc-cursor-page')
```

Ilustración 71 - Segundo fragmento de código del método pastebin() de adquisición de Pastes

Como se ve en la web, el elemento que contiene el número de página es el elemento “div.gsc-cursor-page”.

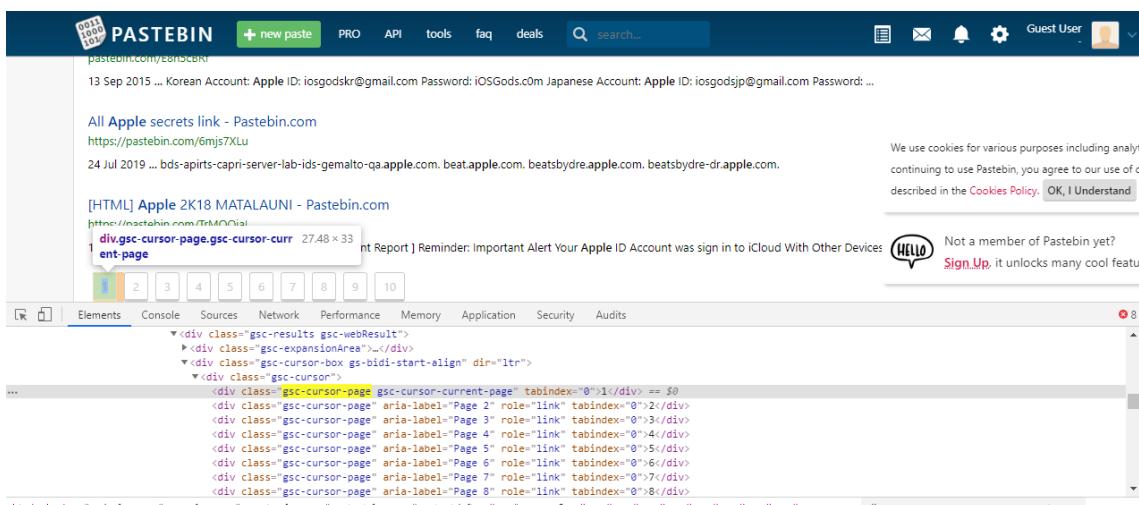


Ilustración 72 - Elemento HTML con el número de página

Por otro lado, los resultados de la página se obtienen de los elementos con la clase CSS “gs-per-result-labels”.

```
urls = []
i = 0
while True:
    elems = driver.find_elements_by_class_name('gs-per-result-labels')
```

Ilustración 73 - Tercer fragmento de código del método pastebin() de adquisición de Pastes

Este elemento no es visible gráficamente en la web y contiene la URL de cada resultado.

The screenshot shows a search results page on Pastebin. The results list various deals, such as an Apple iPad Mini 2 Retina Display 32GB - Space Grey (Certified ...), a PowerTime Apple Watch Charging Dock with 3 USB Ports, and a MobyFox 38mm Apple Watch Band (Flamingos). The developer tools' Elements tab is open, highlighting an HTML element containing a URL: `<div class="gs-per-result-labels" url="https://deals.pastebin.com/sales/ipad-mini-2-retina-display-32gb-4g-unlocked" style="display: none;">Space Grey (Certified Refurbished). This Grade B Refurbished iPad Mini Provides You with a Powerful A7 Chip, Fast`.

Ilustración 74 - Elemento HTML que contiene la URL de cada resultado

Esta URL se obtiene con el método “`get_attribute()`”. Una vez que se han obtenido las URLs, se desechan las que provienen del subdominio `deals.pastebin.com` ya que son enlaces al sitio de compras de Pastebin. Con la expresión regular `“^(?:http|https)://pastebin.+”` se obtienen coincidencias de URLs correspondientes al dominio de `pastebin.com` gracias al módulo `re` de Python.

```
for elem in elems:
    url = elem.get_attribute('url')
    if url is not None:
        urlRegex = re.compile(r'^(?:http|https)://pastebin.+')
        urls += urlRegex.findall(url)
```

Ilustración 75 - Cuarto fragmento de código del método `pastebin()` de adquisición de Pastes

Es necesario realizar la búsqueda de los botones por cada página, ya que al pasar de página los elementos HTML de los botones no son los mismos. Como se avanza accediendo a la página siguiente, `i+1`, cuando se llega a la última página hay que omitir el paso de pasar de página.

```
pages = driver.find_elements_by_class_name('gsc-cursor-page')
if i != len(pages)-1 and len(pages) != 0:
    pages[i+1].click()
    i += 1
    time.sleep(1)
```

Ilustración 76 - Quinto fragmento de código del método `pastebin()` de adquisición de Pastes

Una vez obtenidas las URLs se van haciendo peticiones HTTP GET con el módulo request para posteriormente extraer de ellas los datos de título (title), fecha (date) y contenido (content). En este caso los elementos que queremos obtener no son generados dinámicamente, y se puede analizar el código fuente de la web de manera estática con el módulo BeautifulSoup. Obtener un elemento del código fuente de la página es más rápido que obtenerlo abriendo el navegador por ello siempre que se pueda es conveniente usar este método.

```
if urls:
    for url in urls:
        try:
            res = requests.get(url, headers=headers, timeout=5)
            res.raise_for_status()
            pasteParse = bs4.BeautifulSoup(res.text, 'lxml')
```

Ilustración 77 - Sexto fragmento de código del método pastebin() de adquisición de Pastes

El título se obtiene a partir del elemento title. Buscando con el método select() de BeautifulSoup se obtiene el elemento “title”. Accediendo al atributo “text” se obtiene solo el texto del elemento, ya que la etiqueta y el resto de los atributos no son de utilidad en este caso.

```
title = pasteParse.find('title')
pasteTitle = title.text
```

Ilustración 78 - Séptimo fragmento de código del método pastebin() de adquisición de Pastes

La fecha se obtiene del elemento “span” con atributo “title”.

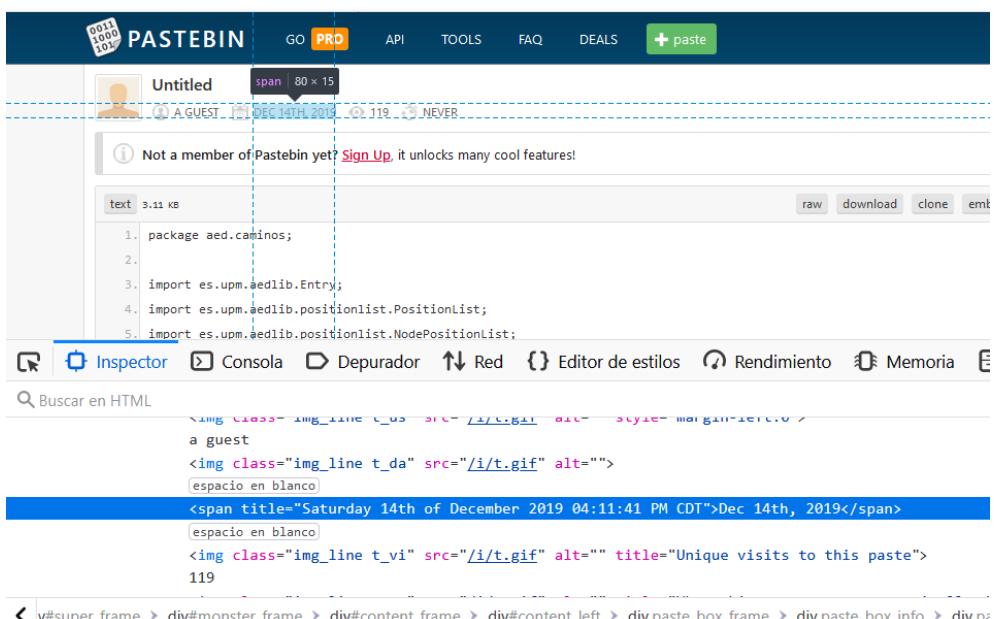


Ilustración 79 - Elemento HTML que contiene la fecha del resultado

De este elemento también se obtiene solo el texto. Sin embargo, en este caso se va a modificar la fecha para que cumpla con el formato estándar.

Para ello, en primer lugar, se usa el módulo de Python “dateparser” que analiza los datos de una fecha soportando diferentes formatos y los devuelve como atributos de un objeto “datetime”.

En segundo lugar, se da formato a la fecha gracias al método “strftime()” del módulo “datetime” pasándole los siguientes parámetros.

- “%d” muestra el día del mes
- “%m” muestra el número de mes
- “%Y” muestra el año completo

```
date = pasteParse.select('.paste_box_line2 > span[title]')
dateParse = dateparser.parse(date[0].getText())
dateDMY = dateParse.strftime("%d/%m/%Y")
```

Ilustración 80 - Octavo fragmento de código del método pastebin() de adquisición de Pastes

Después del campo fecha se obtiene el contenido del paste del elemento HTML “textareaid="paste_code"]”

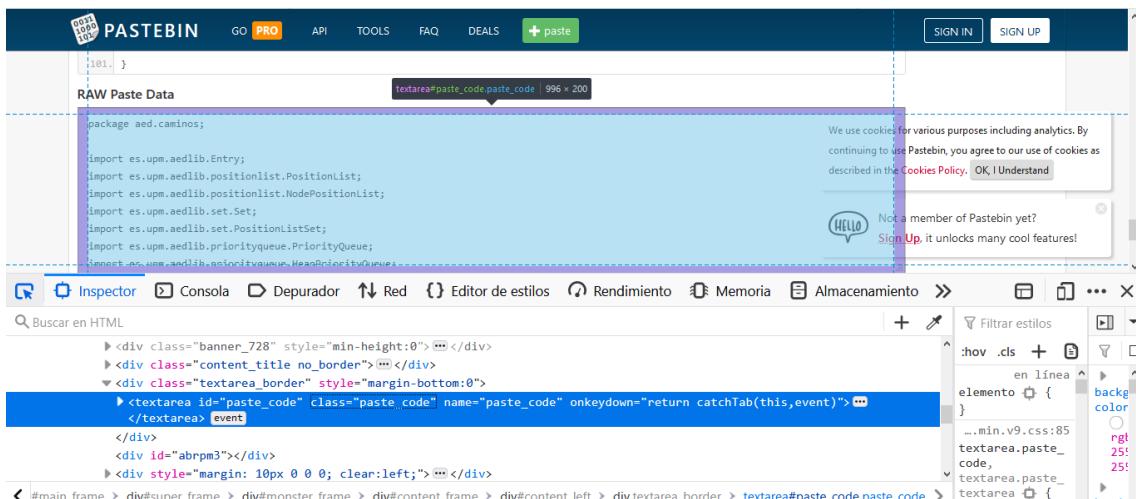


Ilustración 81 - Elemento HTML con el contenido raw

Este contenido se almacena en la variable “rawContent”, limitando el número de caracteres a 32767, que es el número máximo de caracteres que puede almacenar una celda de Excel.

```
rawContent = pasteParse.select('textareaid="paste_code"]')[32767]
```

Ilustración 82 - Noveno fragmento de código del método pastebin() de adquisición de Pastes

El último paso es insertar los datos de URL, título, fecha y contenido en la base de datos.

```
self.alert(pasteTitle + '\n' + url + '\n' + dateDMY)
self.insert_pastes(title=pasteTitle, url=url, date=dateDMY, content=rawContent)
```

Ilustración 83 - Décimo fragmento de código del método pastebin() de adquisición de Pastes

Una vez que se ha terminado de extraer la información deseada de Pastebin, se procede a procesar la información de Gist.

5.6.2 GitHub Gist

En este caso no será necesaria la utilización de Selenium ya que no hace falta que se renderice la página para poder extraer de la web todos los datos requeridos.

```
def gistGithub(self, brand, headers):
    url = 'https://gist.github.com/search?q=' + '"' + brand + '"'
```

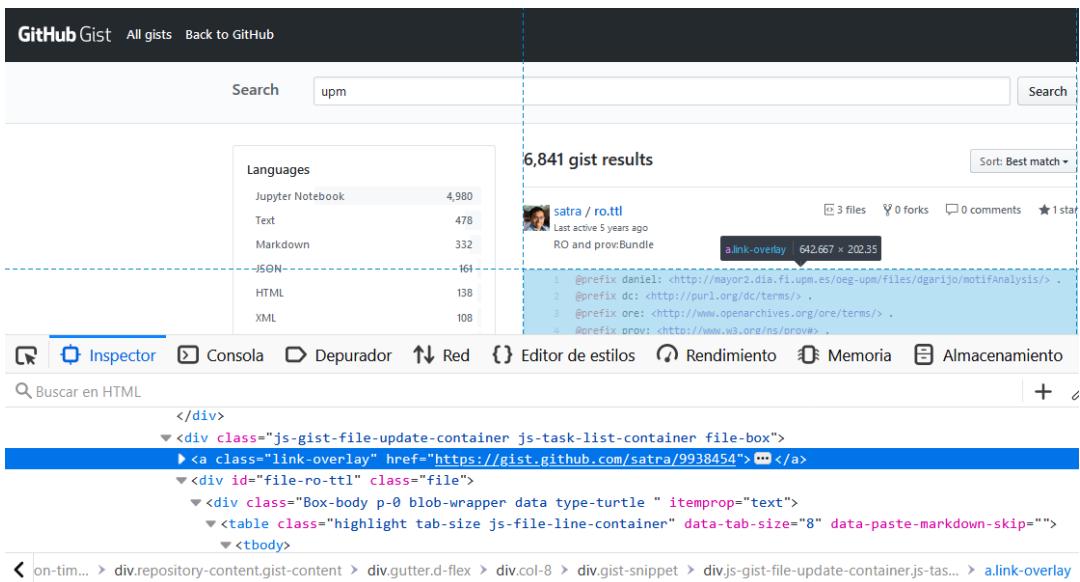
Ilustración 84 - Primer fragmento de código del método gistGithub() de adquisición de Pastes

En primer lugar, se hace una petición HTTP GET a la web de Gist con el módulo BeautifulSoup anteriormente explicado. La URL contiene el parámetro “/search?q=” como en la URL de Pastebin para que se haga una búsqueda. Añadiendo a continuación el nombre de marca, la búsqueda se hará sobre este. Al igual que en Pastebin, es importante que el nombre de marca vaya entre comillas.

```
while url:
    try:
        res = requests.get(url, headers=headers, timeout=5)
        res.raise_for_status()
        soup = bs4.BeautifulSoup(res.text, features='lxml')
        elemsGithub = soup.select('.link-overlay')
```

Ilustración 85 - Segundo fragmento de código del método gistGithub() de adquisición de Pastes

Una vez se tiene la web con los resultados de la búsqueda, se extraen las URLs de los resultados de los elementos HTML “.link-overlay”.

*Ilustración 86 - Elemento HTML que contiene la URL de los resultados*

Una vez obtenidos los links, se hacen peticiones HTTP GET a estos sitios para obtener los datos de título, fecha y contenido.

```
for x in range(len(elemsGithub)):
    try:
        pasteUrl = elemsGithub[x].get('href')
        res = requests.get(pasteUrl, headers=headers, timeout=5)
        res.raise_for_status()
        gistParse = bs4.BeautifulSoup(res.text, 'lxml')
        title = gistParse.find('title')
        pasteTitle = title.text
        date = gistParse.find('time-ago')
        dateParse = dateparser.parse(date.getText())
        dateDMY = dateParse.strftime("%d/%m/%Y")
```

Ilustración 87 - Tercer fragmento de código del método gistGithub() de adquisición de Pastes

Para obtener el contenido en raw de estos sitios, se aprovecha una característica de la web de Gist. El contenido en raw está en una URL igual a la del sitio, pero cambiando la cadena github por githubusercontent, por lo que haciendo uso de la función “replace” de Python se puede obtener el enlace del sitio que contiene el contenido en formato raw.

```
urlRaw = elemsGithub[x].get('href').replace('github', 'githubusercontent') + '/raw'
```

Ilustración 88 - Cuarto fragmento de código del método gistGithub() de adquisición de Pastes

Una vez obtenido el enlace al contenido raw se puede obtener con una petición HTTP GET, y con la función “text” de BeautifulSoup se extrae solo el texto contenido en el sitio.

```
res = requests.get(urlRaw, headers=headers, timeout=5)
res.raise_for_status()
rawContent = res.text[:32767]
```

Ilustración 89 - Quinto fragmento de código del método gistGithub() de adquisición de Pastes

Por último, se insertan en la base de datos los datos de fecha, URL, título, y contenido.

```
self.insert_pastes(title=pasteTitle, url=pasteUrl, date=dateDMY, content=rawContent)
```

Ilustración 90 - Sexto fragmento de código del método gistGithub() de adquisición de Pastes

En el caso de que haya más de una página de resultados se pasará de página y se repetirá el proceso de extracción de datos ya descrito. Para conseguir avanzar por todas las páginas se hace uso del elemento HTML `a` con la clase “`next_page`”.

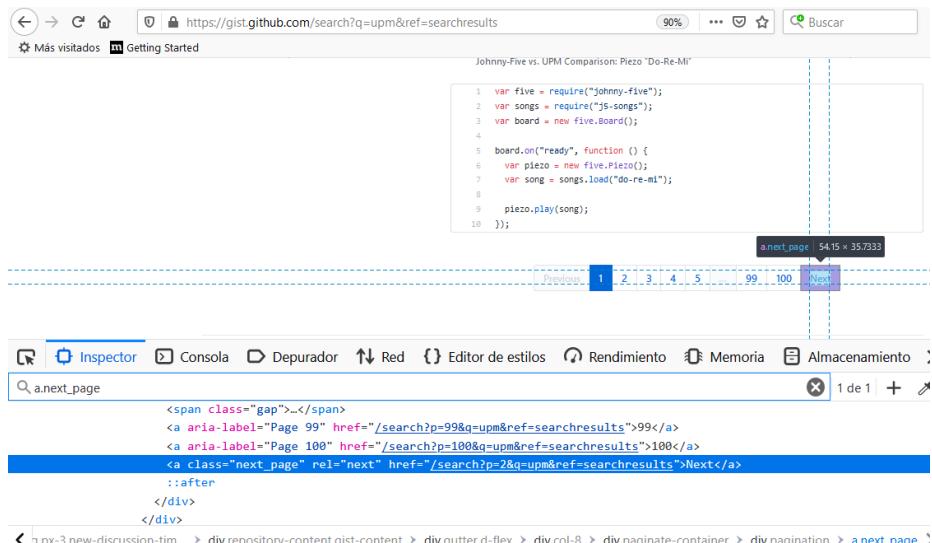


Ilustración 91 - Elemento HTML con la URL de la página siguiente

Este elemento contiene la URL a la página siguiente por lo que si existe se puede avanzar. Si no está ese elemento, quiere decir que no hay página siguiente y se sale del bucle “`while`” con un “`break`”.

```
url = soup.select('a.next_page')
if url:
    url = 'https://gist.github.com' + url[0].get('href')
else:
    self.verbose('No next page.')
    break
```

Ilustración 92 - Séptimo fragmento de código del método gistGithub() de adquisición de Pastes

5.7 Módulo de búsqueda de menciones en foros

Este módulo recopila menciones al nombre de marca en los foros ElOtroLado, Reddit y Forocoches.

