

Exposición de datos personales en Internet

Guillermo Villegas Martín

Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones

- Análisis de datos -

Carlos Hernández Gañán

Helena Rifà Pous

02/06/2020



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

C) Copyright

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Exposición de datos personales en Internet</i>
Nombre del autor:	<i>Guillermo Villegas Martín</i>
Nombre del consultor/a:	<i>Carlos Hernández Gañán</i>
Nombre del PRA:	<i>Helena Rifà Pous</i>
Fecha de entrega (mm/aaaa):	<i>06/2020</i>
Titulación:	<i>Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones</i>
Área del Trabajo Final:	<i>Análisis de datos</i>
Idioma del trabajo:	<i>Español</i>
Palabras clave	<i>Datos, exposición, perfilado, privacidad</i>
<p>Resumen del Trabajo (máximo 250 palabras): Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</p> <p>La ignorancia es el peor de todos los males – Platón –</p> <p>En el actual mundo hiperconectado donde la vida digital se entremezcla con la real afectándose mutuamente, y la explotación de nuestros datos personales se ha convertido en uno de los mayores mercados. Es primordial conocer los datos que se exponen en cada servicio, para ser conscientes y poder aplicar las medidas oportunas para su protección.</p> <p>Actualmente existen servicios que analizan los términos y políticas de privacidad de otros servicios y permiten al usuario conocer de una forma más o menos detallada el uso y exposición de los datos que estos recopilan. Pero no son capaces de llegar a construir este perfil de identidad digital que le dé al usuario la visión de por donde se mueven sus datos.</p> <p>Haciendo uso de una metodología Agile se ha desarrollado durante este trabajo un nuevo servicio web, que permita a los usuarios construir este perfil digital y tener una visión de dónde exponen qué dato.</p> <p>Se ha conseguido realizar una versión funcional, que permite al usuario conformar su perfil de datos personales expuestos, analizar y comparar diferentes servicios en el uso que hacen de los datos personales, así como tener una visión estadística general de los datos personales en internet.</p>	

Abstract (in English, 250 words or less):

Ignorance, the root and stem of all evil – Plato –

We live in a hyperconnected world where the digital and real life's affects each other, and the personal data is the new economic unicorn. In order to be able to protect their life, the users need to know as much as possible of the data they are sharing and where they share it.

Exists different services where you can review the privacy terms and policies some services are applying, but these services are unable to create a holistic profile of the user exposed personal data.

During this TFM and using the Agile methodology, I have created a new web Service where the users can fill their profiles, conforming a global image of their digital exposed identity.

Índice

1	Introducción.....	7
1.1	Contexto y justificación del Trabajo.....	7
1.2	Objetivos del Trabajo.....	8
1.3	Enfoque y método seguido.....	8
1.4	Planificación del Trabajo.....	9
1.5	Breve resumen de productos obtenidos.....	9
1.6	Breve descripción de los otros capítulos de la memoria.....	10
2	Estado del arte.....	11
3	Diseño Funcional.....	14
3.1	Marco de Datos.....	14
3.1.1	Confidencialidad.....	14
3.1.2	Exposición.....	15
3.1.3	Otras características.....	15
3.2	Modelo de Datos.....	16
3.3	Funcionalidades.....	16
3.3.1	Funcionalidades básicas.....	17
3.3.2	Funcionalidades Extendidas.....	17
3.4	Experiencia de usuario.....	18
4	Implementación.....	21
4.1	Selección del lenguaje de programación.....	21
4.2	Selección Framework de desarrollo.....	22
4.3	Funcionamiento y estructura del framework Django.....	23
4.3.1	Estructura de ficheros de un proyecto Django.....	24
4.3.2	Repositorio de código.....	25
4.4	Implementación del modelo de datos.....	26
4.5	Permisos de los usuarios.....	27
4.6	Enrutador de la aplicación.....	28
4.7	Desarrollo operaciones con Datos.....	28
4.8	Desarrollo operaciones con Sitios.....	30
4.9	Desarrollo operaciones con Corporaciones.....	33
4.10	Desarrollo operaciones del perfil y usuarios.....	34
4.11	Desarrollo del dashboard.....	35
4.12	Desarrollo del buscador.....	39
4.13	Desarrollo del Scrapper web.....	40
4.14	Desarrollo del Menú.....	42
5	Limitaciones y trabajo futuro.....	44
5.1	Limitaciones de la implementación realizada.....	44
5.1.1	Gestión de los roles de los usuarios.....	44
5.1.2	Sinónimos y traducciones de los datos.....	44
5.1.3	Limitaciones por idioma en el scrapper.....	45
5.1.4	Localización de datos.....	45
5.1.5	UX.....	46
5.1.6	Categoría de datos personales.....	46
5.1.7	Formularios de Corporaciones.....	46