```
meta = {
    'name': 'Posts',
    'author': 'Rodrigo Baladron',
    'version': '',
    'description': '',
    'dependencies': [],
    'files': ['geckodriver.exe'],
    'required_keys': [],
    'comments': (),
    'query': {'_source': ['name'], 'query': {'match': {'type': 'brands'}}},
    'options': (
        ('driver', os.path.join(BaseModule.data_path, 'geckodriver.exe'),
         True, 'path to selenium driver'),
        ('date', None, False, 'Last hour, day, week, month or year: h, d,
         w, m, y'),
    ),
}
```

Ilustración 93 - Información del módulo de adquisición de Posts

A nivel nacional, ElOtroLado y Forocoches son los foros de mayor actividad y repercusión, mientras que Reddit es el más importante a nivel internacional.

Como en otros módulos, se va a hacer una búsqueda en la web del término que interese para buscar artículos relacionados sobre él.

Hay foros que no permiten utilizar su buscador si no inicias sesión en su web. Este foro no da ese problema y permite usar su motor de búsqueda. Sin embargo, sí que restringe bastante el número de peticiones por minuto, por lo que hay que tener esto en cuenta para no ser bloqueados.

En estos foros también es interesante que exista la opción de filtrar resultados por fecha. Tanto Reddit como ElOtroLado poseen esta opción. En cuanto a Forocoches, como se accede mediante una búsqueda en Google, la opción de filtrar resultados por fecha también está disponible.

Al hacer las búsquedas usando diferentes buscadores, ya sea Google, o el del propio foro, el parámetro para filtrar por fecha (último mes, último día, último año, última semana) es diferente. Por ejemplo, en Reddit se usa la palabra "hour" para filtrar por día y en ElOtroLado se usa el número "1". Esto obliga a pasar diferentes variables por parámetro a cada método para que se pueda llevar a cabo esta opción de filtrado por fecha. Concretamente se usan dos variables diccionario de Python para hacer la conversión entre las palabras reservadas para el tipo de filtro de fecha de Google con las de los otros motores de búsqueda.

```
def module_run(self, brands):
    date = self.options['date']
    elOtroLadoDates = {'h': '', 'd': '1', 'w': '7', 'm': '30', 'y': '365'}
    redditDates = {'h': 'hour', 'd': 'day', 'w': 'week', 'm': 'month', 'y': 'year'}
    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/78.0.3904.108 Safari/537.36'}
    for brand in brands:
        if date is not None:
            self.elOtroLado(brand, elOtroLadoDates[date], headers)
            self.reddit(brand, redditDates[date], headers)
        else:
            self.elOtroLado(brand, None, headers)
            self.reddit(brand, None, headers)
            self.forocoches(brand, date, headers)
```

Ilustración 94 - Código del método module_run() de adquisición de Posts

Hay que añadir que este filtro de fecha no es obligatorio, es decir, si el usuario no da ningún valor a esta opción no se ponen estos filtros de fecha y se buscan resultados desde siempre. La repercusión de esta opción de fecha se verá más detalladamente en la explicación de los métodos de cada foro.

5.7.1 ElOtroLado

La URL para hacer la búsqueda es la siguiente:

```
def elOtroLado(self, brand, date, headers):
    count = 0
    url = 'https://www.elotrolado.net/search.php?sf=all&sr=posts&tips=1&keywords=' + brand + '''
```

Ilustración 95 - Primer fragmento de código del método elOtroLado() de adquisición de Posts

Los parámetros de la URL para realizar la búsqueda son los siguientes:

- “sr=posts” activa la opción para buscar en posts
- “keywords=” permite añadir un término de búsqueda.

Además, con las comillas se realiza una búsqueda exacta del nombre de marca.

```
if date:
    url += '&st=' + date
```

Ilustración 96 - Segundo fragmento de código del método elOtroLado() de adquisición de Posts

Si se ha recibido la fecha por parámetro se aplica este filtro, añadiendo el parámetro “&st=” a la URL. De la web de resultados se obtienen primero las URLs de dichos resultados, analizando la web con la librería BeautifulSoup de Python.

```
while url:
    try:
        response = requests.get(url, headers=headers, timeout=5)
        response.raise_for_status()
        soup = bs4.BeautifulSoup(response.text, features='lxml')
        elemsElOtroLado = soup.select('.title')
```

Ilustración 97 - Tercer fragmento de código del método elOtroLado() de adquisición de Posts

Las URLs se obtienen del elemento <a> con la clase title:

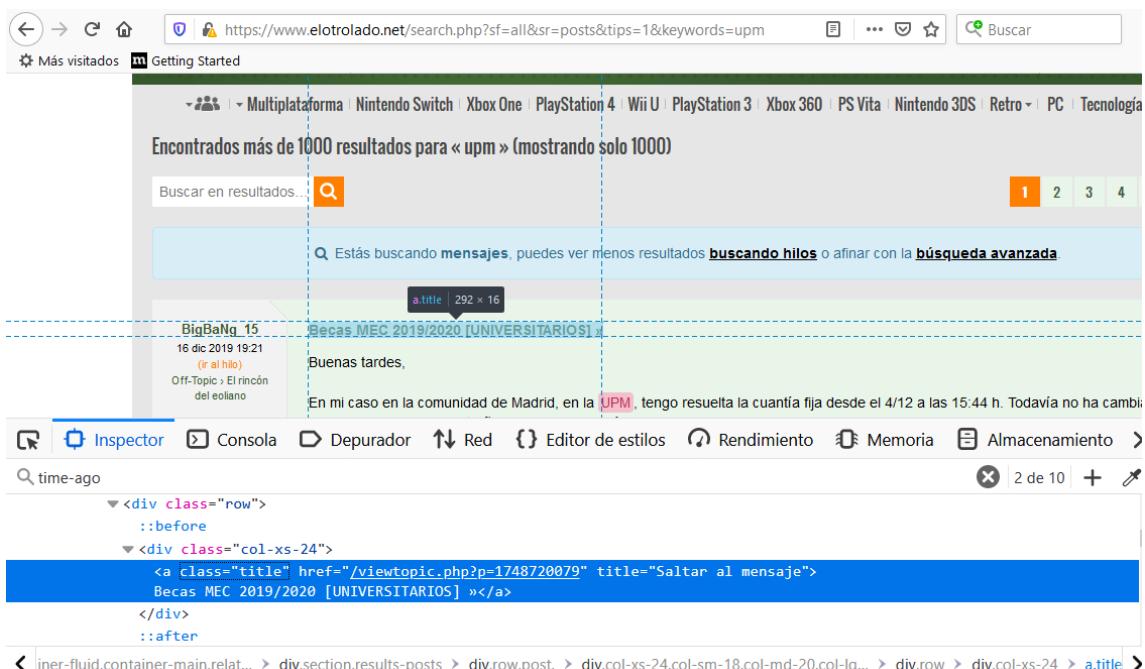


Ilustración 98 - Elemento HTML que contiene la URL del resultado

Los elementos con la clase “.title” encontrados se recorren y se extraen las URLs, ya que es un atributo del elemento, concretamente se obtiene el contenido del atributo “href” con el método “get()”. Posteriormente se hace una petición HTTP GET con el módulo “request” y se obtienen el título del post y la fecha del código HTML. El título se obtiene del elemento “title”, y la fecha se obtiene del atributo “title” del elemento time.

```
if elemsElOtroLado:
    for i in range(len(elemsElOtroLado)):
        try:
            postUrl = 'https://www.elotrolado.net' + elemsElOtroLado[i].get('href')
            response = requests.get(postUrl, headers=headers, timeout=5)
            response.raise_for_status()
            elOtroLadoParse = bs4.BeautifulSoup(response.text, 'lxml')
            title = elOtroLadoParse.find('title')
            postTitle = title.text
```

Ilustración 99 - Cuarto fragmento de código del método elOtroLado() de adquisición de Posts

Para obtener la fecha siempre con el mismo formato, se hace una transformación usando las librerías de Python “dateparser” y “datetime”.

```
dateElem = elOtroLadoParse.select('time')
dateParse = dateparser.parse(dateElem[0].get('title'))
dateDMY = dateParse.strftime("%d/%m/%Y")
```

Ilustración 100 - Quinto fragmento de código del metodo elOtroLado() de adquisición de Posts

Ilustración 101 - Elemento HTML con la fecha del resultado

El módulo “dateparser” analiza la información de la fecha y obtiene sus datos, soportando diferentes formatos. Es capaz de analizar fechas relativas como ‘hace un minuto’, ‘hace un año’, y también diferentes tipos de formatos genéricos como ‘August 14, 2015 EST’. Devuelve un objeto “datetime” con los datos analizados de la fecha como atributos. Estos datos son el día el mes, el año y la hora en caso de que vengan dados.

Finalmente, con el método “strftime()” del objeto “datetime” se puede pasar la fecha al formato deseado. En todos los módulos se usa el siguiente formato: 01/01/1991. Se usan los siguientes parámetros:

- “%d” muestra el día del mes
- “%m” muestra el número de mes
- “%Y” muestra el año completo

Se ha tenido en cuenta la paginación de los resultados, consiguiendo que el módulo recorra todas las páginas. En el módulo de Pastebin se hacía uso del botón siguiente de la web para poder obtener la URL de la página siguiente y se salía del bucle de búsqueda cuando no se encontrase ese botón. Sin embargo, aquí no se dispone de ese botón, por lo que se ha conseguido acceder a todas las páginas de otra manera.

La URL de esta web cambia en función de la página de resultados. A partir de la segunda página de resultados la URL cambia y se puede ver como se añade un parámetro más: “&start=50”.

Este parámetro indica el número de resultados de las páginas anteriores. Cada página muestra un máximo de 50 resultados, por lo tanto, según se avance de página se incrementará este número en 50. Por ello, el método para ir pasando de página es añadir una variable a la URL que va aumentando de 50 en 50.

```
count += 50
url = 'https://www.elotrolado.net/search.php?st=0&sk=t&sd=d&keywords='
+ brand + '&start=' + str(count)
if date:
    url += '&st=' + date
```

Ilustración 102 - Sexto fragmento de código del método elOtroLado() de adquisición de Posts

Si se añade este parámetro y la web no tiene más páginas, se muestra una página notificando que ha habido un error en la búsqueda. Esta página de error no contiene la clase “.title”, ya que no hay ningún título de resultados. Por lo tanto, este va a ser el método para saber si se ha llegado a esta página de error, comprobar si hay títulos.

Si al pasar de página no se encuentra el elemento con la clase “title”, esto indica que se ha sobrepasado la última página de resultados y estamos en la página de error. La página de error es la siguiente:



Ilustración 103 - Página de error

En este caso se sabe que no hay más páginas de resultados y se sale del bucle while.

```
else:
    print('no next page, loop ended')
    break
```

Ilustración 104 - Séptimo fragmento de código del método elOtroLado() de adquisición de Posts

5.7.2 Reddit

Reddit es un foro de temas diversos. Los usuarios pueden votar a favor o en contra del contenido, haciendo que aparezca más o menos destacado. Su público es mayoritariamente anglosajón y por tanto la mayoría del contenido está en inglés, pero también hay una actividad significativa de hispanohablantes.

```
def reddit(self, brand, date, headers):
    try:
        driverPath = self.options['driver']
        options = Options()
        options.add_argument('--headless')
        driver = webdriver.Firefox(executable_path=driverPath, options=options)
        url = 'https://www.reddit.com/search/?q=' + brand + ''
        if date:
            url += '&t=' + date
        driver.get(url)
```

Ilustración 105 - Primer fragmento de código del método reddit() de adquisición de Posts

BeautifulSoup permite analizar el código HTML de una web tras obtenerlo a partir de una petición HTTP GET. Sin embargo, en este caso pueden no estar todos los resultados de la búsqueda, ya que en esta web se van cargando los resultados al hacer scroll.

Por ello, es necesario usar el módulo Selenium de Python, ya que permite hacer scroll de la página de manera automática. Para que Selenium haga el scroll de la web, primero necesita medir la altura inicial del scroll.

```
lastHeight = driver.execute_script("return document.body.scrollHeight")
```

Ilustración 106 - Segundo fragmento de código del método reddit() de adquisición de Posts

Después se ejecuta la acción de hacer scroll del “driver” (elemento de Selenium que interactúa con la web). El scroll que se realiza mide lo mismo que la altura del documento.

```
while True:
    driver.execute_script("window.scrollTo(0, document.body.scrollHeight)
;")
```

Ilustración 107 - Tercer fragmento de código del método reddit() de adquisición de Posts

Mediante un sleep() se espera a que carguen los resultados de la web.

```
time.sleep(1.5)
```

Ilustración 108 - Cuarto fragmento de código del método reddit() de adquisición de Posts

Posteriormente se recalcula la altura actual de la web. Si la altura nueva es igual a la anterior significa que no se ha hecho scroll, por lo tanto, se ha llegado al final de la página y se sale del bucle. Si no es igual, no se sale del bucle y se asigna el contenido de la variable “newHeight” (altura nueva) a la variable “lastHeight” (altura anterior). De este modo al obtenerse la nueva altura en la siguiente iteración se podrá comparar con la altura de la iteración anterior.

```
newHeight = driver.execute_script("return document.body.scrollHeight")
if newHeight == lastHeight:
    break
lastHeight = newHeight
```

Ilustración 109 - Quinto fragmento de código del método reddit() de adquisición de Posts

Una vez se tiene cargada toda la web, se obtienen las URLs de acceso a los resultados de los elementos “SQnoC3ObvgnGjWt90zD9Z”. Este elemento se muestra en la siguiente captura:

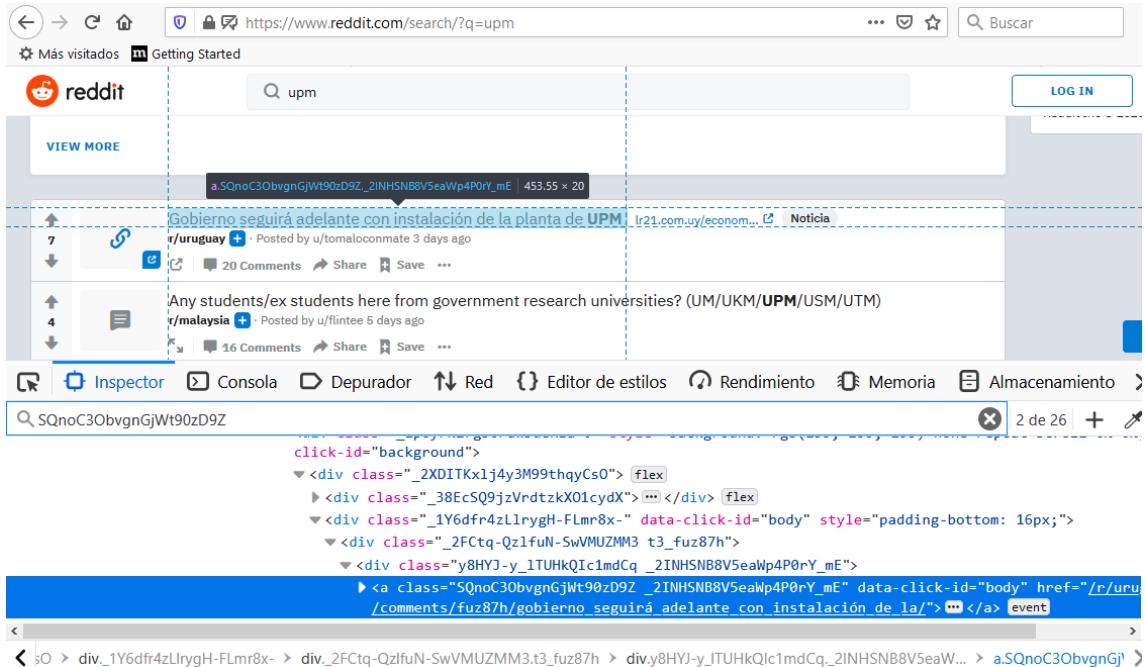


Ilustración 110 - Elemento HTML con la URL de los resultados

```
elems = driver.find_elements_by_class_name('SQnoC3ObvgnGjWt90zD9Z')
for i in range(len(elems)):
    postUrl = elems[i].get_attribute('href')
```

Ilustración 111 - Sexto fragmento de código del método reddit() de adquisición de Posts

Una vez se tienen las URLs, se obtiene la web a la que direccionan, esta vez con una petición HTTP GET.

```
response = requests.get(postUrl, headers=headers, timeout=5)
response.raise_for_status()
```

Ilustración 112 - Séptimo fragmento de código del método reddit() de adquisición de Posts

Como en los casos anteriores, se obtiene de esta web el título y la fecha de los posts usando BeautifulSoup, sabiendo previamente el nombre de los elementos que contienen estos datos. Finalmente, se insertan en Elasticsearch.

```

redditParse = bs4.BeautifulSoup(response.text, 'lxml')
title = redditParse.select('title')
postTitle = title[0].getText()
date = redditParse.select('._3j0xDPIQ0Ka0WpzvSQo-1s')
dateParse = dateparser.parse(date[0].getText())
dateDMY = dateParse.strftime("%d/%m/%Y")
self.alert(postTitle + '\n' + postUrl + '\n' + dateDMY)
self.insert_posts(title=postTitle, url=postUrl, date=dateDMY)

```

Ilustración 113 - Octavo fragmento de código del método reddit() de adquisición de Posts

5.7.3 Forocoches

Forocoches es un foro de Internet en español orientado inicialmente a la automoción, que permite la creación de hilos de discusión sobre prácticamente cualquier tema. Es uno de los 100 sitios web más visitados de España, por ello es una fuente interesante de información, sobre todo a nivel reputacional.

Se realiza una búsqueda utilizando “dorks” de Google. Los dorks de Google son palabras reservadas que permiten hacer búsquedas avanzadas. Con la palabra “site:” seguida de la dirección del sitio web, en este caso “forocoches.com”, Google muestra resultados únicamente de este sitio web.

```

def forocoches(self, brand, date, headers):
    url = 'https://www.google.es/search?q=site:forocoches.com ' + brand
    + '&filter=0'

```

Ilustración 114 - Primer fragmento de código del método forocoches() de adquisición de Posts

Se hace una petición HTTP GET de la URL mostrada en la Ilustración, la cual dirige a la página de Google con los resultados de la búsqueda en Forocoches.

```

while url:
    try:
        res = requests.get(url, headers=headers, timeout=5)
        res.raise_for_status()
        soup = bs4.BeautifulSoup(res.text, features='lxml')
        notFound = soup.select('.obp')
        if not notFound:
            elemsGoogle = soup.select('.r > a:first-of-type')

```

Ilustración 115 - Segundo fragmento de código del método forocoches() de adquisición de Posts

Utilizando BeautifulSoup se obtiene cada primer elemento `<a>` cuyo padre tenga asignada la clase “r”, y de este se extrae la URL, que contiene como atributo. Así se consiguen las URLs de los resultados mostrados.

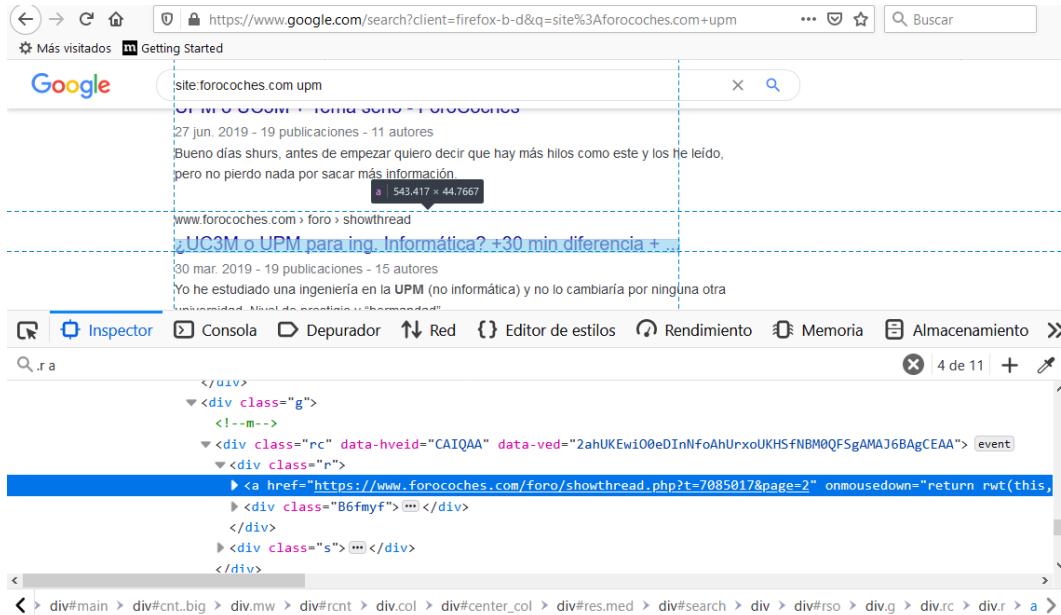


Ilustración 116 - Elemento HTML con la URL de los resultados

Posteriormente se extraen los títulos de los elementos `<h3>` cuya clase es “r”.