5.1.8	Edición datos del scrapper	46
5.2	Trabajo futuro	46
5.2.1	Scrapper Autónomo	47
5.2.2	Scrapper Ampliado.....	47
5.2.3	Perfil detallado	47
5.2.4	Scrapper de perfil.....	47
5.2.5	Sistema de alertas.....	47
5.2.6	Dashboar interactivo	48
5.2.7	Buscador vitaminado.....	48
6	Conclusiones.....	49
7	Glosario	50
8	Bibliografía	51
9	Anexos	52
9.1	URLs de la aplicación	52

Lista de tablas

Tabla 1 - Escala de niveles de Confidencialidad	15
Tabla 2 - Escala de niveles de Exposición	15

Lista de figuras

Ilustración 1 - Infografía datos gestionados en las grandes compañías	11
Ilustración 2 - Captura servicio ToSDR	12
Ilustración 3 - Captura del servicio Guard	12
Ilustración 4 - Modelo de datos	16
Ilustración 5 - Mapa web	18
Ilustración 6 - Wireframe página inicio	19
Ilustración 7 - Wireframe página detalle sitio	19
Ilustración 8 - Wireframe página perfil	20
Ilustración 9 - Wireframe página dashboard	20
Ilustración 10 - Lenguajes de programación más populares	22
Ilustración 11 - Tabla comparativa Frameworks Web Python	22
Ilustración 12 - Arquitectura MVC (Foto de la Wikipedia)	23
Ilustración 13 - Estructura del proyecto Django de este TFM	24
Ilustración 14 - Estructura de la aplicación de este TFM	25
Ilustración 15 – Captura interfaz ExpID: Detalle Dato Personal	29
Ilustración 16 - Captura interfaz ExpID: Formulario Dato Personal	30
Ilustración 17 - Captura interfaz ExpID: Lista de Datos Personales	30
Ilustración 18 - Captura Interfaz ExpID: Detalles de un Sitio	31
Ilustración 19 - Captura interfaz expID: Formulario de Sitio	32
Ilustración 20 - Captura Interfaz expID: Lista de Sitios	32
Ilustración 21 - Código parcial: Función save del modelo Site	33
Ilustración 22 - Captura Interfaz expID: Detalle de una Corporación	33
Ilustración 23 - Captura Interfaz expID: Formulario Corporación	33
Ilustración 24 - Captura Interfaz expID: Lista de Corporaciones	34
Ilustración 25 - Código parcial: Función save del modelo Corporación	34
Ilustración 26 - Captura Interfaz expID: Página Perfil usuario	35
Ilustración 27 - Captura Interfaz expID: Formulario de registro de usuario	35
Ilustración 28 - Dashboard: gráfica de Datos por categoría	36
Ilustración 29 - Dashboard: gráfica “Recogida de datos”	37
Ilustración 30 - Dashboard: gráfica “Top 10 Datos”	37
Ilustración 31 - Dashboard: gráfica “Resumen de Sitios”	37
Ilustración 32 - Dashboard: tabla “TOP Sitio por tipo de Dato”	38
Ilustración 33 - Dashboard: gráfica “Mi perfil vs la media”	38
Ilustración 34 – Código parcial: Función vista del Dashboard	39
Ilustración 35 - Código parcial: Template Dashboard	39
Ilustración 36 - Captura Interfz expID: Página del buscador	40
Ilustración 37 - Captura Interfz expID: Página de resultados del buscador	40
Ilustración 38 - Captura Interfaz expID: Scrapper (Analizador de Sitios)	41
Ilustración 39 - Código parcial: Análisis de sitios (scraper web)	42
Ilustración 40 - Captura Interfaz expID: Menú	43
Ilustración 41 - Código: enrutador urls de expID	52

1 Introducción

1.1 Contexto y justificación del Trabajo

Desde el origen de los tiempos, la recopilación de datos del mundo en el que vivimos nos facilita la comprensión de este. Con cada adelanto tecnológico estas tareas de recolección, almacenamiento y análisis de los datos se vuelven cada vez más sencillas, potentes y económicas.