```
titlesGoogle = soup.select('.r h3')
```

Ilustración 117 - Tercer fragmento de código del método forocoches() de adquisición de Posts

El elemento se muestra a continuación:

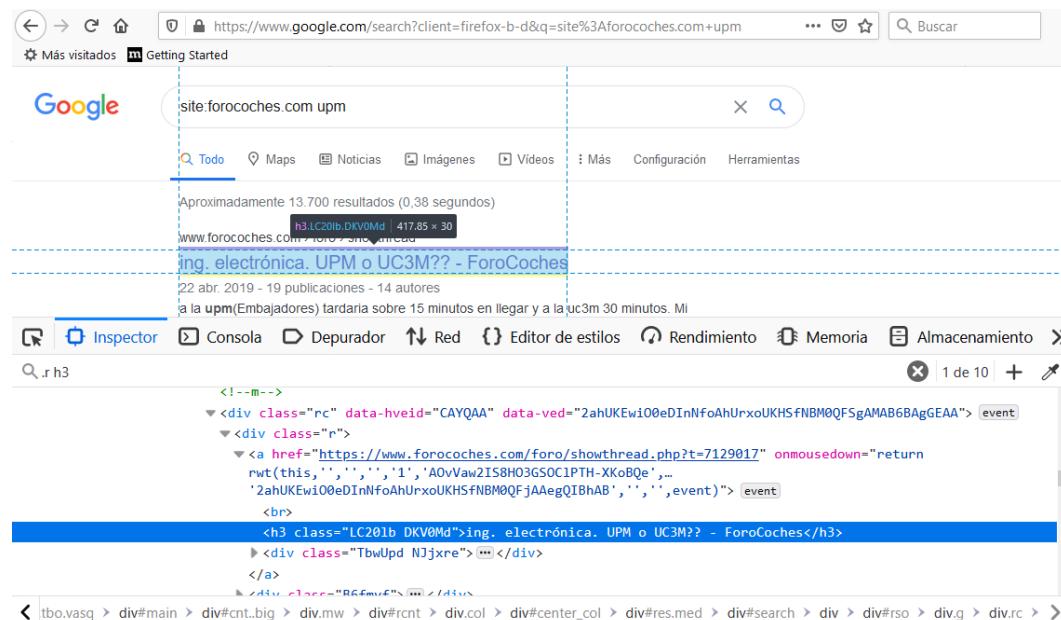


Ilustración 118 - Elemento HTML con el título del resultado

Las fechas se obtienen de los elementos con la clase “s”.

```
dateGoogle = soup.select('.s')
```

Ilustración 119 - Cuarto fragmento de código del método forocoches() de adquisición de Posts

El elemento se muestra a continuación:

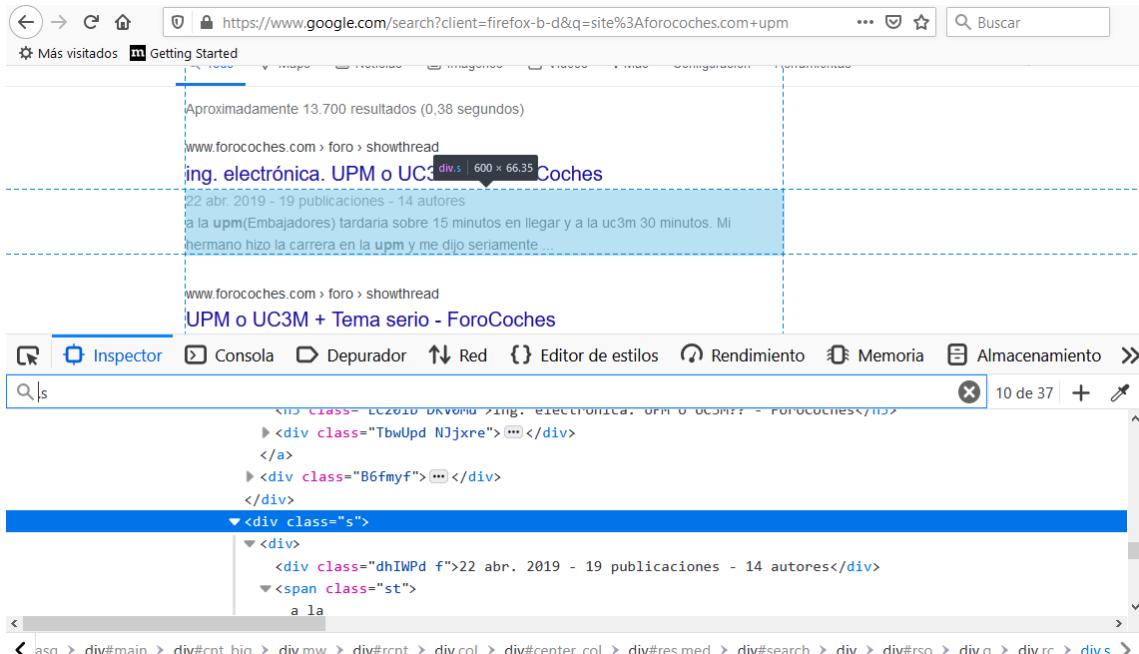


Ilustración 120 - Elemento HTML que contiene la fecha

Este elemento, además de la fecha, contiene la descripción. Por lo tanto, se ha tenido que extraer la fecha como se verá más adelante.

Una vez se han obtenido las listas de los elementos, se accede a cada uno para obtener su información. Sin embargo, en el caso del título, para verlo completo se necesita entrar en el resultado, ya que Google solo muestra una parte. Por ello, es necesario obtener la URL del resultado, que se encuentra en el atributo “href” nombrado anteriormente. Después, tras realizar una petición HTTP GET de dicha URL, se obtiene el título del elemento “title” con BeautifulSoup.

```
for i in range(len(elemsGoogle)):
    postUrl = elemsGoogle[i].get('href')
    res = requests.get(elemsGoogle[i].get('href'), headers=headers , timeout=5)
    try:
        res.raise_for_status()
        soupWeb = bs4.BeautifulSoup(res.text, features='lxml')
        title = soupWeb.find('title')
```

Ilustración 121 - Quinto fragmento de código del método forocoches() de adquisición de Posts

Si el título no se puede obtener de la web, se guarda el que se ha obtenido de los resultados de Google.

```
if title != None:
    if title.getText() != '':
        postTitle = title.getText()
    else:
        postTitle = titlesGoogle[i].getText()
else:
    postTitle = titlesGoogle[i].getText()
```

Ilustración 122 - Sexto fragmento de código del método forocoches() de adquisición de Posts

La fecha se obtiene de los elementos cuya clase es “.s” usando la siguiente expresión regular:

```
try:
    dateRegex = re.compile(r'^(\s|\.|\w)* - ')
    dateMatch = dateRegex.search(dateGoogle[i].getText())
```

Ilustración 123 - Séptimo fragmento de código del método forocoches() de adquisición de Posts

Esta expresión regular selecciona el texto a comienzo de línea que contenga letras, espacios o puntos antes de un guion. Como se aprecia en la siguiente imagen, esta expresión regular es adecuada para extraer la fecha.



Ilustración 124 - Elemento HTML que contiene la fecha

De manera similar que en los métodos correspondientes a los demás foros, se analiza la fecha con el módulo “dateparser”. A continuación, se da el formato de fecha deseado usando el objeto “datetime” que devuelve con la función “strftime”.

```

try:
    ...
    dateParse = dateparser.parse(str(dateMatch.group(0)))
    dateDMY = dateParse.strftime("%d/%m/%Y")
    self.alert(postTitle + '\n' + postUrl + '\n' + dateDMY)
    self.insert_posts(title=postTitle, url=postUrl, date=dateDMY)
except Exception as e:
    pass

```

Ilustración 125 - Octavo fragmento de código del método forocoches() de adquisición de Posts

Para obtener el enlace para pasar de página, se busca el elemento cuyo id es “pnnext”, del cual se obtiene la URL que permite ir a la página siguiente.

```

url = soup.find('a', id='pnnext')
if url:
    url = 'https://www.google.es/' + url['href']
else:
    self.verbose('No next page.')
    break

```

Ilustración 126 - Noveno fragmento de código del método forocoches() de adquisición de Posts

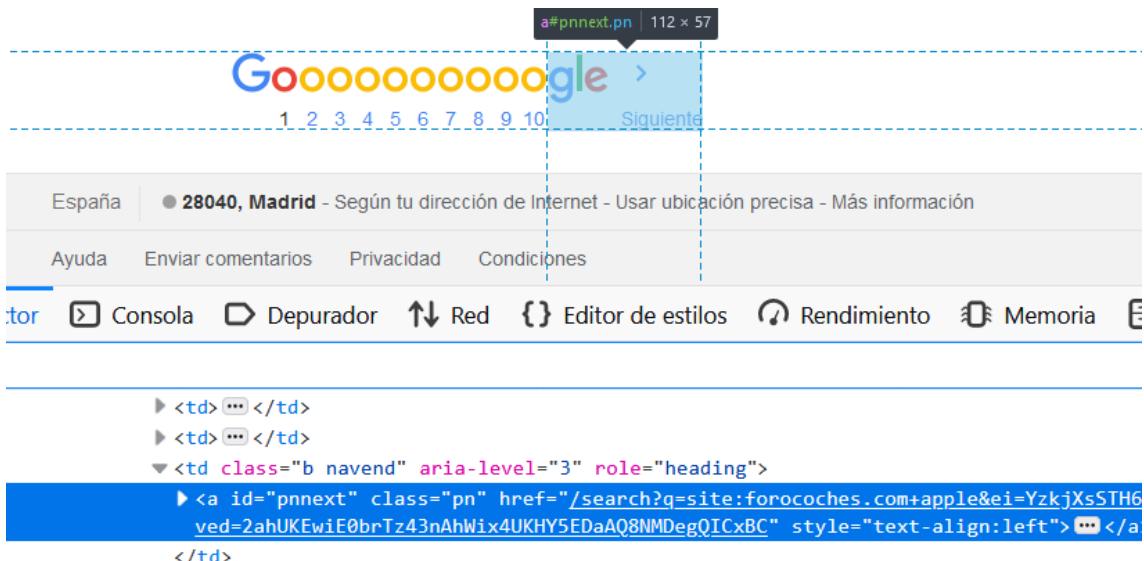


Ilustración 127 - Elemento HTML con el enlace a la página siguiente de Google

Si no se encuentra este elemento quiere decir que no hay página siguiente, por lo que se sale del bucle y se finaliza la ejecución.

5.8 Módulo de comprobación de dominios en listas negras

Este módulo comprueba si alguna de las IPs de los registros MX de los dominios dados aparecen en alguna de las listas negras de spam seleccionadas.

```
meta = {
    'name': 'Blacklist Check',
    'author': 'Rodrigo Baladrón',
    'version': '',
    'description': '',
    'dependencies': [],
    'files': [],
    'required_keys': [],
    'comments': (),
    'query': {'_source': ['domain'], 'query': {'match': {'type': 'domains'}}},
    'options': (),
}
```

Ilustración 128 - Información del módulo de comprobación de dominios en listas negras

En la lista “blacklists” se encuentran 77 páginas con listas negras de spam.

```
def module_run(self, domains):
    blacklists = ['bl.score.senderscore.com', 'bl.mailspike.net', ...]
    resolver = self.get_resolver()
```

Ilustración 129 - Primer fragmento de código del módulo de Listas negras de spam

Se recorre la lista de diccionarios con los dominios que se van a comprobar. Por cada dominio, con el método “dns.resolver.query()” del módulo dnspython se obtiene un objeto de tipo <dns.resolver.Answer>, que contiene información sobre los registros MX, y se almacena en la variable answer.

```
for domain in domains:
    try:
        answers = resolver.query(domain, 'MX')
```

Ilustración 130 - Segundo fragmento de código del módulo de Listas negras de spam

Dicho objeto se recorre obteniendo los servidores de correo de los registros MX mediante el método “exchange()”, y se elimina el último punto con el método strip().

```
for rdata in answers:
    mailServer = str(rdata.exchange).strip('.'
```

Ilustración 131 - Tercer fragmento de código del módulo de Listas negras de spam

De cada servidor de correo se obtiene una tupla con información sobre estos mediante el método “gethostbyname_ex()” del módulo “socket”. En la tercera posición de dicha tupla se encuentra una lista con las IPs de los servidores.

```
try:
    serverInfo = socket.gethostbyname_ex(mailServer)
    IPs = serverInfo[2]
```

Ilustración 132 - Cuarto fragmento de código del módulo de Listas negras de spam

Una vez se tienen las IPs, se procede a comprobar si están listadas en alguna de las listas negras de spam. Para ello se hace una consulta a los servidores de listas negras con el siguiente formato: <IP al revés>.<dominio de lista negra>

Para ello hay que dar la vuelta a la IP y concatenarla con los dominios de listas negras. Primero se divide la IP almacenándola en la lista “reverseIPList”, cuyos elementos son los números de la IP separados por puntos con el método “split()”, invirtiendo su orden mediante el método “reversed()”. Posteriormente se crea el string “reverseIP” correspondiente a la IP invertida, concatenando los números de la lista separados por puntos, y en último lugar en la variable “checkHost” se concatena la blacklist a la IP invertida, separadas por un punto.

```
for blacklist in blacklists:
    self.verbose(blacklist)
    for IP in IPs:
        reverseIPList = list(reversed(IP.split('.')))
        reverseIP = ''
        for i in range(len(reverseIPList)-1):
            reverseIP += str(reverseIPList[i]) + '.'
        reverseIP += reverseIPList[len(reverseIPList)-1]
        checkHost = reverseIP + '.' + blacklist
```

Ilustración 133 - Quinto fragmento de código del módulo de Listas negras de spam

Con el método “socket.gethostbyname()” se realiza la consulta para obtener la IP del host, pasándole como parámetro la variable “checkHost”.

```
try:
    socket.gethostbyname(checkHost)
    self.alert(domain + '\n' + mailServer + '\n' + IP + '\n' + 'Blacklist
: ' + blacklist)
    self.insert_spamMailServers(domain=domain, mail_server=mailServer, ip
=IP, blacklist=blacklist)
except socket.gaierror:
    pass
```

Ilustración 134 - Sexto fragmento de código del módulo de Listas negras de spam

Si alguna IP de los servidores de correo del dominio se encuentra en alguna lista negra, se insertará en Elasticsearch un documento de tipo “spamMailServers” con el dominio, el servidor de correo, la IP y la blacklist. Si salta la excepción provocada por el error “getaddrinfo failed” significa que la IP del servidor de correo no se encuentra en la blacklist, por lo que no se insertarán dichos datos.

5.9 Módulo de exportación de los datos adquiridos a Excel

Este módulo exporta los datos almacenados en la base de datos de Elasticsearch, previamente adquiridos con el resto de módulos desarrollados, siguiendo el formato corporativo de los informes de Vigilancia Digital.

Cada sección del informe va en una hoja distinta. Las secciones automatizadas por este módulo son las siguientes: Reputación, Emails localizados, Fuente de localización del email, Documentos, Metadatos, Autores de documentos, Pastes, Listas negras, Foros y Posible phishing.

```
meta = {
    'name': 'excelVD',
    'author': 'Rodrigo Baladrón, Rubén Álvarez',
    'version': '',
    'description': '',
    'options': (
        ('filename', os.path.join(BaseModule.workspace, os.path.basename(
            BaseModule.workspace) + ' - ' + datetime.datetime.today().strftime(
            '%d-%m-%Y') + '.xlsx'), True, 'path and filename for output'),
        ('date', None, False, 'Last X hours, days, weeks, months or years
        : 1h, 2d, 3w, 4M, 5y')
    ),
}
```

Ilustración 135 - Información del módulo de exportación a Excel

Algunas de estas secciones tienen modos de representación de datos comunes, por lo que el método de extracción es muy similar:

- Reputación, pastes y foros → Se representan los datos en una única tabla con un encabezado.
- Emails, documentos, dominios similares → Este método crea una tabla por cada dominio, escribiéndose en ellas los datos correspondientes a ese dominio. Las tablas se disponen unas al lado de las otras.
- Fuente de localización de emails, listas negras → Además del email, se representa la fuente de la que se ha obtenido dicho email. Si un email tiene varias fuentes de obtención, el email aparece en una celda combinada y las fuentes aparecen en las celdas contiguas a esta celda combinada.
- Metadatos → Se escribe la URL del documento, debajo los metadatos y debajo de estos la URL del siguiente documento con sus metadatos.

Estos métodos se describirán más abajo, con ejemplos para entender mejor su funcionamiento.

5.9.1 Extracción de los datos indexados en Elasticsearch

Para poder mostrar los datos primero se deben extraer de la base de datos. Se realiza de la misma manera para todos los datos, sin embargo, por su disposición hay algunos que tienen consultas diferentes. A continuación, se van a mostrar algunos de los casos:

El primer caso es el básico, la extracción de noticias para la sección reputación.

```
wb = openpyxl.Workbook()
```

Ilustración 136 - Primer fragmento de código del método module_run() de exportación a Excel

Se crea un documento Excel sobre el que trabajar con el método “openpyxl.Workbook()”. Por defecto se crea una hoja al crear el documento, por lo que se le cambia el título a la hoja por el que corresponda en lugar de crear una hoja nueva.

```
sheet = wb.active
sheet.title = 'Reputación'
```

Ilustración 137 - Segundo fragmento de código del método module_run() de exportación a Excel

Se crea la consulta a la base de datos para obtener los campos de título, URL y fecha de las noticias. Si se activa la opción “date”, se hará la consulta a Elasticsearch obteniendo los datos indexados en el rango de tiempo indicado por el usuario.

```
if date is None:
    query = {'_source': ['title', 'url', 'date'], 'query': {'match': {
        'type': 'news'}}}
else:
    query = {'_source': ['title', 'url', 'date'], 'query': {'bool': {
        'must': {'match': {'type': 'news'}}, 'filter': {'range': {'timestamp':
            : {'gte': 'now-' + date}}}}}}
```

Ilustración 138 - Tercer fragmento de código del método module_run() de exportación a Excel

Se recorren los documentos obtenidos de la consulta. Estos documentos contienen el diccionario “_source” con los datos indexados. Por cada documento obtenido se introduce su diccionario “_source” en la lista “sources”, inicialmente vacía.

Esta lista de diccionarios se le pasa al método básico para exportar en Excel “dictsToTable()”. Cada uno de estos diccionarios tiene los datos de título de noticia, URL y fecha.

```
docs = self.read_doc_ES(os.path.basename(self.workspace), query)
sources = []
for doc in docs:
    sources.append(doc['_source'])
if sources:
    self.dictsToTable(sheet, sources)
```

Ilustración 139 - Cuarto fragmento de código del método module_run() de exportación a Excel

Hay otros dos casos de extracción en los que se hacen algunas comprobaciones más antes de usar los datos.

En el caso de los metadatos la extracción es diferente, ya que se tiene que comprobar que el documento no tenga el campo “metadata” vacío; si dicho campo está vacío ese documento no se incluirá en la pestaña metadatos. Una vez que se tienen todos los datos en la lista de diccionarios “sources”, esta se pasa como parámetro al método “metadataToTable()” junto con la hoja de Excel en la que se incluirán esos datos.

```
if date is None:
    query = {'_source': ['url', 'metadata'], 'query': {'match': {'type': 'documents'}}}
else:
    query = {'_source': ['url', 'metadata'], 'query': {'bool': {'must': {'match': {'type': 'documents'}}, 'filter': {'range': {'timestamp': {'gte': 'now-' + date}}}}}}
docs = self.read_doc_ES(os.path.basename(self.workspace), query)
sources = []
for doc in docs:
    source = doc['_source']
    if source['metadata']:
        sources.append(source)
if sources:
    self.metadataToTable(sheet, sources)
```

Ilustración 140 - Quinto fragmento de código del método module_run() de exportación a Excel

Para la pestaña de los autores hay que hacer alguna comprobación más que en el caso de los metadatos, teniendo en cuenta que los autores se encuentran en los diccionarios de metadatos.

Si hay metadatos y en el diccionario metadatos se encuentra el campo “Author”, entonces se guarda esa pareja clave-valor en la variable “author”. Posteriormente se comprueba si ese autor ha sido introducido con anterioridad en la lista de diccionarios “authors”, y si no está se incluye. De este modo no habrá autores duplicados.

```
authors = []
for doc in docs:
    source = doc['_source']
    if source['metadata']:
        if 'Author' in source['metadata'].keys():
            author = source['metadata']['Author']
            if not any(authorDict['Author'] == author for authorDict
                       in authors):
                authors.append({'Author': author})
if authors:
    self.dictsToTable(sheet, authors)
```

Ilustración 141 - Sexto fragmento de código del método module_run() de exportación a Excel

Si la lista de diccionarios con los autores no está vacía, se pasa por parámetro al método “dictsToTable()”, junto con la hoja de Excel en la que se escribirán los datos.

5.9.2 Método básico de creación de tablas (reputación, pastes y foros)

Este método recibe por parámetro una lista de diccionarios que contienen los datos a exportar (con las mismas claves) y el objeto de la hoja de Excel a modificar.

Con el método “keys()” se obtiene una lista con todas las claves del diccionario. Como todos los diccionarios tienen las mismas claves, con recorrer el primer diccionario de la lista es suficiente. Antes de crear ninguna tabla con datos se tiene que definir a partir de qué celda empezará la tabla, en este caso se ha optado por la celda B2 (columna B, fila 2).

```
def dictsToTable(self, sheet, dictList):
    columnLenDictList = []
    keys = dictList[0].keys()
    headerRow = 2
    columnNum = 2
    for key in keys:
        sheet.cell(row=headerRow, column=columnNum).value = key
        sheet.cell(row=headerRow, column=columnNum).alignment = Alignment
            (horizontal='center', vertical='center')
```

Ilustración 142 - Primer fragmento de código del método dictsToTable() de exportación a Excel

Una vez definida la celda inicial se comienza a escribir el contenido de la tabla, empezando por llenar lo que luego será el encabezado. El encabezado se escribe a partir del nombre de las claves del diccionario, asignando los valores a las celdas con el método “sheet.cell().value”. Para que el contenido de las celdas esté centrado se usa la función “alignment” del módulo “openpyxl”.

Las columnas de Excel tienen un tamaño determinado por defecto. Sin embargo, este tamaño no es adecuado y es conveniente ajustarlo para cada columna en función de la celda de la columna que más contenido tenga y por tanto más anchura necesite. La anchura de las columnas en Excel utiliza una unidad de medida especial. Se mide en número de caracteres de la fuente Arial a tamaño 10 que caben en la celda; es decir, en una celda de tamaño 10 caben aproximadamente 10 caracteres.

Para ajustar en las tablas el ancho de la columna primero es necesario obtener el número máximo de caracteres escritos por celda de la columna, para así ajustar el ancho de la columna al de la celda con más caracteres. Este valor no se puede obtener hasta que estén todas las tablas escritas, por lo que se hace uso de un diccionario para ir guardando ese valor máximo.

```
for key in keys:  
    ...  
    columnLenDictList.append({'column': columnNum, 'maxWidth': len(key)})  
    columnNum += 1
```

Ilustración 143 - Segundo fragmento de código del método dictsToTable() de exportación a Excel

Después se pasa a la siguiente fila para escribir los valores de los campos de los diccionarios. Para obtener dichos valores se recorre la lista de diccionarios “dictList” y mediante el método “values()” se obtiene una lista con todos los valores de cada uno de ellos. Estos valores se escriben en una fila, así habrá una fila por cada diccionario.

```
rowNum = 3  
for d in dictList:  
    values = d.values()  
    columnNum = 2  
    for value in values:  
        sheet.cell(row=rowNum, column=columnNum).value = value  
        sheet.cell(row=rowNum, column=columnNum).alignment = Alignment(  
            vertical='center', wrap_text=True)
```

Ilustración 144 - Tercer fragmento de código del método dictsToTable() de exportación a Excel

Por cada celda que se escribe se actualiza el valor “maxWidth”, correspondiente al ancho de la columna, si el contenido de la celda supera en longitud de caracteres al máximo registrado anteriormente para esa columna. Esto se consigue comparando la longitud de la variable “value” con el valor de la máxima longitud hasta el momento dentro de esa misma columna.

```
for value in values:
    ...
    for columnLenDict in columnLenDictList:
        if columnLenDict['column'] == columnNum:
            valueLen = len(str(value))
            if valueLen > columnLenDict['maxWidth']:
                columnLenDict['maxWidth'] = valueLen
            break
    columnNum += 1
rowNum += 1
```

Ilustración 145 - Cuarto fragmento de código del método dictsToTable() de exportación a Excel

Al finalizar este bucle interno se incrementa la variable “columNum” para avanzar en la fila, y tras el bucle anterior se incrementa el contador “rowNum” para escribir en la siguiente fila.

Después de los bucles “for” utilizados para escribir el contenido de las tablas, se define el ajuste del ancho de las columnas. Dentro de estos bucles, mientras se escribía en celdas, se fueron guardando los tamaños máximos de columna en la lista de diccionarios “columnLenDictList”. Recorriendo esta lista de diccionarios y accediendo a los valores “column” y “maxWidth” de cada diccionario, se ajusta esa columna con el valor “maxWidth”, asignándoselo al atributo “width” de la columna.

```
for columnLenDict in columnLenDictList:
    columnLetter = openpyxl.utils.cell.get_column_letter(columnLenDict[
        'column'])
    sheet.column_dimensions[columnLetter].width = columnLenDict['maxWidth']
    '] + 2
```

Ilustración 146 - Quinto fragmento de código del método dictsToTable() de exportación a Excel

Una vez se han escrito todos los datos que compondrán la tabla y se han ajustado los anchos de las columnas, se coge la referencia de la primera y última celda de la tabla para poder dar formato de tabla a los datos escritos.

```

lastColumn = openpyxl.utils.cell.get_column_letter(len(keys) + 1)
lastRow = str(headerRow + len(dictList))
ref = 'B2:' + lastColumn + lastRow
tableName = sheet.title.replace(' ', '')
tab = Table(displayName=tableName, ref=ref)

```

Ilustración 147 - Sexto fragmento de código del método dictsToTable() de exportación a Excel

Después se le da un nombre a la tabla, que será el título de la hoja sin espacios, y con la referencia y el nombre se crea el objeto “tab” de la clase “Table”.

Por último, para darle un estilo por defecto a la tabla se especifica modificando el atributo “tableStyleInfo” del objeto “tab”, y se añade a la hoja.

```

style = TableStyleInfo(name='TableStyleMedium9', showFirstColumn=False, s
howLastColumn=False, showRowStripes=True, showColumnStripes=False)
tab.tableStyleInfo = style
sheet.add_table(tab)

```

Ilustración 148 - Octavo fragmento de código del método dictsToTable() de exportación a Excel

El resultado es el siguiente:

A	B	C	D	E
1	date	title	url	
2	06/03/2020	COMUNICADO: Expominerales Madrid 2020, 40 años dando a conocer las Ciencias de la Tierra a la Sociedad	https://www.lavanguardia.com/vida/20200306/47398232089/comunicado-	
3	23/03/2020	La Universidad de Barcelona, la Politécnica y la Complutense, las mejores de España por las materias que	https://www.leonoticias.com/sociedad/educacion/universidad-barcelona-	
4	23/03/2020	Las universidades madrileñas donan miles de guantes, mascarillas y respiradores - Madrid es Noticia	https://www.madridesnoticia.es/2020/03/as-universidades-madrileñas-	
5	25/03/2020	ENAIRE gana el I Premio de Sostenibilidad en los Maverick Awards 2020 de ATM - Hispaviación	http://www.hispaviacion.es/enaire-gana-el-i-premio-de-sostenibilidad-en-los-	
6	04/03/2020	Poder, glamour y bodegas: la familia Cortina-Lapique, una saga unida a la discreción	https://www.revistavintyfair.es/sociedad/articulos/alfonso-cortina-mujer-	
7	03/03/2020	Una aplicación permite a invidentes familiarizarse con sitios desconocidos	https://www.eldiario.es/tecnologia/aplicacion-permite-invidentes-	
8	04/03/2020	Fenin se une a la Universidad Politécnica de Madrid Noticias Dentales	https://dentalstaentuciudad.com/noticias-dentales/fenin-se-une-a-la-	
9	24/03/2020	Los siete oros de los Premios Emporia a la arquitectura efímera	https://www.eventoplus.com/noticias/los-siete-etros-de-los-premios-emporia-	
10	26/03/2020	¿Cómo ser ingeniero de tráfico? requisitos, sueldo y cursos Cursos.com	https://cursos.com/ingeniero-trafico/	
11	25/03/2020	“Decidi volver cuando anunciaron que el confinamiento sería total” NuevaAlcarria - Guadalajara	https://nuevaalcarria.com/articulos/decidi-volver-cuando-anunciaron-que-el-	
12	08/03/2020	Estas son las 100 mejores universidades del mundo (y hay dos españolas)	https://www.elconfidencial.com/almacorazonvida/educacion/2020-03-	
13	09/03/2020	Miguel Becer, el estilista viral de Rosalia, Kylie Jenner, Kim Kardashian y Dua Lipa Personajes	https://www.expandia.com/fueradeserie/personajes/2020/03/24/se77d7a346	
14	24/03/2020	¿Qué necesito estudiar para ser Ingeniero Industrial?	https://www.mastermania.com/noticias_masters/que-necesito-estudiar-para-	
15	20/03/2020	La UME desplegará en Ifema un hospital de 5.500 camas	https://www.20minutos.es/noticia/4194815/0/la-ume-desplegar%C3%A1-en-ifema-un-	
16	22/03/2020	La UME desplegará en Ifema un hospital de 5.500 camas	https://www.20minutos.es/noticia/4194815/0/la-ume-desplegar%C3%A1-en-ifema-un-	
17	19/03/2020	Foros de empleo 2020 en las universidades	https://www.mastermania.com/noticias_masters/foros-de-empleo-2020-en-	
18	03/03/2020	Todas las actividades de AUA 2020 - Madrid es Noticia	https://www.madridesnoticia.es/2020/03/todas-las-actividades-de-aula-2020/	
19	23/03/2020	Ejercicios con cartones de leche para mayores diabéticos	https://www.larazon.es/salud/20200323/fmyldz7yrdjoleqxfznwvnm.html	
20	24/03/2020	HISTORICAL AND METRIC REFERENCE OF THE MURCIAN COTAGE: ESTIMATION OF ITS CONSTRUCTION COSTS .	https://blogs.laopiniondemurcia.es/pedro-pina/2020/03/25/historical-and-	
21	24/03/2020	Protección Civil Móstoles dona dos respiradores para afrontar el coronavirus en el Hospital Universitario	https://www.soy-d.com/noticia-mostoles/proteccion-civil-mostoles-dona-dos-	
22	12/03/2020	450 castellano-manchegos se organizan para fabricar equipos de protección con sus impresoras 3D -	https://www.encastillalamancha.es/castilla-la-mancha/450-castellano-	
23	23/03/2020	Científicos analizan cómo cambia el cerebro humano a través de la formación musical	https://www.lavanguardia.com/local/sevilla/20200309/47404686045/cientifico	
24	04/03/2020	La Universidad Autónoma y la Carlos III alargan los exámenes hasta julio	https://www.lavanguardia.com/deportes/20200312/474101159203/la	
25	01/03/2020	Un sismo de magnitud 6 en la falla “acabarla con todo” - La Opinión de Murcia	https://www.laopiniondemurcia.es/comunidad/2020/03/02/seisimo-magnitud-	
26	12/03/2020	Hilamos la sede de gusano para cultivar neuronas - NIUS	https://www.niusdiario.es/ciencia-y-tecnologia/ciencia/hilamos-seda-gusano-	
27	23/03/2020	Una niña de tres años da positivo en coronavirus en Torrejón de Ardoz Madridario	https://www.madridario.es/nina-tres-anos-positivo-coronavirus-torrejon	

Ilustración 149 - Hoja de Reputación

A	B	C	D	E	F
1	date	title	url	content	
	06/03/2018	oiplayapi.json - GitHub	https://gist.github.com/jeffkenichi/1fe8576dfb472f78ec06a6e23d4b681	<pre> echo script requires that packages 'curl' and 'jq' are installed echo .upm_deployAddon.sh echo upm_deploy http://localhost:8080/jira admin:passwd path_toAddon.jar upm_deploy() { TOKEN=\$(curl -s -I -u \$2 \$1/rest/plugins/1.0/grep ^upm- token cut -c12-[tr -d "[space:]") out=\$(curl -s -X POST -u \$2 -H "Accept: application/vnd.atl.plugins.installed+json" -F plugin=@\$3 \$1/rest/plugins/1.0/?token=\$TOKEN) while [\$(echo \$out jq -r .status.done)" = "false"]; do sleep 1; echo -n . pending=\$(echo \$out jq -r .links.self) out=\$(curl -s -u \$2 \$1%\$1/\$pending) done echo "\$?" } # coding: utf-8 # In[]: </pre>	
3					

Ilustración 150 - Hoja de Pastes

Este método de exportar es el básico, a partir de este método se han realizado otros métodos que requieren de una lógica más compleja.

5.9.3 Método de creación de tablas agrupadas por dominios (emails, documentos y dominios similares)

Este método crea una tabla por cada dominio, escribiéndose en ellas los datos correspondientes a ese dominio. Estas tablas se disponen unas al lado de las otras. En este primer ejemplo, se va a comenzar explicando el método de representación de los emails.

Se comienza recorriendo la lista de diccionarios “emailDictList”, que contiene diccionarios con el campo “email”, correspondiente a la dirección de correo electrónico. De esta entrada se extrae el dominio del email con una expresión regular “(?<=@).+”, con la cual se buscan los caracteres posteriores al arroba.

```

def emailsToTables(self, sheet, emailDictList):
    maxColumn = 0
    tableDictList = []
    columnLenDictList = []
    for emailDict in emailDictList:
        email = emailDict['email']
        domainRegex = re.compile('(?<=@).+')
        domain = domainRegex.search(email).group()

```

Ilustración 151 - Primer fragmento de código del método emailsToTables() de exportación a Excel

Una vez se ha extraído el dominio, se comprueba si ha sido insertado en algún diccionario de la lista de diccionarios “tableDictList”. Si no se ha insertado, significa que es la primera vez que aparece un email con este dominio. Por el modo en el que se van a mostrar los datos en Excel, con una tabla por cada dominio que contenga los emails de ese dominio, si es la primera vez que aparece un dominio se tiene que escribir el email con ese dominio en una tabla nueva.

```
headerRow = 2
if not any(tableDict.get('domain', None) == domain for tableDict in tableDictList):
    columnNum = maxColumn + 2
```

Ilustración 152 - Segundo fragmento de código del método emailsToTables() de exportación a Excel

Lo primero que se debe determinar para escribir una tabla nueva es en qué celda empezará, igual que pasaba en el método básico. Para este caso la fila se sabe, ya que como las tablas van a ir dispuestas unas al lado de las otras, la primera fila de cada una de ellas siempre será la misma, en este caso se ha optado por escribir todas las tablas a partir de la fila 2. Sin embargo, la columna no va a ser constante, ya que se va a escribir una tabla nueva por cada dominio. Como las tablas se van a ir creando de izquierda a derecha, hay que saber en qué columna termina la última tabla para escribir la nueva a partir de esa.

Para saber esta última columna escrita, basta con almacenar en una variable global este valor “maxColumn”. Por cada tabla que se escribe nueva este valor se actualiza, de modo que cuando se va a escribir una tabla nueva para saber por qué columna empezar, basta con sumarle dos a este valor, uno por el hueco y otro por la nueva columna en la que va a empezar la siguiente tabla.

Una vez que se sabe por dónde empezar a escribir la siguiente tabla, el proceso de creación es similar que en el método básico. Se comienza por el encabezado, se añaden las filas, y se termina dando formato a la tabla.

```
sheet.cell(row=headerRow, column=columnNum).value = 'Emails @' + domain
sheet.cell(row=headerRow, column=columnNum).alignment = Alignment(horizontal='center', vertical='center')
columnNum = maxColumn + 2
rowNum = headerRow + 1
value = emailDict['email']
sheet.cell(row=rowNum, column=columnNum).value = value
sheet.cell(row=rowNum, column=columnNum).alignment = Alignment(vertical='center', wrap_text=True)
```

Ilustración 153 - Tercer fragmento de código del método emailsToTables() de exportación a Excel

En este caso el encabezado no está formado por el nombre de las claves obtenidas con el método “keys()”, sino que se concatena el string “Emails @” al dominio del email. Además, si se quiere dar cierto orden en la representación de los datos del diccionario, no vale con acceder a los valores obteniéndolos con el método “values()” y recorriéndolos después, sino que hay que ir accediendo a ellos por su nombre de clave en el orden deseado: dict[‘Clave1’], dict[‘Clave2’], etc. Esto se debe a que, aunque un diccionario es una colección de pares clave-valor ordenada desde la versión 3.7 de Python, estos diccionarios provienen de documentos JSON almacenados en Elasticsearch, que por definición son colecciones desordenadas de pares clave-valor. Al obtener los documentos JSON se hace la conversión a diccionarios de Python, cuyos campos sí están ordenados, pero el orden de estos no tiene por qué coincidir con el orden inicial que se le hubiera dado en su inserción a la base de datos.

Como en el método básico, se crea el diccionario “columnLenDictList” para llevar la referencia de la celda de cada columna con mayor número de caracteres, con el fin de que ese valor sea finalmente el del ancho de la columna correspondiente.

```
columnLenDictList.append({'column': columnNum, 'maxWidth': len(str(value))})
```

Ilustración 154 - Cuarto fragmento de código del método emailsToTables() de exportación a Excel

Una vez escrito el contenido de los diccionarios de emails, se le da formato como tabla.

```
firstRow = str(headerRow)
firstColumn = openpyxl.utils.cell.get_column_letter(maxColumn + 2)
lastColumn = openpyxl.utils.cell.get_column_letter(maxColumn + 1 + len(emailDict.keys()))
lastRow = str(rowNum)
ref = firstColumn + firstRow + ':' + lastColumn + lastRow
tableName = domain.replace('-', '_') + '_' + sheet.title.replace(' ', '')
tab = Table(displayName=tableName, ref=ref)
tab.tableStyleInfo = TableStyleInfo(name='TableStyleMedium9', showFirstColumn=False, showLastColumn=False, showRowStripes=True, showColumnStripes=False)
sheet.add_table(tab)
```

Ilustración 155 - Quinto fragmento de código del método emailsToTables() de exportación a Excel

Después se añade un diccionario con los pares clave-valor “domain” y “firstColumn” a la lista de diccionarios tablaDictList. En el campo “domain” se almacena el dominio del email de la tabla, mientras que en ‘fistColumn’ se almacena el número de la primera columna de la tabla creada.