En la actualidad, la recolección de datos personales es una tarea que está presente en mayor o menor grado en todas las compañías, y servicios, para prestar su servicio directamente o para mejorar sus campañas de marketing. Quedando a criterio del propio servicio la cantidad de datos personales que se recopilan y el posterior tratamiento o uso de estos.

Para regular estas capacidades de obtención de datos personales y su uso, se están llevando a cabo por diferentes organismos gubernamentales figuras legislativas para limitar y proteger la privacidad de los individuos, limitando la recolección y tratamiento de estos datos personales. El mayor ejemplo lo tenemos en el GDPR, *General Data Protection Regulation*, o en castellano, el Reglamento General de Protección de datos. Una regulación a nivel europeo, aprobada en 2016 y aplicada en 2018, que protege los datos personales y derechos de todo ciudadano de un país europeo, aunque la empresa que realiza la recolección y tratamiento no sea europea.

A pesar de las medidas legislativas que se están llevando a cabo por los diferentes gobiernos para proteger la privacidad de sus ciudadanos, los servicios y empresas siguen recopilando y tratando, dentro de la legalidad, los datos de sus usuarios ya que es un mercado muy suculento. Sólo en EE. UU. este mercado de datos de usuarios mueve 18.000 millones de dólares. [1]

Los usuarios, aunque más protegidos, siguen teniendo un gran desconocimiento de qué datos personales están siendo recopilados por cada una de las páginas, servicios o empresas que utilizan. Bajo el marco del GDPR, los ciudadanos europeos tienen el derecho de saber qué datos están siendo recopilados, pero tienen que ir solicitándolos uno a uno en los diferentes sitios/servicios o empresas en los que se han registrado, y eso sin tener en cuenta las páginas por las que pasen que registren datos sin necesidad de estar registrado.

Una vez el usuario haya recabado todos los datos personales que iba acumulando cada sitio, tiene que invertir tiempo en analizarlos y extraer conclusiones de todo lo que está exponiendo en internet.

El objetivo de este proyecto es generar un marco común sobre la exposición de los datos personales de los usuarios. Este marco común

permitirá clasificar los datos personales de una forma uniforme para todos los usuarios, y que los usuarios puedan conformar un perfil de todas las aplicaciones, servicios o compañías que han utilizado, sean usuarios registrados o no. Al estar utilizando un marco común, será posible también tener una referencia de tus datos personales con relación al resto de usuarios, sabiendo si estás más o menos expuesto en internet.

1.2 Objetivos del Trabajo

- Creación de un marco común para la clasificación de los datos
- Desarrollo de un servicio, Exposed ID, en el que poder consultar los datos que recopilan los diferentes sitios
- Desarrollo de un servicio por el que gestionar un perfil de tus datos expuestos en internet y quién los recopila
- Desarrollo de un dashboard de estadísticas de los datos expuestos en internet

1.3 Enfoque y método seguido

El trabajo se puede dividir en dos puntos principales a desarrollar.

1. Definición del marco común de clasificación de datos
2. Implementación del servicio de consulta/gestión de dicho marco

Para desarrollar el primero de los puntos, he seguido un enfoque continuista y conciliador. Me he basado en las leyes y estándares ya definidos y utilizados en la actualidad para generar la definición con la que construir el marco para tratar los datos personales.

Para realizar el segundo de los puntos he podido escoger entre diferentes estrategias a la hora de implementar el servicio a los usuarios, tanto en la forma de presentar el servicio a los usuarios:

- Creación de una aplicación móvil
- Creación de una página web

Cómo en la forma de funcionamiento del servicio:

- Servicio colaborativo
- Servicio cerrado

El enfoque que he decidido utilizar es el de desarrollar una página web con un funcionamiento colaborativo. He optado por una página web porque un solo desarrollo permite el acceso de todos los usuarios potenciales, ya que las páginas web también pueden ser consumidas desde un móvil, y una aplicación nativa para estos habría tenido o que adaptarla a todos los S.O. o haberme centrado en uno concreto.

El modelo de funcionamiento colaborativo va a permitir que el servicio crezca y se adapte a los cambios de los servicios de una forma mucho más rápida. Esta decisión está inspirada en el modelo de la Wikipedia,

donde son los propios usuarios los que completan y actualizan la información contenida, permitiendo que el sistema siga siendo útil al resto de usuarios sin requerir un gran esfuerzo de mantenimiento por parte del creador del servicio.