```
tableDictList.append({'domain': domain, 'firstColumn': maxColumn + 2})
maxColumn += 1 + len(emailDict.keys())
```

Ilustración 156 - Sexto fragmento de código del método emailsToTables() de exportación a Excel

Llevar este registro es necesario por dos motivos. El primer motivo es saber si un dominio de email ha aparecido ya, para no crear más de una tabla por dominio de email. El segundo motivo tiene relación con este primero, pues si el dominio del email ya ha aparecido, es necesario conocer dónde se encuentra esta tabla para incluir el email en ella. Con guardar el número de columna correspondiente a la primera columna es suficiente, ya que se sabe el número de columnas de cada tabla, que es el número de campos de los diccionarios de emails

En la primera iteración la tabla solo va a tener una fila. Si en las siguientes iteraciones hay otro email con el mismo dominio, se inserta en esta tabla debajo de la fila ya escrita. Como ya se tiene registrado el dominio del email en uno de los diccionarios de la lista “tableDictList”, se obtiene la columna de la tabla del campo “firstColumn” del diccionario cuyo valor del campo “domain” sea corresponda con el dominio del email.

```
else:
    for tableDict in tableDictList:
        if tableDict['domain'] == domain:
            columnNum = tableDict['firstColumn']
            break
    columnLetter = openpyxl.utils.cell.get_column_letter(columnNum)
```

Ilustración 157 - Séptimo fragmento de código del método emailsToTables() de exportación a Excel

Para buscar la última fila la opción más sencilla es recorrer con un bucle “while” las celdas de la columna mientras no se encuentre una celda vacía, sumando uno a la variable “rowNum” en cada iteración. Esta variable se inicializa con el número de fila de la cabecera más dos, ya que es la fila siguiente a la última escrita en el mejor de los casos, es decir, en el caso de que la tabla solo tenga un email. El bucle finaliza cuando se encuentra una celda vacía, siendo entonces el valor la variable “rowNum” igual al número de la siguiente fila a la última escrita en la tabla.

```
while sheet.cell(row=rowNum, column=columnNum).value is not None:
    rowNum += 1
```

Ilustración 158 - Octavo fragmento de código del método emailsToTables() de exportación a Excel

Una vez se tiene la fila y la columna, se escribe el email en la celda. Además, se actualiza el valor del ancho de la columna si el contenido de la celda supera en longitud de caracteres al registrado anteriormente para esa columna.

```
value = emailDict['email']
sheet.cell(row=rowNum, column=columnNum).value = value
sheet.cell(row=rowNum, column=columnNum).alignment = Alignment(vertical='center', wrap_text=True)
for columnLenDict in columnLenDictList:
    if columnLenDict['column'] == columnNum:
        valueLen = len(str(value))
        if valueLen > columnLenDict['maxWidth']:
            columnLenDict['maxWidth'] = valueLen
        break
```

Ilustración 159 - Noveno fragmento de código del método emailsToTables() de exportación a Excel

```
for i, table in enumerate(sheet._tables):
    if table.name == tableName:
        tableRef = i
sheet._tables[tableRef] = tab
```

Ilustración 160 - Décimo fragmento de código del método emailsToTables() de exportación a Excel

Como ya se ha dado formato antes a la tabla, ahora que se ha escrito una celda más hay que volver a dar formato a la tabla para que no se quede fuera esta nueva celda. Tras definir el formato de la misma manera que se ha explicado cuando se crea una tabla nueva, no se puede añadir a la hoja con el método “sheet.add_table()”, ya que hay que modificar el formato de una tabla ya existente. Para ello hay que actualizar la referencia de la tabla, y esto se consigue recorriendo las tablas de la hoja buscando la que tiene el mismo nombre. Una vez que se ha encontrado, se actualiza en el array de tablas de la hoja.

Por último, se ajusta el ancho de las columnas con los valores registrados como se ha explicado anteriormente.

```
for columnLenDict in columnLenDictList:
    columnLetter = openpyxl.utils.cell.get_column_letter(columnLenDict['column'])
    sheet.column_dimensions[columnLetter].width = columnLenDict['maxWidth'] + 2
```

Ilustración 161 - Undécimo fragmento de código del método emailsToTables() de exportación a Excel

El resultado es el siguiente:

A	B	C	D	E	F
1	Emails @upm.es	Emails @etsisi.upm.es			
2	@upm.es	@etsisi.upm.es			
3	@upm.es	@etsisi.upm.es			
4	@upm.es	@etsisi.upm.es			
5	@upm.es	@etsisi.upm.es			
6	@upm.es	@etsisi.upm.es			
7	i@upm.es	@etsisi.upm.es			
8	@upm.es	@etsisi.upm.es			
9	@upm.es	@etsisi.upm.es			
10	@upm.es	@etsisi.upm.es			
11)@upm.es	@etsisi.upm.es			
12	@upm.es	@etsisi.upm.es			
13	@upm.es	@etsisi.upm.es			
14	@upm.es	@etsisi.upm.es			
15	@upm.es	@etsisi.upm.es			
16	@upm.es	@etsisi.upm.es			
17	@upm.es	@etsisi.upm.es			
18	@upm.es	@etsisi.upm.es			
19	@upm.es	@etsisi.upm.es			
20	@upm.es	@etsisi.upm.es			

Ilustración 162 - Hoja de Emails

Para representar los datos de los diccionarios de dominios similares y de los diccionarios de documentos, la lógica es la misma que la mostrada anteriormente, se crean tablas por cada dominio y se escriben en ella los datos correspondientes a ese dominio, disponiéndose unas al lado de las otras separadas por una columna.

El resultado para la hoja de Documentos es el siguiente:

A	B	C	D	E	F
1	upm.es	etsisi.upm.es			
2	http://o.upm.es/62375/Characterization_of_electrodeposited_zinc_oxide.pdf	http://jramiro.etsisi.upm.es/curriculum/curriculum_jra_resumen.pdf			
3	http://www.etsam.upm.es/tablon/ComunicadoU.pdf	https://www.etsisi.upm.es/sites/default/files/secretaria/legalizacion_documentos_acad			
4	http://o.upm.es/61434/First_principles_characterization_of_direct.pdf	https://www.etsisi.upm.es/sites/default/files/fsq_practicas_externas.pdf			
5	http://o.upm.es/61437/Development_and_implementation_of_the_exact.pdf	https://www.etsisi.upm.es/sites/default/files/itec_trainee_programme_flyer.pdf			
6	http://o.upm.es/53236/1336533.pdf	https://www.etsisi.upm.es/sites/default/files/masteres/normativa_tfm_2013.pdf			
7	http://o.upm.es/53854/1335046.pdf	http://franmic.etsisi.upm.es/CPaRaMinano(ESP).pdf			
8	http://o.upm.es/53151/1336062.pdf	https://www.etsisi.upm.es/sites/default/files/curso_2013_20/Grado_Planificacion/calc			
9	http://o.upm.es/60317/136051.pdf	http://www.etsisi.upm.es/sites/default/files/secretaria/periodo_extraordinario_de_mat			
10	http://o.upm.es/58457/1464240.pdf	https://www.etsisi.upm.es/sites/default/files/secretaria/resumenes_normas_de_matricula_1			
11	http://o.upm.es/58397/1335332.pdf	https://www.etsisi.upm.es/sites/default/files/curso_2018_19/Grado_Planificacion/plaz			
12	http://o.upm.es/61419/1415028.pdf	https://www.etsisi.upm.es/sites/default/files/curso_2019_20/Grado_Planificacion/plan			
13	http://o.upm.es/61458/1420034.pdf	https://www.etsisi.upm.es/sites/default/files/curso_2019_20/Grado_Planificacion/hor			
14	http://www.criptored.upm.es/descarga/Clase4Crypto4c10_2_Intercambio_clave_Diffie	https://www.etsisi.upm.es/sites/default/files/curso_2019_20/Grado_Planificacion/hor			
15	http://o.upm.es/53186/1336193.pdf	https://www.etsisi.upm.es/sites/default/files/curso_2019_20/Grado_Planificacion/plan			
16	http://o.upm.es/53209/1336260.pdf	https://www.etsisi.upm.es/sites/default/files/curso_2018_19/Grado_Planificacion/exa			
17	http://o.upm.es/62094/1406378.pdf	https://www.etsisi.upm.es/sites/default/files/ajigz/programamiento_parallel/practicas/			
18	http://o.upm.es/58309/1335197.pdf	https://www.etsisi.upm.es/sites/default/files/ajigz/programamiento_parallel/practicas/			
19	http://o.upm.es/58734/1334854.pdf	https://www.etsisi.upm.es/sites/default/files/ajigz/sistemas_operativos_CI/practicas/			
20	http://o.upm.es/58234/1336351.pdf	https://www.etsisi.upm.es/sites/default/files/ajigz/sistemas_operativos_CI/practicas/			
21	http://o.upm.es/60323/1406633.pdf	http://gjts.etsisi.upm.es/wp-			
22	http://o.upm.es/53254/1336409.pdf	https://www.etsisi.upm.es/sites/default/files/elecciones/19-20/resultadosgrupo_4.pdf			
23	http://o.upm.es/62054/17TFM_Ene20_Ormaeza_Martinez_Maria_2de2.pdf	https://www.etsisi.upm.es/sites/default/files/curso_2019_20/Grado_Planificacion/no			
24	http://o.upm.es/44634/14Edificio_intercambiador_energia.pdf	https://www.etsisi.upm.es/sites/default/files/curso_2019_20/Grado_Planificacion/itod			
25	http://dst.etsi.upm.es/gihp/cm-coronavirus	http://fmds.etsi.upm.es/wp-content/uploads/2019/07/tramProcessing.pdf			
26	http://dst.etsi.upm.es/print/56193/contests	http://www.dms.etsisi.upm.es/usuarios/fmartin/Docencia/19MD-19yMD-19-Inducción			
27	http://o.upm.es/55665/1334713.pdf	https://www.etsisi.upm.es/sites/default/files/secretaria/cambios-titulacion-19-20-1.pdf			
28	http://o.upm.es/55837/1335112.pdf	http://pastillero.etsisi.upm.es/module/pluginfile.php/85/mod_folder/content/0/TFG2			
29	http://o.upm.es/55026/1335640.pdf	http://m2de.etsi.upm.es/wp-content/uploads/2019/07/AMPLIACIONES-DE-LA-			
30	http://o.upm.es/55833/1335332.pdf	http://m2de.etsi.upm.es/wp-content/uploads/2019/07/Arquitecturas-par-			
31	http://o.upm.es/53127/1335932.pdf				
32	http://o.upm.es/55811/133592.pdf				
33	http://www.upm.es/133592/Rectorado/Gerencia/igualdad/Diversidadyleyii.pdf				
34	http://www.robolab.etsi.upm.es/avizuntas/transparencia/rrhh/Hibridas.pdf				
35	http://www.cbg.upm.es/archivos/avisos/carreras/265665215.pdf				
36	http://o.upm.es/print/59436/content				
37	http://o.upm.es/53006/1335582.pdf				
38	http://www.upm.es/133584/H334270.pdf				

Ilustración 163 - Hoja de Documentos

Sin embargo, la diferencia en el módulo de dominios similares con respecto al de emails es que se añade una captura de pantalla, por lo que en las celdas no solo se escriben caracteres, sino que también se insertan imágenes correspondientes a las capturas de los sitios web. La URL de la imagen a insertar se obtiene del diccionario “similarDomainDict”, que es uno de los diccionarios de la lista de diccionarios pasada como parámetro con el contenido de la consulta a Elasticsearch. Después se hace una petición HTTP GET con el método “requests.get()” para obtener la imagen.

```
urlImage = similarDomainDict['screenshot']
try:
    res = requests.get(urlImage, headers=headers, timeout=5)
    res.raise_for_status()
    imgFile = io.BytesIO(res.content)
```

Ilustración 164 - Primer fragmento de código del método similarDomainsToTables() de exportación a Excel

Para no descargar las imágenes en disco, se guardan en memoria mediante el método “io.BytesIO()”. Si la web presenta scrolling, lo que es habitual, la captura va a tener una altura demasiado grande. Por ello, es necesario recortar la captura para que adopte dimensiones con proporciones de altura y anchura adecuadas.

Para hacer esta modificación es necesario utilizar la librería “Pillow”, pasando el objeto de la imagen guardada en memoria al método “PIL.Image.open()” para así poder editar la imagen.

```
imgObj = PIL.Image.open(imgFile)
cropWidth = imgObj.width
cropHeight = 620
imgObj = imgObj.crop((0, 0, cropWidth, cropHeight))
```

Ilustración 165 - Segundo fragmento de código del método similarDomainsToTables() de exportación a Excel

Con el método “crop()” se recorta el alto de la imagen, manteniendo el ancho. Un paso importante es crear un nuevo objeto de la clase “io.BytesIO”, pasando dicho objeto al método “save()” para guardar como PNG en memoria la imagen editada con “Pillow”.

```
imgFile = io.BytesIO()
imgObj.save(imgFile, 'PNG')
img = Image(imgFile)
img.width = 284
img.height = (img.width*cropHeight)/cropWidth
```

Ilustración 166 - Tercer fragmento de código del método similarDomainsToTables() de exportación a Excel

Después, el objeto correspondiente a la imagen almacenada en memoria se le pasa por parámetro a la función “Image()” del módulo “openpyxl.drawing.image”, redimensionando la imagen con el tamaño deseado en pixeles para que aparezca con ese tamaño en la tabla de Excel. Se le ha dado un valor fijo de anchura, mientras que el valor de altura se calcula proporcionalmente según el recorte realizado anteriormente a la imagen, de manera que se respeten las proporciones para que no se pierda calidad.

También se tiene que ajustar el tamaño de la celda al tamaño de la imagen. El ancho de la celda no se puede calcular, ya que la unidad de medida es el número de caracteres en fuente Arial con tamaño 10 que caben en la celda, como ya se explicó anteriormente. Por tanto, se realiza una aproximación correspondiente al ancho de imagen fijado. Por otro lado, sí que es posible calcular la altura de la celda, ya que se mide en puntos. Un pixel son 0,75 puntos, por lo que tan solo hay que multiplicar esta cifra por el alto de la imagen.

```
imgColumn = openpyxl.utils.cell.get_column_letter(columnNum)
imgRow = str(rowNum)
img.anchor = imgColumn + imgRow
sheet.add_image(img)
sheet.column_dimensions[imgColumn].width = 51
sheet.row_dimensions[rowNum].height = img.height*0.75
```

Ilustración 167 - Cuarto fragmento de código del método similarDomainsToTables() de exportación a Excel

El resultado es el siguiente:

A	B	C	D	E	F	G	
1	dominio original: upmz.com	IP	Screenshot	Snapshot	positivos en VirusTotal		
2	upmv.com	45.33.2.79 45.73.10.195 198.58.103.167 45.33.20.183 96.126.123.244 45.56.79.23		326193110/http://www6.upmv.com/?tdfs=1&c_token=1585251070.0016466121&uid=1585251070.0016	0		
3	upma.com	109.248.250.100		https://web.archive.org/web/20200326193110/http://upma.com/	0		
4	upmw.com	75.126.100.21		https://web.archive.org/web/20200326193109/http://upmw.com/	0		
5	upo.com	45.33.14.247		https://web.archive.org/web/20200326193156/http://upo.com/	0		
6							

Ilustración 168 - Hoja de Posible Phishing

5.9.4 Método de creación de tablas con agrupación por filas con datos en común (fuente de localización de emails, listas negras)

Este método consiste en representar todos los datos en una misma tabla, pero agrupados por filas que tengan un determinado dato en común en su primera columna. Por ejemplo, en el caso de la fuente de la localización de emails se agrupará por email, y dado que un email puede tener varias fuentes se muestra en la primera columna el email y en la segunda sus fuentes de localización, teniendo que combinar en la primera columna tantas filas como fuentes tenga el email.

En esta tabla el caso más sencillo sería que los emails solo aparecieran en una fuente, ya que como cada diccionario tiene un campo “email” y otro campo “source”, no haría falta agrupar. Se va a mostrar directamente el caso en el que puede aparecer más de una fuente de localización de email para un mismo email, habiendo escrito ya la primera aparición del email, ya que es el que se diferencia de lo explicado hasta ahora.

```
if not any(tableDict.get('email', None) == email for tableDict in tableDictList):
    ...
    tableDictList.append({'email': email, 'firstRow': maxRow, 'lastRow': maxRow})
```

Ilustración 169 - Primer fragmento de código del método emailSourcesToTable() de exportación a Excel

Como se lleva un registro de emails escritos en la lista de diccionarios tableDictList, hay que comprobar en qué fila se encuentra la última fuente de este email para poder incluir la nueva fuente en una nueva fila. Esta fila está registrada en los diccionarios de la lista “tableDictList”, cada uno tiene el campo “email” con el email escrito y los campos “firstRow” y “lastRow”, que contienen los números de fila de su primera y última fuente escrita, respectivamente.

Tras obtener la fila en la que aparece la última fuente del email de la lista de diccionarios “tableDictList”, se inserta una fila nueva debajo de esta.

```
else:
    for i in range(len(tableDictList)):
        if tableDictList[i]['email'] == email:
            rowNum = tableDictList[i]['lastRow'] + 1
            sheet.insert_rows(rowNum)
            maxRow += 1
```

Ilustración 170 - Segundo fragmento de código del método emailSourcesToTable() de exportación a Excel

También se guarda la posición de la última fila escrita en la tabla de Excel con la variable “maxRow”, ya que es necesario llevar un registro para el caso de que aparezca un email que aún no estuviera en la tabla y deba añadirse al final, y después se escribe en la tercera columna la fuente del email.

```
columnNum = 3
value = emailDict['source']
sheet.cell(row=rowNum, column = columnNum).value = value
sheet.cell(row=rowNum, column=columnNum).alignment = Alignment(vertical='center', wrap_text=True)
sheet.cell(row=rowNum, column=columnNum).border = thinBorder
```

Ilustración 171 - Tercer fragmento de código del método emailSourcesToTable() de exportación a Excel

Además, se guarda el ancho máximo de las celdas escritas como ya se ha explicado en los métodos anteriores. Una vez escrita esta nueva fila, hay que actualizar el valor el campo “lastRow” con el número de dicha fila en del diccionario de la lista “tableDictList” correspondiente. También es necesario actualizar las referencias de los emails que puedan encontrarse debajo en la tabla de Excel, ya que al insertarse encima una nueva fila han pasado a ocupar la fila siguiente a la que se encontrasen.

```
tableDictList[i]['lastRow'] = rowNum
for j in range(i+1, len(tableDictList)):
    tableDictList[j]['firstRow'] += 1
    tableDictList[j]['lastRow'] += 1
break
```

Ilustración 172 - Cuarto fragmento de código del método emailSourcesToTable() de exportación a Excel

Una vez están todos los emails escritos con sus fuentes, se combinan las celdas correspondientes a las filas del cada email en caso de que la primera fila no coincida con la última, las cuales se han ido guardando en los diccionarios de la lista “tableDictList”.

```
for tableDict in tableDictList:
    if tableDict['firstRow'] != tableDict['lastRow']:
        sheet.merge_cells(start_row=tableDict['firstRow'], start_column=2
                          , end_row=tableDict['lastRow'], end_column=2)
```

Ilustración 173 - Quinto fragmento de código del método emailSourcesToTable() de exportación a Excel

El resultado es el siguiente:

A	B	C	D	E
1	Emails	Fuentes		
2				
3		http://www.upm.es/contacto		
4		http://www.upm.es/Estudiantes/Estudios_Titulaciones/Estudios_Master/Matricula		
5	orientacion.mater.oficina@upm.es	http://www.upm.es/Estudiantes/Estudios_Titulaciones/Estudios_Master		
6		http://www.upm.es/Estudiantes/Estudios_Titulaciones/Estudios_Master/Admision		
7		https://www.etsisi.upm.es/lmiw		
8		http://www.upm.es/UPM/SalaPrensa/NoticiasPortada/contacto		
9		http://www.upm.es/contacto		
10	orientacion.universitaria@upm.es	http://www.upm.es/UPM/SalaPrensa/NoticiasPortada/contacto		
11		http://www.upm.es/contacto		
12		http://www.upm.es/FuturosEstudiantes/Ingresar/Orientacion_Alumnos		
13		http://www.upm.es/FuturosEstudiantes/InformacionEstudiantes		
14	acceso.universitario@upm.es	https://www.upm.es/FuturosEstudiantes/Ingresar/Acceso/25anios		
15		http://www.upm.es/FuturosEstudiantes/Ingresar/Acceso/EvAU		
16		http://www.upm.es/UPM/SalaPrensa/NoticiasPortada/contacto		
17		http://www.upm.es/contacto		
18	formacion_continua@upm.es	http://www.upm.es/UPM/SalaPrensa/NoticiasPortada/contacto		
19		http://www.upm.es/contacto		
20		http://www.upm.es/FuturosEstudiantes/InformacionEstudiantes		
21	gestion_academica@upm.es	http://www.upm.es/Estudiantes/OrdenacionAcademica/Matricula		
22		http://www.upm.es/UPM/SalaPrensa/NoticiasPortada/contacto		
23		http://www.upm.es/contacto		
24	trabajo.propio@upm.es	http://www.upm.es/UPM/SalaPrensa/NoticiasPortada/contacto		
25		http://www.upm.es/contacto		
26		http://www.upm.es/FuturosEstudiantes/Ingresar/Orientacion_Alumnos		
27		http://www.upm.es/FuturosEstudiantes/InformacionEstudiantes		
28	admitirongradu@upm.es	https://www.upm.es/FuturosEstudiantes/Ingresar/Acceso/Admision/PGGrado		
29		https://www.upm.es/FuturosEstudiantes/Ingresar/Acceso/Admision/TEExpediente		

Ilustración 174 - Hoja de Fuentes de localización de email con resultados de UPM

A	B	C	D	E
193	irina.arguelles@upm.es	http://www.aesla.org.es/es/paneles-cientificos		
194		https://www.etsisi.upm.es/estudios/master/mdasdmlig/personal		
195	jerome.jeron@etsisi.upm.es	https://www.etsisi.upm.es/estudios/doctorado/61d1lpal/Workshop14		
196		https://www.etsisi.upm.es/estudios/doctorado/61d1lpal/Workshop15		
197		https://www.etsisi.upm.es/secretaria-alumnos/localizacion		
198		https://www.etsisi.upm.es/informacion/C3xB3n_general		
199		https://www.etsisi.upm.es/estudios/grados/61cille/matricula		
200		https://www.etsisi.upm.es/estudios/grados/61cille/matricula		
201		https://www.etsisi.upm.es/estudios/master/mctclie/plazos		
202		https://www.etsisi.upm.es/estudios/master/mctclie/matricula		
203		https://www.etsisi.upm.es/futuros_alumnos/		
204		https://www.etsisi.upm.es/estudios/grados/61cille/matricula		
205		https://www.etsisi.upm.es/int_general?page=4&qt=calendario_zircon=1		
206		https://www.etsisi.upm.es/estudios/grados/61cille/matricula		
207	llorente@etsisi.upm.es	https://www.etsisi.upm.es/estudios/master/mdasdmlig/personal		
208	llorente@etsisi.upm.es	https://www.etsisi.upm.es/estudios/master/mdasdmlig/personal		
209	llorente@etsisi.upm.es	https://www.etsisi.upm.es/estudios/master/mdasdmlig/personal		
210		https://www.etsisi.upm.es/practicas-externas		
211		https://www.etsisi.upm.es/practicasexternas		
212		http://www.etsisi.upm.es/practicasexternas		
213		https://llope.etsisi.upm.es/		
214		https://www.etsisi.upm.es/estudios/master/mdasdmlie		
215		https://www.etsisi.upm.es/estudios/master/mdasdmlig		
216		https://www.etsisi.upm.es/estudios/master/mdasdmlie/carac		
217		https://www.etsisi.upm.es/estudios/master/mdasdmlie/matricula		
218		https://www.etsisi.upm.es/estudios/master/mdasdmlie		
219		https://www.etsisi.upm.es/estudios/master/mctclie/plazos		
220		https://www.etsisi.upm.es/estudios/master/mctclie/matricula		
221	llorente@etsisi.upm.es	https://www.etsisi.upm.es/estudios/master/mdasdmlig		

Ilustración 175 - Hoja de Fuentes de localización de email con resultados de ETSISI

5.9.5 Método de creación de tablas con combinación de columnas (metadatos)

En este método todos los diccionarios se van a representar en una misma tabla como en el método anterior de agrupación por filas. Sin embargo, este método es más sencillo, ya que estos diccionarios se van escribiendo de manera secuencial unos debajo de otros sin ninguna agrupación.

El formato particular que tiene es que un dato del diccionario ocupa dos celdas contiguas combinadas, de modo que hace de encabezado de los otros datos que ocupan las filas inferiores. Aunque todas las tablas estén juntas unas debajo de otras, realmente son distintos diccionarios cuyos datos se encuentran separados por los encabezados.

Los campos de la lista de diccionarios “metadataDictList” son “url” y “metadata”, el cual contiene otro diccionario con los metadatos. La URL es el campo que hace de encabezado y los metadatos van debajo. En la primera columna se muestra el tipo de metadato, y en la segunda el propio metadato. El único registro que hay que llevar en este caso es el de la última fila escrita, que se irá incrementando por cada fila que se escriba.

Como en este caso no da formato como tabla, sino que simplemente se crea la tabla añadiendo los bordes de las celdas. Para poner un borde grosor medio se va a inicializar la variable “mediumBorder” a la que se accederá posteriormente.

```
def metadataToTable(self, sheet, metadataDictList):
    columnNum = 2
    rowNum = 2
    maxLengthURL = 0
    maxLengthMetadataKey = 0
    for metadataDict in metadataDictList:
        columnNum = 2
        mediumBorder = Border(left=Side(style='medium'), right=Side(style='medium'), top=Side(style='medium'), bottom=Side(style='medium'))
        sheet.cell(row=rowNum, column=columnNum).value = metadataDict['url']
        sheet.cell(row=rowNum, column=columnNum).alignment = Alignment(vertical='center')
```

Ilustración 176 - Primer fragmento de código del método metadataToTable() de exportación a Excel

Para escribir los datos en Excel es necesario obtener el valor de cada campo, ya que como ya se ha explicado anteriormente los diccionarios con estos datos provienen de documentos JSON almacenados en Elasticsearch, y los documentos JSON son colecciones desordenadas de pares clave-valor. Por lo tanto, si se obtuvieran los valores con el método “values()”, se desconocería a qué campo corresponden.

De este modo, primeramente se accede al campo “url” y después se accede al campo “metadata”, que contiene a su vez un diccionario con los metadatos como ya se ha comentado. Tras escribir la URL se alinea el texto en el medio de forma vertical, se combinan las celdas contiguas y se pone el borde medio en las celdas combinadas.

```
sheet.cell(row=rowNum, column=columnNum).value = metadataDict['url']
sheet.cell(row=rowNum, column=columnNum).alignment = Alignment(vertical='center')
sheet.merge_cells(start_row=rowNum, start_column=2, end_row=rowNum, end_column=3)
sheet.cell(row=rowNum, column=columnNum).border = mediumBorder
sheet.cell(row=rowNum, column=columnNum+1).border = mediumBorder
```

Ilustración 177 - Segundo fragmento de código del método metadataToTable() de exportación a Excel

Para ajustar posteriormente el ancho de la columna en la que se escriben las URLs, hay que guardar la longitud máxima de estas.

```
length = len(metadataDict['url'])
if length > maxLengthURL:
    maxLengthURL = length
rowNum += 1
```

Ilustración 178 - Cuarto fragmento de código del método metadataToTable() de exportación a Excel

Después se pasa a la fila siguiente, donde se van a escribir los metadatos. Estos metadatos se obtienen del diccionario contenido en el campo “metadata” del diccionario “metadataDict”. Con el método “items()” se obtiene de este diccionario una lista con los pares clave-valor “metadataKey” y “metadataValue”, los cuales se corresponden con el nombre del metadato y su valor, respectivamente.

```
thinBorder = Border(left=Side(style='thin'), right=Side(style='thin'),
top=Side(style='thin'), bottom=Side(style='thin'))
for metadataKey, metadataValue in metadataDict['metadata'].items():
    try:
        sheet.cell(row=rowNum, column=columnNum).value = metadataKey
        sheet.cell(row=rowNum, column=columnNum).border = thinBorder
        sheet.cell(row=rowNum, column=columnNum).alignment = Alignment(
            vertical='center', wrap_text=True)
```

Ilustración 179 - Quinto fragmento de código del método metadataToTable() de exportación a Excel

Una vez escrito el nombre del metadato contenido en la variable “metadataKey”, se comprueba si su longitud es mayor que la máxima escrita hasta el momento, y en tal caso se guarda para el posterior ajuste del ancho de la columna.

```
length = len(metadataKey)
if length > maxLengthMetadataKey:
    maxLengthMetadataKey = length
```

Ilustración 180 - Sexto fragmento de código del método metadataToTable() de exportación a Excel

En la siguiente columna se escribe el valor del metadato contenido en la variable “metadataValue”, con las mismas propiedades de celda que en el caso anterior.

```
sheet.cell(row=rowNum, column=columnNum+1).value = metadataValue
sheet.cell(row=rowNum, column=columnNum+1).border = thinBorder
sheet.cell(row=rowNum, column=columnNum+1).alignment = Alignment(vertical='center', wrap_text=True)
rowNum += 1
```

Ilustración 181 - Séptimo fragmento de código del método metadataToTable() de exportación a Excel

Por último, tras escribir todas las claves y valores de los metadatos se ajusta el ancho de las columnas. En el caso de la segunda columna, para ajustarla a la longitud máxima de las URLs se resta el tamaño de la primera columna, ya que están combinadas.

```
columnLetter = openpyxl.utils.cell.get_column_letter(columnNum)
sheet.column_dimensions[columnLetter].width = maxLengthMetadataKey + 1
columnLetter = openpyxl.utils.cell.get_column_letter(columnNum + 1)
sheet.column_dimensions[columnLetter].width = maxLengthURL - maxLengthMetadataKey
```

Ilustración 182 - Octavo fragmento de código del método metadataToTable() de exportación a Excel

El resultado es el siguiente:

A	B	C	D	E
1				
2	http://osspm.csic.es/2375/II/Characterisation_of_electrodeposited_zinc_oxide.pdf			
3	CreationDate	D20200310182806Z		
4	Producer	ABEYY FineReader 8.0 Site License Edition		
5	Author	Bernabé		
6	Title	Characterisation of electrodeposited zinc oxide/tetravalent copper phthalocyanine (ZnO/T _x CuPc) hybrid films and their photoelectrochemical properties		
7	ModDate	D20200310182700Z		
8	Subject	Journal of Electroanalytical Chemistry, 653 (2011) 86-92, doi:10.1016/j.jelechem.2010.12.023		
9	http://www.ctpm.csic.es/bases/ConsejosB.pdf			
10	CreationDate	D202003100085-0700Z		
11	Company	Universidad Politécnica de Madrid		
12	Producer	Adobe PDF Library 15.0		
13	SourceModified	D2020031009025		
14	Author	Alfonso		
15	Creator	Acrobat PDFMaker 15 para Word		
16	ModDate	D2020031000301-0700Z		
17	http://osspm.csic.es/6434/I/Final_principles_characterisation_of_direct.pdf			
18	CreationDate	D20200223100045+0700Z		
19	Producer	Acrobat		
20	Title	doi:10.1016/j.solidstate.2020.020425-1		
21	ModDate	D20200223100055+0700Z		
22	http://osspm.csic.es/6439/IV/Development_and_implementation_of_the_exact.pdf			
23	CreationDate	D20200223100329+0700Z		
24	Producer	ABEYY FineReader 10		
25	Title	doi:10.1016/j.solidstate.2020.020413-T		
26	ModDate	D20200223100410+0700Z		
27	http://osspm.csic.es/9336/II/33653.pdf			
28	CreationDate	D20200223132306		
29	Producer	file:///usr/share/doc/magick-6-common/html/index.html		
30	ModDate	D20200223132306		
31	http://osspm.csic.es/58854/II/33551.pdf			
32	CreationDate	D20200223132901		
33	Producer	file:///usr/share/doc/magick-6-common/html/index.html		
34	ModDate	D20200223132901		
35	http://osspm.csic.es/59159/II/33606.pdf			
36	CreationDate	D20200223132632		
37	Producer	file:///usr/share/doc/magick-6-common/html/index.html		
38	ModDate	D20200223132632		

Ilustración 183 - Hoja de Metadatos

6 Guía de uso

6.1 Recon-vd

6.1.1 Instalación

Nuestra versión de Recon-*ng* necesita la instalación previa de Elasticsearch, ya que se utiliza como base de datos. También se deben instalar las dependencias con el comando `pip3 install -r REQUIREMENTS`. Una vez instaladas, y con Elasticsearch funcionando, ya se puede ejecutar Recon-*vd*.

Al ejecutarlo por primera vez aparece el mensaje: "no se han detectado módulos". Esto es así porque los módulos se tienen que añadir manualmente en la carpeta "modules" dentro de la carpeta oculta ".recon-vd", que se ha creado tras la ejecución.

```
kali㉿kali:~/Desktop/recon-vd$ ./recon-vd
[*] Created Index default
[*] Version check disabled.

Sponsored by ...
[ ] [ ] [ ] [ ] [ ] [ ]
ciberseguridad.oesia.com

[recon-vd v5.1.0, Tim Tomes (@lanmaster53), Rubén Álvarez, Rodrigo Baladrón]

[*] No modules enabled/installed.

[recon-vd][default] > █
```

Ilustración 184 - Ejecución de Recon-vd

En la carpeta “.recon-vd” se encuentra la carpeta con el workspace “default” y las carpetas “data” y “modules”. La carpeta “modules” contiene los módulos que se van a utilizar, mientras que la carpeta “data” contiene los archivos utilizados en los módulos.

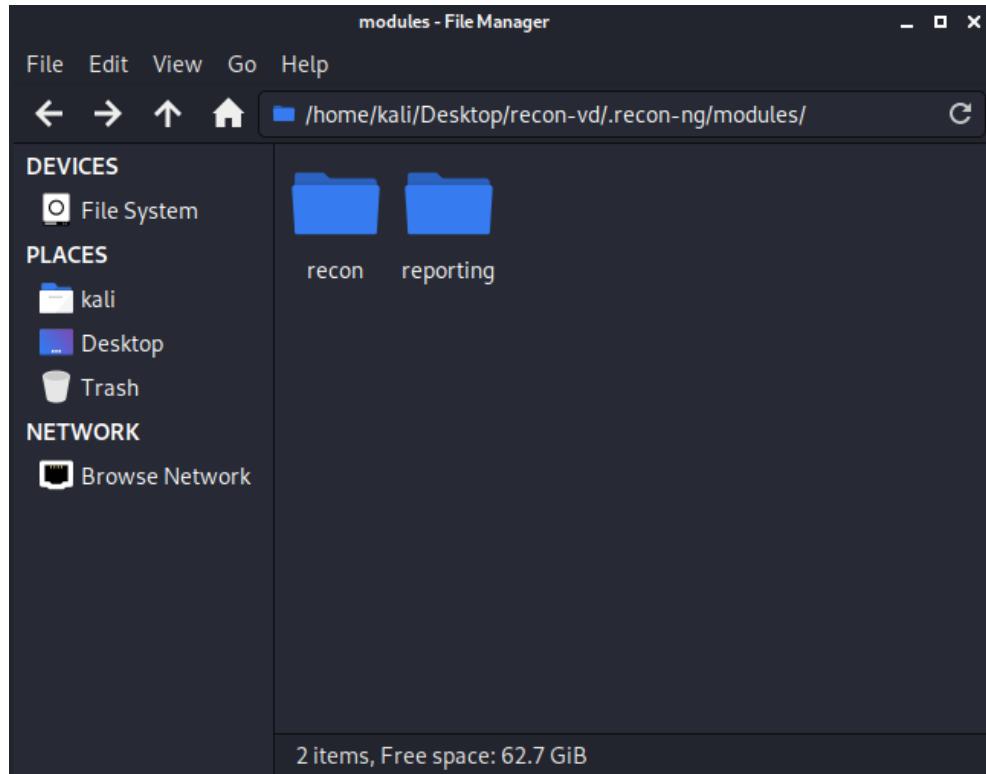


Ilustración 185 - Carpeta "modules"

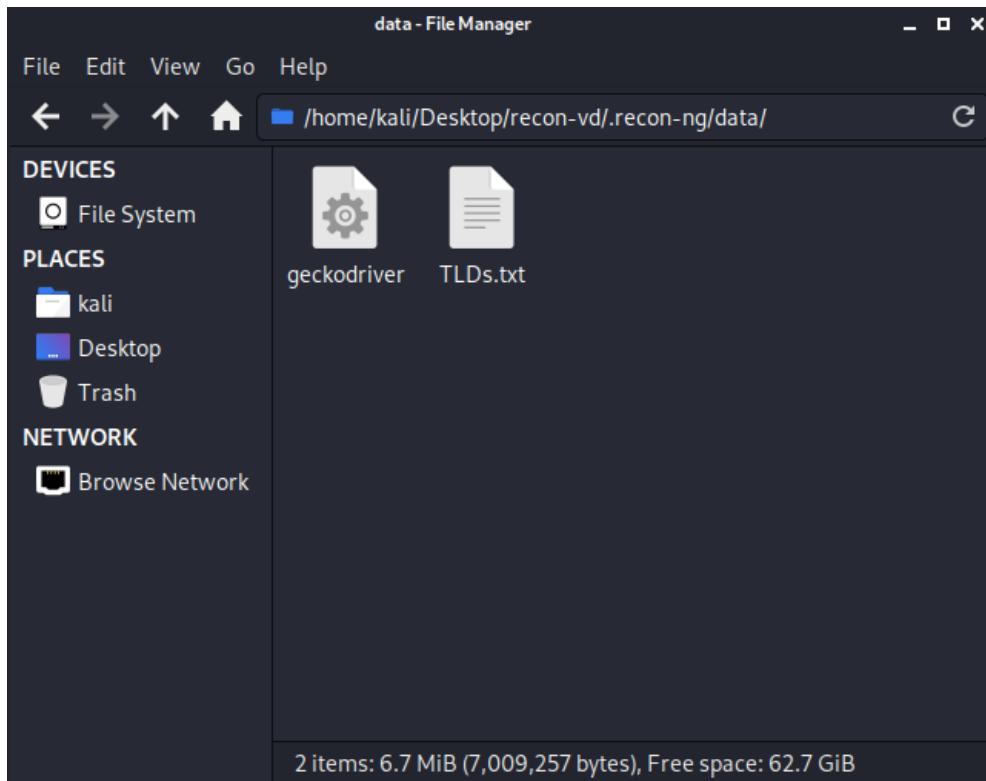


Ilustración 186 - Carpeta "data"

Una vez añadidos los módulos y los archivos correspondientes, se ejecuta el comando **modules reload** para cargar los módulos disponibles.