1.4 Planificación del Trabajo

El trabajo se ha organizado en Sprints, siguiendo los conceptos de las metodologías Agile y de Mínimo producto viable.

La meta que persigo es tener un servicio que demuestre los objetivos del TFM desde el primer entregable, iterando para evolucionarlo e irle añadiendo funcionalidades. El número de entregas o fases ha estado marcado por los tiempos y plazos del proyecto, así como por el ritmo de los desarrollos realizados.

Entrega 1:

- Documento de Plan de Trabajo

Entrega 2:

- Sprint 1 (2 semanas): Diseño funcional de la aplicación y casos de uso
- Sprint 2 (1 semana): Diseño del modelo de datos
- Sprint 3 (1 semana): Selección del framework e instalación

Entrega 3:

- Sprint 4 (1 semana): Implementación de la BBDD
- Sprint 5 (1 semana): Desarrollo de las operativas CRUD para los sites y datos personales
- Sprint 6 (1 semana): Desarrollo de las operativas CRUD para los usuarios y su perfil
- Sprint 7 (1 semana): Desarrollo del Dashboard de estadísticas y buscador

Entrega 4:

- Sprint 8 (1 semana): Desarrollo del web-scrapper
- Sprint 9 (1 semana): Poblado de datos en la aplicación
- Sprint 10 (2 semanas): Redacción final de la memoria y presentación del TFM

1.5 Breve resumen de productos obtenidos

El producto obtenido es un servicio web, con el que el usuario va a poder consultar, modificar, añadir o eliminar las fichas en Exposed ID de los datos que recopilan los diferentes sitios, servicios o compañías, así como ver las estadísticas de estos datos, y generarse su propio perfil con el que saber cuántos datos personales están exponiendo en internet y dónde.

El resultado está accesible en <http://expid.herokuapp.com> y el código fuente se puede encontrar en <https://github.com/qiyevm/expID>

1.6 Breve descripción de los otros capítulos de la memoria

- **Estado del arte:** es un breve capítulo en el que se analiza el estado actual del mapeo de datos personales en internet.
- **Diseño Funcional:** en este capítulo se repasan las principales decisiones de diseño del servicio.
- **Implementación:** en los diferentes apartados de este capítulo se detallan las decisiones técnicas realizadas para poder implementar las funcionalidades diseñadas.
- **Limitaciones y trabajo futuro:** este capítulo recoge las carencias y limitaciones que presenta el producto desarrollado durante el tiempo del trabajo final de máster, y plantea mejoras futuras que aportarán nuevas funcionalidades a los usuarios.
- **Conclusiones:** en este capítulo se enumeran las conclusiones del trabajo realizado.

2 Estado del arte

Después de ciertos incidentes de seguridad en los que se descubrió la gran cantidad de datos personales que son recopilados por las aplicaciones, y sobre todo desde la aplicación del reglamento europeo sobre protección de datos, RGPD [2] se ha generado mucho revuelo con los datos personales y el tratamiento que las empresas hacen de estos.

Para cumplir la legislación europea, todos los servicios que quieran tener entre sus usuarios a ciudadanos europeos tienen que facilitar el ejercicio de sus derechos a los usuarios, y para ello deben como mínimo poder recopilar y entregar todos los datos que recolectan, aparte de disponer de mecanismos para modificar dichos datos o eliminarlos a petición del usuario.

Y a raíz de tener disponible esta posibilidad de consulta de datos, se ha ido generando en la red un gran volumen de contenido, artículos, infografías o aplicaciones que permiten a los usuarios conocer qué datos recopilan algunos servicios.

Security Baron realizó una infografía en la que mostraba una lista bastante extensa de datos recopilados por 6 de las más grandes compañías de tecnología: Apple, Facebook, Twitter, Google, Amazon y Microsoft [4], como se ve en la imagen 1.

Esta infografía se basa en el análisis de los términos y condiciones de cada uno de los servicios de estas compañías.

Otra alternativa que le proporciona al usuario información del tratamiento de sus datos a parte de otra información relevante sobre los términos y condiciones del servicio es ToSDR (Terms of Service, Didn't Read) [5].

En esta página se analizan los términos y condiciones de los servicios para extraer una clasificación de si el servicio tiene unas condiciones más favorables para los usuarios, o protegen y van a favor de la compañía.

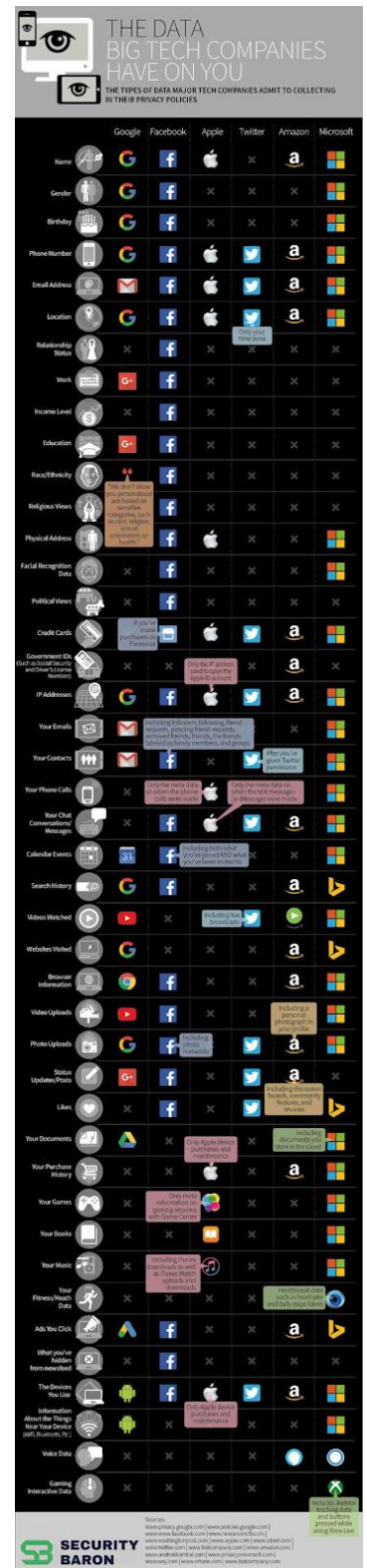


Ilustración 1 - Infografía datos gestionados en las grandes compañías

Google Class C

- This service may collect, use, and share location data
 - The service can read your private messages
 - You agree to defend, indemnify, and hold the service harmless in case of a claim related to your use of the service
 - This service tracks you on other websites
 - Limited copyright license to operate and improve all Google Services
- [More details](#)

YouTube Class D

- Terms may be changed any time at their discretion, without notice to the user
 - Processes a personal information (email, id but also device info, location)
 - Users should revisit the terms periodically, although in case of material changes, the service will notify
 - If you are the target of a copyright claim, your content may be removed
 - The service is not responsible for linked or (clearly) quoted content from third-party content providers
- [More details](#)

Amazon Class C

- Terms may be changed any time at their discretion, without notice to the user
 - The service can delete your account without prior notice and without a reason
 - This service tracks you on other websites
 - This service forces users into binding arbitration in the case of disputes
 - Blocking cookies may limit your ability to use the service
- [More details](#)

twitter Class D

- Very broad copyright license on your content
 - Third party cookies
 - This service ignores the Do Not Track (DNT) header and tracks users anyway even if they set this header.
 - The service can delete your account without prior notice and without a reason
 - This service reserves the right to disclose your personal information without notifying you
- [More details](#)

Ilustración 2 - Captura servicio ToSDR

En la misma línea que ToSDR, hay otro servicio llamado Guard [6] que, por medio de una IA, analiza los términos y condiciones para extraer ciertas conclusiones indicando qué partes son más peligrosas para la privacidad del usuario como problemas potenciales en la aplicación. Esta herramienta aparte de hacer un análisis del texto de las condiciones del servicio también tiene en cuenta posibles escándalos que haya tenido la aplicación a la hora de generar la calificación del sitio.