```
kali㉿kali:~/Desktop/recon-vd
File Actions Edit View Help
Sponsored by ... [REDACTED]
[recon-vd v5.1.0, Tim Tomes (@lanmaster53), Rubén Álvarez, Rodrigo Baladrón]
[*] No modules enabled/installed.

[recon-vd][default] > modules reload
[*] Reloading modules ...
[!] 'hunter_api' key not set. emails_hunter module will likely fail at runtime. See 'keys add'.
[recon-vd][default] >
```

Ilustración 187 - Ejecución del comando "modules reload"

6.1.2 API Keys

Hay módulos que necesitan de API keys para su funcionamiento. Estas API keys se introducen con el comando **keys**, que tiene diferentes funcionalidades:

keys <add | list | remove> [. . .]

- **keys add <nombre_API> <key>**: almacena la API key
- **keys list**: muestra la lista de API keys y sus valores
- **keys remove <nombre_API>**: borra la API key

```
[recon-vd][default] > keys add hunter_api
[*] Key 'hunter_api' added.
[recon-vd][default] > keys list

+-----+-----+
|     Name      |          Value       |
+-----+-----+
| hunter_api   | [REDACTED] |
+-----+-----+
```

Ilustración 188 - Introducción de API Key

6.1.3 Workspace

Para comenzar a usar Recon-vd, es recomendable crear un nuevo espacio de trabajo (workspace) en lugar de usar el “default”, ya que así los datos adquiridos se almacenarán separados en distintos índices de Elasticsearch.

Las opciones del comando workspace son las siguientes:

workspaces <create | list | load | remove> [. . .]

- **workspaces create <nombre_workspace>**: crea un workspace con el nombre dado
- **workspaces list**: lista los workspaces disponibles
- **workspaces load <nombre_workspace>**: carga el workspace
- **workspaces remove <nombre_workspace>**: borra el workspace

```
[recon-vd][default] > workspaces create upm
[*] Created Index upm
[recon-vd][upm] > █
```

Ilustración 189 - Ejecución del comando “workspaces create”

6.1.4 Módulos

Con el workspace creado se puede empezar a usar los módulos para ir recopilando información sobre el objetivo. La gestión de los módulos se realiza con el comando modules que tiene varias opciones:

modules <load | reload | search> [. . .]

- **modules load <ruta_módulo>**: carga el módulo
- **modules reload**: recarga los módulos
- **modules search**: muestra todos los módulos disponibles

```
[recon-vd][upm] > modules load recon/domains-emails/emails
[recon-vd][upm][emails] > █
```

Ilustración 190 - Ejecución del comando “modules load”

```
[recon-vd][upm] > modules search

Recon
-----
recon/brands-news/news
recon/brands-pastes/pastes
recon/brands-posts/forums
recon/domains-documents/docs
recon/domains-emails/emails
recon/domains-emails/emails_hunter
recon/domains-similarDomains/phishing
recon/domains-spamMailServers/blacklist_check

Reporting
-----
reporting/excelVD
```

Ilustración 191 - Ejecución del comando "modules search"

1.1.1.1 Opciones

Cada módulo dispone de sus propias opciones. Para gestionar estas opciones se hace uso del comando options:

options <list | set | unset> [. . .]

- **options list**: muestra la lista de opciones del módulo

```
[recon-vd][upm][emails] > options list

Name      Current Value  Required  Description
-----  -----
DATE          no        Last hour, day, week, month or year: h, d, w, m, y
SOURCE    default      yes       source of input (see 'info' for details)

[recon-vd][upm][emails] > █
```

Ilustración 192 - Ejecución del comando "options list"

- **options set <nombre_opción> <valor>**: establece el valor de la opción

```
[recon-vd][upm][emails] > options set DATE m
DATE ⇒ m
[recon-vd][upm][emails] > █
```

Ilustración 193 - Ejecución del comando "options set"

- **options unset <nombre_opción>**: quita el valor establecido en la opción

```
[recon-vd][upm][emails] > options unset DATE
DATE ⇒ None
[recon-vd][upm][emails] > █
```

Ilustración 194 - Ejecución del comando "options unset"

1.1.1.2 Input

Los módulos también disponen de la opción SOURCE para definir el input. Por defecto el valor de esta opción es “default”, de manera que el input se obtiene mediante una consulta a Elasticsearch.

```
[recon-vd][upm][emails] > options set SOURCE upm.es
SOURCE => upm.es
[recon-vd][upm][emails] > options list

  Name      Current Value  Required  Description
  -----  -----  -----  -----
  DATE          no        no        Last hour, day, week, month or year: h, d, w, m, y
  SOURCE        upm.es    yes       source of input (see 'info' for details)
```

Ilustración 195 - Definición del input del módulo de manera manual

Se puede comprobar el input del módulo con el comando **input**.

```
[recon-vd][upm][emails] > input
[*] Document(s) read:
[{'_index': 'upm', '_type': '_doc', '_id': 'GNsau3AB5Gzd04ht4bvK', '_score': 0.2876821, '_source': {'domain': 'upm.es'}}]

+-----+
| Module Inputs |
+-----+
| upm.es         |
+-----+
```

Ilustración 196 - Ejecución del comando “input”

El input de los módulos desarrollados son dominios o nombres de marca dependiendo del módulo. Si queremos que se obtenga el input de Elasticsearch hay que indexarlo primero, para ello se pueden usar las “Dev Tools” de Kibana:

- Inserción de un dominio

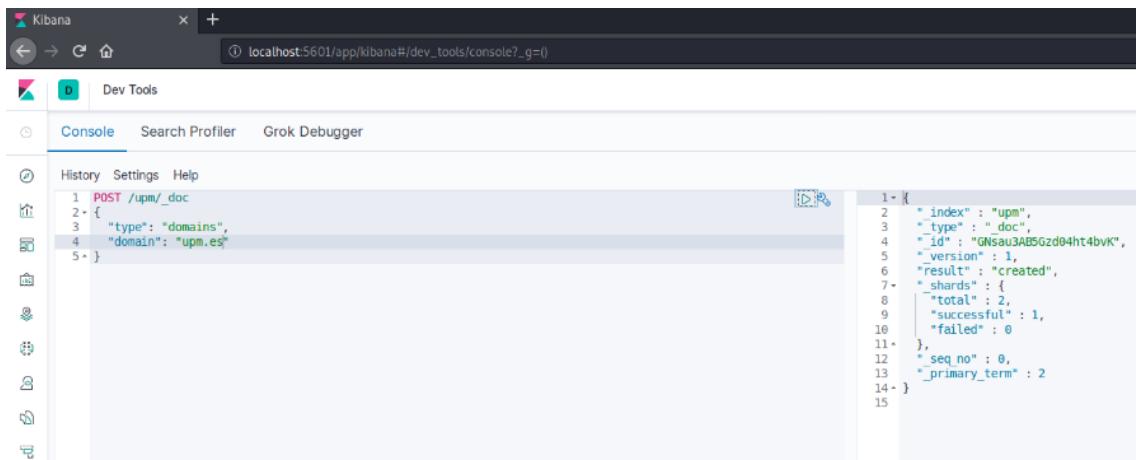
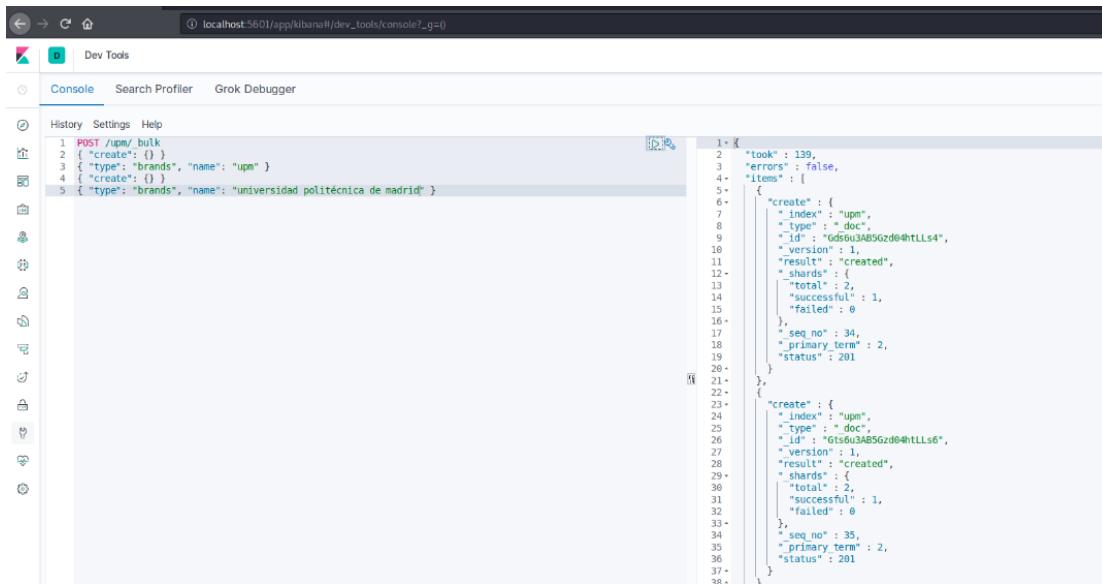


Ilustración 197 - Inserción de un dominio en Elasticsearch utilizando las “Dev Tools” de Kibana

- Inserción de varios nombres de marca



The screenshot shows the Kibana Dev Tools interface with the 'Console' tab selected. The URL in the address bar is `localhost:5601/app/kibana#/dev_tools/console?_g=()`. The console window displays the following Elasticsearch bulk indexing command and its successful execution:

```

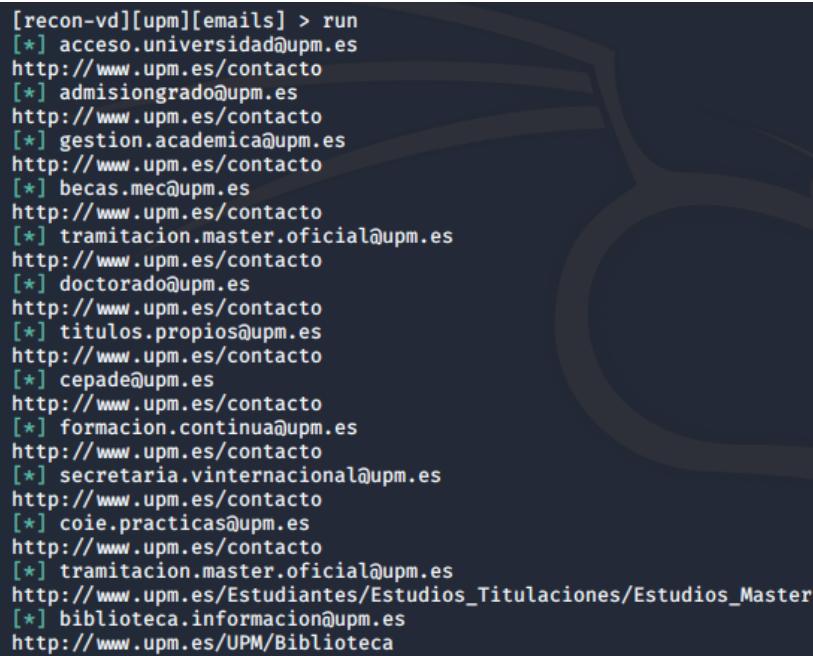
POST /upm/_bulk
1 { "create": { "_index": "brands", "name": "upm" } }
2 { "create": {} }
3 { "type": "brands", "name": "universidad politécnica de madrid" }
4 { "type": "brands", "name": "upm" }
5 { "type": "brands", "name": "universidad politécnica de madrid" }

1+ {
  "took": 139,
  "errors": false,
  "items": [
    {
      "create": {
        "_index": "upm",
        "_type": "doc",
        "_id": "G650u3A05Gzd0@tLLs4",
        "_version": 1,
        "result": "created",
        "shards": {
          "total": 2,
          "successful": 1,
          "failed": 0
        },
        "seq_no": 34,
        "primary_term": 2,
        "status": 201
      }
    },
    {
      "create": {
        "_index": "upm",
        "_type": "doc",
        "_id": "G156u3A05Gzd0@tLLs6",
        "_version": 1,
        "result": "created",
        "shards": {
          "total": 2,
          "successful": 1,
          "failed": 0
        },
        "seq_no": 35,
        "primary_term": 2,
        "status": 201
      }
    }
  ]
}

```

Ilustración 198 - Inserción de nombres de marca en Elasticsearch utilizando Kibana

Tras cargar el módulo y configurar sus opciones, se ejecuta con el comando `run`:



The screenshot shows the terminal output of the Reconnaissance module's 'Emails' module. The command run was executed in the directory [recon-vd][upm][emails]. The output lists various email addresses and their corresponding URLs:

```

[recon-vd][upm][emails] > run
[*] acceso.universidad@upm.es
http://www.upm.es/contacto
[*] admisiongrado@upm.es
http://www.upm.es/contacto
[*] gestion.academica@upm.es
http://www.upm.es/contacto
[*] becas.mec@upm.es
http://www.upm.es/contacto
[*] tramitacion.master.oficial@upm.es
http://www.upm.es/contacto
[*] doctorado@upm.es
http://www.upm.es/contacto
[*] titulos.propios@upm.es
http://www.upm.es/contacto
[*] cepade@upm.es
http://www.upm.es/contacto
[*] formacion.continua@upm.es
http://www.upm.es/contacto
[*] secretaria.vinternacional@upm.es
http://www.upm.es/contacto
[*] coie.practicas@upm.es
http://www.upm.es/contacto
[*] tramitacion.master.oficial@upm.es
http://www.upm.es/Estudiantes/Estudios_Titulaciones/Estudios_Master
[*] biblioteca.informacion@upm.es
http://www.upm.es/UPM/Biblioteca

```

Ilustración 199 - Ejecución del módulo de Emails

6.1.5 Ejecución de comandos desde un fichero

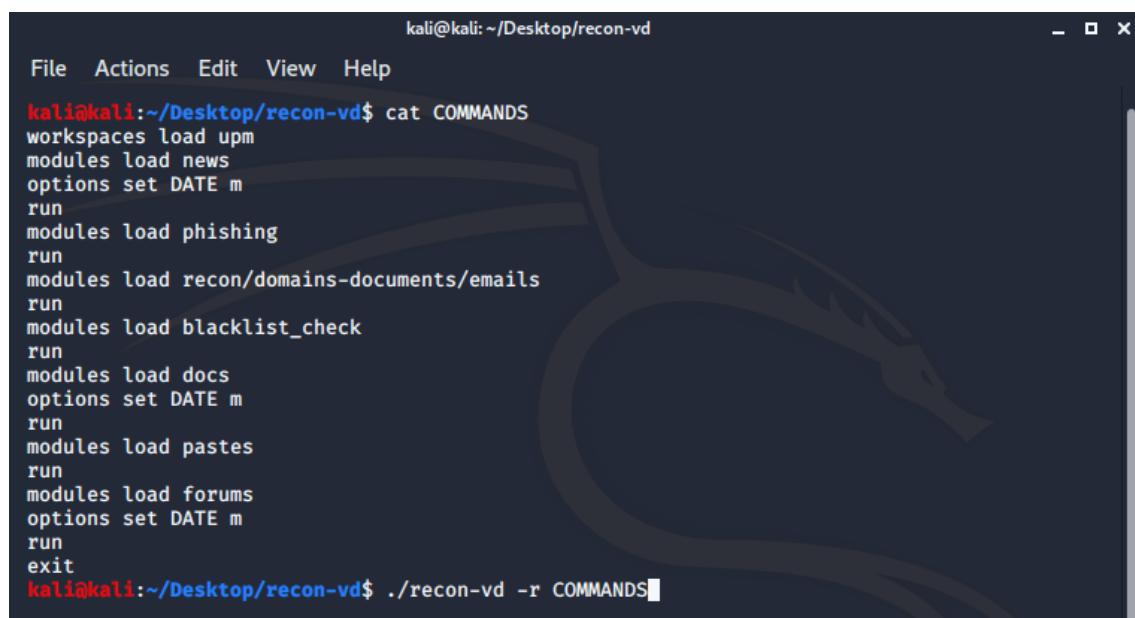
Este framework dispone de dos opciones que permiten la ejecución de los comandos cargándolos de un fichero. Desde Recon se puede hacer con el comando:

```
script execute <ruta_fichero>.
```

Y para ejecutarlo desde la línea de comandos de Linux:

```
~$./recon-vd -r <ruta_fichero>
```

Este fichero debe contener los comandos de Recon que se pretenden ejecutar, separados por saltos de línea.



The screenshot shows a terminal window titled 'kali@kali: ~/Desktop/recon-vd'. The window has a standard Linux desktop interface with a menu bar (File, Actions, Edit, View, Help) and a window control bar (minimize, maximize, close). The terminal content is as follows:

```
kali@kali:~/Desktop/recon-vd$ cat COMMANDS
workspaces load upm
modules load news
options set DATE m
run
modules load phishing
run
modules load recon/domains-documents/emails
run
modules load blacklist_check
run
modules load docs
options set DATE m
run
modules load pastes
run
modules load forums
options set DATE m
run
exit
kali@kali:~/Desktop/recon-vd$ ./recon-vd -r COMMANDS
```

Ilustración 200 - Ejecución del módulo de Emails

6.2 Instalación de Elasticsearch y Kibana

Para llevar a cabo la instalación se descarga tanto Elasticsearch como Kibana y se descomprimen los .zip.

En primer lugar, se ejecuta Elasticsearch mediante el comando bin/elasticsearch (o bin\elasticsearch.bat en Windows):

```
kali@kali:~/Desktop/elasticsearch-7.5.2$ bin/elasticsearch
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
OpenJDK 64-Bit Server VM warning: Option useConcMarkSweepGC was deprecated in version 9.0 and will likely be removed in a future release.
[2020-03-07T13:51:51,920][INFO ][o.e.e.NodeEnvironment ] [kali] using [1] data paths, mounts [[/ (/dev/sda1)]], net u
sable_space [62.7gb], net total_space [76.2gb], types [ext4]
[2020-03-07T13:51:51,924][INFO ][o.e.e.NodeEnvironment ] [kali] heap size [1007.3mb], compressed ordinary object poin
ters [true]
[2020-03-07T13:51:51,919][INFO ][o.e.n.Node          ] [kali] node name [kali], node ID [vHcFlANqTL67EtbDDt2gw], c
luster name [elasticsearch]
[2020-03-07T13:51:51,920][INFO ][o.e.n.Node          ] [kali] version[7.5.2], pid[1816], build[default/tar/8bec50e1
e0ad29ad5653712cf3bb580dcfa0df/2020-01-15T12:11:52.3135762], JVM[AdoptOpenJDK/Open
JDK 64-Bit Server VM/13.0.1/13.0.1+9]
[2020-03-07T13:51:51,921][INFO ][o.e.n.Node          ] [kali] JVM home [/home/kali/Desktop/elasticsearch-7.5.2/jdk]
[2020-03-07T13:51:51,922][INFO ][o.e.n.Node          ] [kali] JVM arguments [-Des.networkaddress.cache.ttl=60, -Des
.networkaddress.cache.negative.ttl=10, -XX:+AlwaysPreTouch, -Xss1m, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Dj
na.nosys=true, -XX:-OptimizeStackTraceInFastThrow, -Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -Dio.nett
y.recycler.maxCapacityPerThread=0, -Dio.netty.allocator.numDirectArenas=0, -Dlog4j.shutdownHookEnabled=false, -Dlog4j2.d
isable.jmx=true, -Djava.locale.providers=COMPAT, -Xms1g, -Xmx1g, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccupancyFrac
tion=75, -XX:+UseCMSInitiatingOccupancyError, -Djava.io.tmpdir=/tmp/elasticsearch-9031563079137064874, -XX:+HeapDumpOnOut
OfMemoryError, -XX:HeapDumpPath=data, -XX:ErrorFile=logs/hs_err_pid%p.log, -Xlog:gc*, gc+age=trace,safepoint:file=logs/gc
.log:utctime,pid,tags:filecount=32, filesize=64m, -XX:MaxDirectMemorySize=536870912, -Des.path.home=/home/kali/Desktop/el
asticsearch-7.5.2, -Des.path.conf=/home/kali/Desktop/elasticsearch-7.5.2/config, -Des.distribution.flavor=default, -Des
distribution.type=tar, -Des.bundled_jdk=true, -Dawt.useSystemAAFontSettings=on, -Dswing.aatext=true]
[2020-03-07T13:51:51,923][INFO ][o.e.p.PluginsService ] [kali] loaded module [aggs-matrix-stats]
[2020-03-07T13:51:51,924][INFO ][o.e.p.PluginsService ] [kali] loaded module [analysis-common]
[2020-03-07T13:51:51,925][INFO ][o.e.p.PluginsService ] [kali] loaded module [flattened]
[2020-03-07T13:51:51,926][INFO ][o.e.p.PluginsService ] [kali] loaded module [frozen-indices]
[2020-03-07T13:51:51,927][INFO ][o.e.p.PluginsService ] [kali] loaded module [ingest-common]
[2020-03-07T13:51:51,928][INFO ][o.e.p.PluginsService ] [kali] loaded module [ingest-geopip]
[2020-03-07T13:51:51,929][INFO ][o.e.p.PluginsService ] [kali] loaded module [ingest-user-agent]
[2020-03-07T13:51:51,930][INFO ][o.e.p.PluginsService ] [kali] loaded module [lang-expression]
```

Ilustración 201 - Ejecución de Elasticsearch

Una vez Elasticsearch está en funcionamiento, se ejecuta Kibana mediante el comando bin/kibana (o bin\ kibana.bat en Windows):