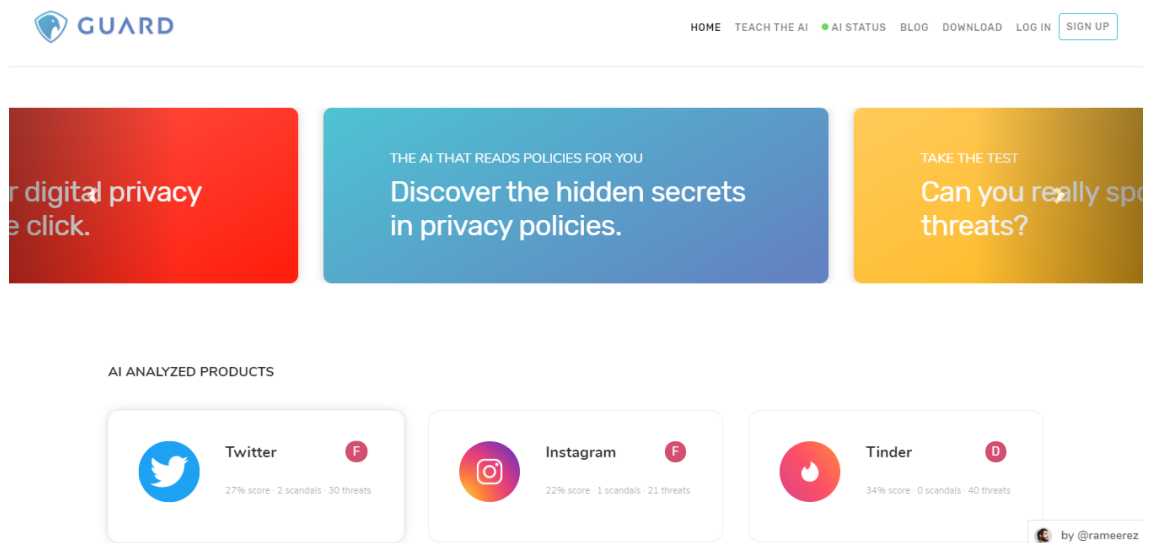


Ilustración 3 - Captura del servicio Guard

La forma de entrenar a la IA del sistema es de una forma colaborativa por parte de los propios usuarios.

Los usuarios también tienen unas aplicaciones para móvil que permiten analizar qué datos están recopilando y a donde los están mandando, el resto de las aplicaciones instaladas en el terminal [7].

Dos ejemplos de estas aplicaciones de monitorización son:

- **AppCensus** [8], es una aplicación que analiza el comportamiento real que tienen las aplicaciones en el móvil donde está instalada. Para ello requiere que se le concedan todos los permisos a la aplicación a analizar, y se haga uso de ella mientras *AppCensus* monitoriza los datos que recopila y envía.
- **Exodus Privacy** [9]. Esta aplicación analiza el comportamiento de las aplicaciones inspeccionando los permisos que solicita en el terminal y buscando los rastreadores de terceros que puedan haberse codificado en la aplicación.

En todas estas alternativas sigue presente la carencia principal que intento resolver con mi proyecto. No se le da al usuario una visión completa de qué datos son los que está el usuario compartiendo en la totalidad de su vida digital. Entre estas alternativas hay algunas centradas en páginas web, otras dedicadas a las aplicaciones, y casi todas te dan una visión general de las características del servicio sin especificar los datos personales concretos que recogen.

Tampoco, ninguna de las alternativas presenta un dashboard comparativo con el que tener una visión general de la exposición de los datos personales de los usuarios en internet.

3 Diseño Funcional

3.1 Marco de Datos

Como se ha visto anteriormente, existen diversas alternativas en el mercado que clasifican las aplicaciones o servicios en función de unos criterios más o menos conocidos, mostrando una cualificación general de estos análisis sin saber cómo se realiza dicho cálculo.

Uno de los objetivos de este TFM es construir un marco de datos, con los que poder clasificar de la forma más objetiva posible los datos personales y su recolección por parte de los servicios y aplicaciones. Con este marco el usuario podrá tener una idea clara y concisa desde su punto de vista de los datos que tiene expuestos en cada sitio, y como es su recolección.

Considero que los datos tienen dos características fundamentales que los definen. Estas características principales más algunas otras que acaban de perfilar el dato son las que voy a estandarizar en mi marco de referencia para que los datos resultantes en mi servicio sean objetivos y útiles para los usuarios.

3.1.1 Confidencialidad

La confidencialidad de un dato es una de las dos características principales de un dato.

En este proyecto, se entiende por confidencialidad el nivel que se espera de confianza de que se mantendrá la reserva del dato.

Tomando como referencia los niveles de clasificación de datos, descritos en el CISP [10]. He adaptado los niveles de la clasificación tanto empresarial como militar. Con el objetivo de obtener una única escala de niveles.

La definición de cada nivel está enfocada al impacto que tendría la exposición de ese dato desde el punto de vista del usuario, teniendo en cuenta si es necesario para la realización de las funciones del servicio o aplicación. La definición también contempla las definiciones de dato personal que realiza el RGPD.