```
kali@kali:~/Desktop/kibana-7.5.2-linux-x86_64$ bin/kibana
log [19:20:44.927] [info][plugins-system] Setting up [15] plugins: [timelion,features,code,licensing,spaces,security
,uiActions,expressions,newsfeed,data,inspector,embeddable,advancedUiActions,eui_utils,translations]
log [19:20:44.941] [info][plugins][timelion] Setting up plugin
log [19:20:44.944] [info][features][plugins] Setting up plugin
log [19:20:44.945] [info][code][plugins] Setting up plugin
log [19:20:44.947] [info][licensing][plugins] Setting up plugin
log [19:20:44.951] [info][plugins][spaces] Setting up plugin
log [19:20:44.960] [info][plugins][security] Setting up plugin
log [19:20:44.962] [warning][config][plugins][security] Generating a random key for xpack.security.encryptionKey. To
prevent sessions from being invalidated on restart, please set xpack.security.encryptionKey in kibana.yml
log [19:20:45.114] [warning][config][plugins][security] Session cookies will be transmitted over insecure connection
s. This is not recommended.
log [19:20:45.162] [info][data][plugins] Setting up plugin
log [19:20:45.164] [info][plugins][translations] Setting up plugin
log [19:21:07.128] [warning][legacy-plugins] Skipping non-plugin directory at /home/kali/Desktop/kibana-7.5.2-linux-
x86_64/src/legacy/core_plugins/visualizations
log [19:21:09.164] [info][licensing][plugins] Imported changed license information from Elasticsearch for the [data]
cluster: type: basic | status: active
log [19:21:10.731] [info][plugins-system] Starting [8] plugins: [timelion,features,code,licensing,spaces,security,da
ta,translations]
log [19:21:21.678] [info][status][plugin:kibana@7.5.2] Status changed from uninitialized to green - Ready
log [19:21:21.693] [info][status][plugin:elasticsearch@7.5.2] Status changed from uninitialized to yellow - Waiting
for Elasticsearch
log [19:21:21.699] [info][status][plugin:xpack_main@7.5.2] Status changed from uninitialized to yellow - Waiting for Elas
ticsearch
log [19:21:21.729] [info][status][plugin:graph@7.5.2] Status changed from uninitialized to yellow - Waiting for Elas
ticsearch
log [19:21:21.769] [info][status][plugin:monitoring@7.5.2] Status changed from uninitialized to green - Ready
log [19:21:21.784] [info][status][plugin:spaces@7.5.2] Status changed from uninitialized to yellow - Waiting for Ela
sticsearch
```

Ilustración 202 - Ejecución de Kibana

```
log [19:21:29.722] [info][listening] Server running at http://localhost:5601
log [19:21:29.799] [info][server][Kibana][http] http server running at http://localhost:5601
```

Ilustración 203 - Ejecución de Kibana en el puerto 5601

Para abrir Kibana desde el navegador hay que acceder a <http://localhost:5601>

The screenshot shows the Kibana home page with a sidebar containing links like Discover, Visualize, Dashboard, Canvas, Maps, Machine Learning, Metrics, Logs, APM, Uptime, SIEM, Dev Tools, Stack Monitoring, and Management. The main content area is titled "Add Data to Kibana" and includes sections for APM, Logging, Metrics, and SIEM, each with an "Add" button. Below these are buttons for "Add sample data", "Upload data from log file", and "Use Elasticsearch data". Another section titled "Visualize and Explore Data" shows options for APM and Canvas, and a "Manage and Administer the Elastic Stack" section with Console and Index Patterns.

Ilustración 204 - Pantalla inicial de Kibana

Después, para explorar y visualizar datos en Kibana, se crea un “index pattern” en Management > Index Patterns. De esta manera, en el primer paso se define el “index pattern”, indicando a Kibana los índices de Elasticsearch que contienen los datos:

The screenshot shows the "Create index pattern" step 1 interface. It has a sidebar with Management, Index Patterns, and Create index pattern selected. The main form has a title "Step 1 of 2: Define index pattern" and a text input field containing "upm". A success message says "Success! Your index pattern matches 4 index." There are buttons for "Next step" and "Rows per page: 10".

Ilustración 205 - Creando un "index pattern" en Kibana (paso 1)

En el segundo paso, si Kibana detecta un campo timestamp, se puede seleccionar dicho campo para filtrar por fecha:

The screenshot shows the "Create index pattern" step 2 interface. It has a sidebar with Management, Index Patterns, and Create index pattern selected. The main form has a title "Step 2 of 2: Configure settings" and a dropdown for "Time Filter field name" set to "timestamp". A note says "The Time Filter will use this field to filter your data by time. You can choose not to have a time field, but you will not be able to narrow down your data by a time range." There are buttons for "Show advanced options", "Back", and "Create index pattern".

Ilustración 206 - Creando un "index pattern" (paso 2)

Una vez creado el “index pattern”, es posible explorar los datos del índice en la pestaña Discover. Por defecto se muestran los datos de los últimos 15 minutos, pudiendo modificar el rango de tiempo mediante el filtro de tiempo:

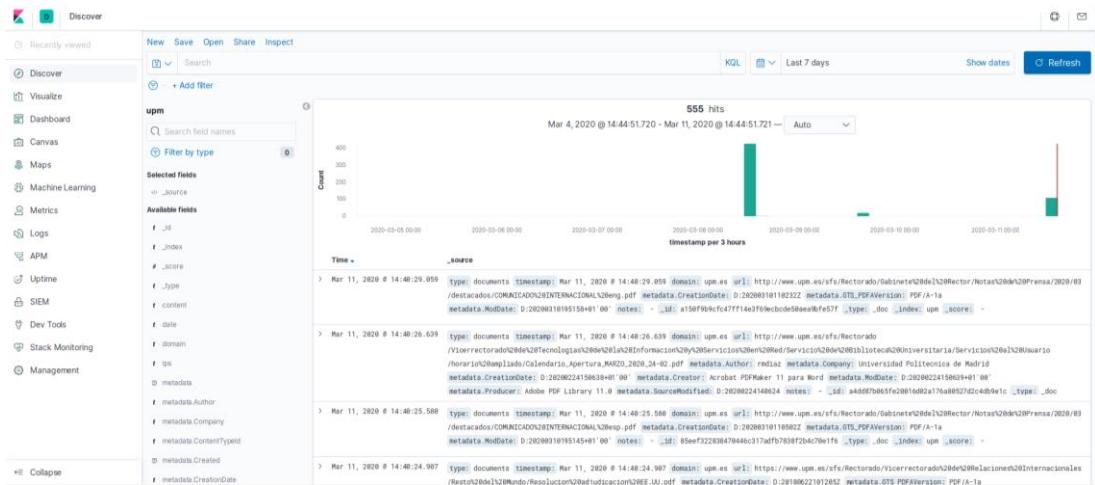


Ilustración 207 - Pestaña Discover de Kibana

Para visualizar los datos en distintos gráficos, Kibana tiene tanto su propia herramienta denominada Dashboard como Canvas. En nuestro caso nos hemos decantado por la segunda opción, ya que te permite hacer un diseño más flexible y creativo.

De esta manera, con Canvas se puede crear un “workpad” personalizado:

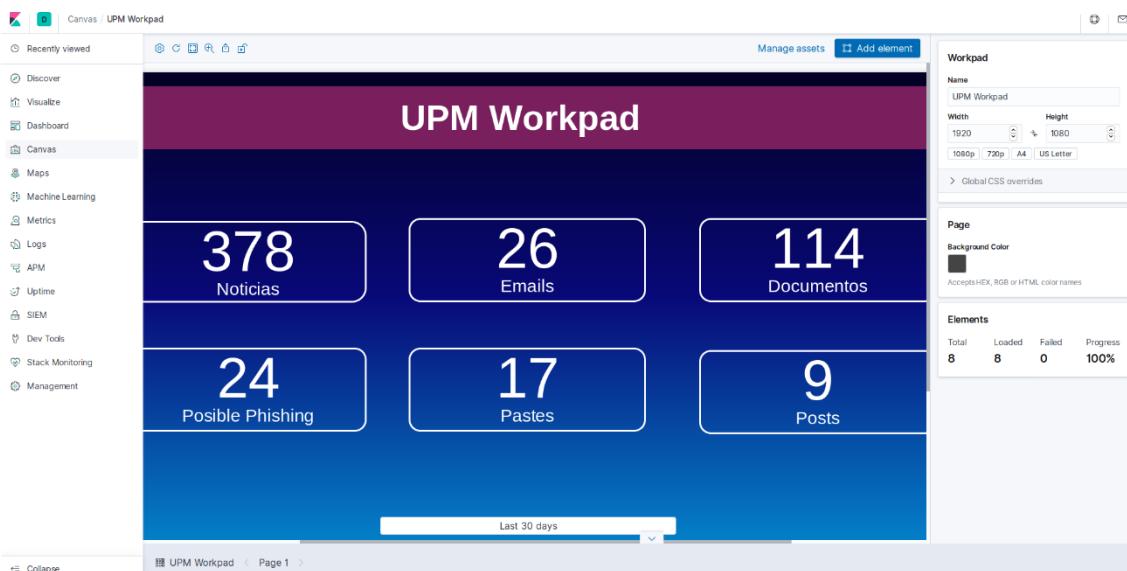


Ilustración 208 - Workpad realizado con Canvas

Una vez se han añadido los elementos deseados, como en nuestro caso que se trata de métricas (un número con una etiqueta), es necesario cambiar la fuente de los datos (data source) en la pestaña “Data”. Para ello, se selecciona la opción “Elastic raw documents” y después se indica tanto el índice del cual obtener los documentos como la consulta a realizar:

The screenshot shows a Kibana Workpad dashboard titled "UPM Workpad". The dashboard features six cards displaying metrics: "378 Noticias", "26 Emails", "114 Documentos", "24 Possible Phishing", "17 Pastes", and "9 Posts". Below the cards is a time range selector set to "Last 30 days". On the right side, the "Selected element" panel is open under the "Display" tab. In this panel, the "Change your data source" section is active, showing a warning about using the "Elasticsearch Doc" datasource for raw documents. It also displays the index name "upm" and a query type "news" in Lucene query string syntax. The "Sort Field" and "Sort Order" dropdowns are set to their default values. The "Fields" section is collapsed. At the bottom of the panel are "Preview" and "Save" buttons.

Ilustración 209 - Especificación de los datos a extraer de Elasticsearch para representar en el Workpad

Tras guardar pulsando en “Save”, en la pestaña “Display” hay que indicar la función y la clave del documento utilizada para extraer el valor:

This screenshot shows the same Kibana Workpad interface as the previous one, but the "Selected element" panel is now focused on the "Measure" section for the "Noticias" metric. The "Measure" section includes a "Number" field set to "Unique" and "title" as the column or function to extract the value. The "Metric" section shows the label "Noticias" and its description "Describes the metric". Under "Label text settings", the font size is set to "30" and "Open Sans". Under "Metric text settings", the font size is set to "96" and "Open Sans". The "Elementstyle" section contains "Container style" and "CSS" settings. The "Expression editor" button is visible at the bottom right of the panel.

Ilustración 210 - Especificación de los datos a mostrar en el Workpad

7

Conclusiones

Este proyecto, que hemos llevado a cabo para nuestra empresa, ha supuesto una mejora considerable en el servicio prestado a los clientes. Asimismo, nos ha permitido poner en práctica y ampliar los conocimientos adquiridos durante la carrera, incluyendo desde planificación de proyectos, bases de datos NoSQL y reingeniería de datos, hasta scripting con Python y web scraping.

En cuanto a la planificación del proyecto, aunque el esfuerzo estimado en horas por cada tarea no ha sido del todo desacertado, las fechas previstas de comienzo y finalización en algunos casos no se han cumplido, lo que ha provocado una demora en la realización de las tareas posteriores y por tanto del proyecto en su conjunto. Esto se debe a que no siempre se ha dedicado el tiempo previsto por cada día, principalmente porque contamos con más horas de las que finalmente hemos podido dedicar al proyecto en nuestra empresa.

Con los módulos desarrollados en Python para el framework Recon-ng, se ha reducido notablemente el tiempo de adquisición de la información que anteriormente se obtenía de manera manual, utilizando para ello la técnica de web scraping. No obstante, presenta ciertas dificultades, como la restricción de las peticiones que se pueden realizar a las páginas de las cuales se obtienen los datos, o que haya algún cambio en la estructura de estas páginas.

Anteriormente la información recopilada se reportaba directamente en informes técnicos de Excel, pues en nuestra empresa no se disponía de una base de datos donde almacenarla. La integración de Elasticsearch con Recon-ng ha permitido almacenar los datos obtenidos con los módulos, trayendo consigo todas las ventajas que una base de datos NoSQL proporciona: esquema flexible, escalabilidad horizontal, velocidad de búsqueda, etc. Por su parte, Kibana ha facilitado la tarea de análisis de los datos indexados en Elasticsearch al mostrarlos en tiempo real y permitir realizar búsquedas.

Además, el dashboard realizado en Kibana proporciona un feedback continuo mediante la representación de los datos en gráficas.

Por otro lado, el módulo desarrollado para generar reportes en Excel con el formato corporativo también ha supuesto una mejora significativa en la realización de tareas ofimáticas, ya que se evita tener que incluir manualmente en Excel los datos adquiridos y tener que comprobar que no hayan sido añadidos con anterioridad.

Adicionalmente, el hecho de generar el reporte en Excel automáticamente facilita la representación estructurada de los datos, lo cual puede no ser factible cuando se realiza de manera manual. También es muy útil cuando se requiere hacer una demo para un cliente potencial, ya que se reduce el tiempo necesario para su realización, permitiendo así dedicarlo a otras tareas más prioritarias.

Por último, cabe destacar que gracias al trabajo en equipo este proyecto nos ha supuesto un crecimiento tanto personal como profesional, siendo esta una de las cualidades que más se valoran en el mundo laboral.

8 Impacto social y legal

Vivimos en la denominada sociedad de la información, donde la mayoría de las personas hacen uso de servicios de Internet, muchas veces indispensables para su trabajo o su vida cotidiana. Sin embargo, en ocasiones se publican datos inconscientemente que pueden ser aprovechados por atacantes de manera maliciosa.

Esta información es de fácil acceso, lo que la hace verdaderamente peligrosa. Sin embargo, es posible prevenir ciberataques detectando las fugas de información antes de que sean utilizadas con fines maliciosos.

Con este proyecto se pretende obtener información sensible de personas u organismos presente en Internet, para así reportarla a quien corresponda y que se puedan tomar medidas a tiempo, ya que es la primera vía que utilizan los atacantes para estafar o extorsionar a las víctimas y comprometer sus sistemas.

Además, que toda esta información esté expuesta públicamente no implica que su uso sea legal. Según el inciso 2º del artículo 197 del Código Penal, se considera delito acceder, apoderarse, utilizar o modificar los datos de carácter personal de otro que se encuentren en cualquier registro público o privado, sin estar autorizado y en perjuicio de un tercero. Es decir, no supondría un delito utilizar los datos recopilados siempre que sea con autorización y en beneficio del titular de estos, como en el caso en el que un cliente lo solicite con la finalidad de reportárselo.

9

Bibliografía

- [1] I. Portillo, «ginseg,» diciembre 2018. [En línea]. Available: <https://ginseg.com/2018/897/inteligencia/introduccion-a-la-inteligencia/>. [Último acceso: 2020].
- [2] Association of Former Intelligence Officers, «Intelligence Collection, Covert Operations, and International Law,» *From AFIO's The Intelligencer Journal of U.S. Intelligence Studies*, vol. 23, nº 1, p. 6, 2017.
- [3] A. Martínez, «OSINT - La información es poder,» mayo 2014. [En línea]. Available: <https://www.incibe-cert.es/blog/osint-la-informacion-es-poder>. [Último acceso: 2020].
- [4] Y. Rodríguez, «INTELIGENCIA DE FUENTES ABIERTAS (OSINT): CARACTERÍSTICAS, DEBILIDADES Y ENGAÑO,» noviembre 2019. [En línea]. Available: <https://www.seguridadinternacional.es/?q=es/content/inteligencia-de-fuentes-abiertas-osint-caracter%C3%ADsticas-debilidades-y-engao%C3%B1o>. [Último acceso: 2020].
- [5] «Intelligence Studies: Types of Intelligence Collection,» [En línea]. Available: <https://usnwc.libguides.com/c.php?g=494120&p=3381426>. [Último acceso: 2020].
- [6] H. J. Williams y I. Blum, «Defining Second Generation Open Source Intelligence (OSINT) for the Defense Enterprise,» RAND Corporation, 2018.
- [7] Association of Former Intelligence Officers, «The Evolution of Open Source Intelligence (OSINT),» *The Intelligencer Journal of U.S. Intelligence Studies*, vol. 19, nº 3, 2013.
- [8] M. Glassman y M. JuKangb, «Intelligence in the internet age: The emergence and evolution of Open Source Intelligence (OSINT),» *Computers in Human Behavior*, vol. 28, nº 2, 2012.
- [9] P. N. F. G. M. G. M. P. Javier Pastor-Galindo, «The Not Yet Exploited Goldmine of OSINT: Opportunities, Open Challenges and Future Trends,» *IEEE*, 2020.
- [10] «FOCA,» 2017. [En línea]. Available: <https://www.elevenpaths.com/es/labstools/foca-2/index.html>. [Último acceso: 2020].

- [11] N. Shah, D. Willick y V. Mago, «A framework for social media data analytics using Elasticsearch and Kibana,» *Wireless Networks* , 2018.
- [12] «¿Qué es elasticsearch? | Elastic,» [En línea]. Available: <https://www.elastic.co/es/what-is/elasticsearch>. [Último acceso: 2020].
- [13] Z. Parker, Z. Parker, S. Poe, S. Poe, S. V. Vrbsky y S. V. Vrbsky, «Comparing NoSQL MongoDB to an SQL DB,» de *Proceedings of the 51st ACM Southeast Conference*, 2013.