Nivel	Nombre	Descripción
1	Unclassified	Datos del dominio público
2	Protected	Controvertido, pero ampliamente difundido
3	Restricted	Necesario para el funcionamiento. No es muy peligroso si se hace público

4	Confidential	Necesario para transacciones oficiales/administrativas
5	Secret	Muy peligroso si se hace público, puede ser utilizado para hacer mucho daño
6	Top Secret	Identifica inequívocamente a una persona física por sí solo

Tabla 1 - Escala de niveles de Confidencialidad

3.1.2 Exposición

El concepto de exposición está definido como la peligrosidad de que el dato personal sea expuesto y otros usuarios tengan acceso al mismo.

No tiene el mismo nivel de exposición un apodo que un número de cuenta, o un DNI.

He considerado solamente 4 niveles de exposición, suficientes para poder clasificar todos los datos y dar suficiente granularidad a la variable.

Nivel	Nombre	Descripción
1	Basic	Fácilmente accesible, no se puede realizar gran cosa con el dato
2	Medium	Si se publica no supone prácticamente peligro para el afectado
3	High	Alto grado de producir problemas o ser usado delictivamente
4	Critical	Te expone completamente como persona física.

Tabla 2 - Escala de niveles de Exposición

3.1.3 Otras características

A parte de definir las características principales de los datos en los dos puntos anteriores, existe otro conjunto de características que van a afinar el uso de estos datos y aumentarán la información que se le proporcionará al usuario en el servicio.

- Categoría: todos los datos tendrán una categoría asignada, que agrupará los datos personales similares.

Existe un conjunto de características que no definen exactamente un dato personal, sino a cada relación de un sitio con un dato personal, ya que para el mismo dato personal estas características pueden ser diferentes según el sitio.

- Oculto (Shadow): esta característica indicará si el dato personal es recogido de forma oculta al usuario o este es consciente de que ha indicado el dato.
- Logado (logged): indicará si un dato es recopilado para todos los usuarios del sitio o solamente a los que se han registrado.
- Inferido (inferred): los datos pueden ser recogidos de forma directa, o que sea inferido o calculado por el sitio.

- Opcional (optional): en muchos sitios el usuario no está obligado a proveer todos los datos que se le solicitan para poder disfrutar del servicio.

3.2 Modelo de Datos

Una vez definido el marco de datos que voy a utilizar, he definido el modelo de datos que va a tener mi aplicación, y sobre el que construiré las funcionalidades para los usuarios.

He definido 4 tipos de datos para mi aplicación, los cuales están relacionados entre sí:

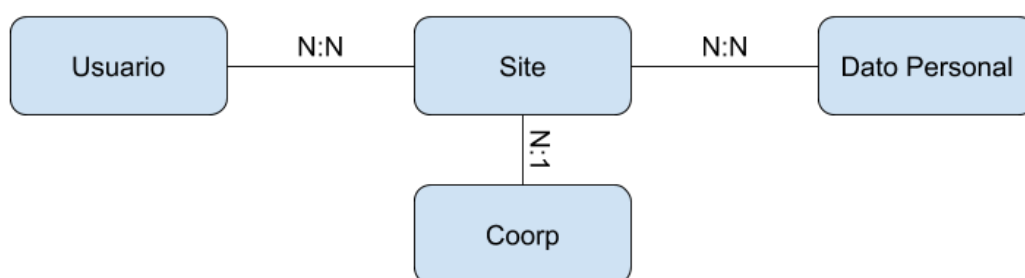


Ilustración 4 - Modelo de datos

Las características generales del modelo diagramado en la Ilustración 3 son:

- Reaprovechar las entradas del tipo dato personal, de forma que ocupen menos espacio en BBDD, y cualquier modificación en la caracterización de un dato concreto se actualice en todos los registros Site.
- Un Site podrá estar relacionado con múltiples datos personales, porque un site puede estar recopilando múltiples datos personales.
- Un usuario podrá estar relacionado con múltiples Sites, porque puede estar utilizando y tener cuenta en múltiples sitios.
- Todas estas relaciones N:N me permitirán también de una forma sencilla las búsquedas inversas, y saber para un Dato Personal concreto, que Sites lo recopilan, o cuantos usuarios tienen cuenta en un Site concreto.

3.3 Funcionalidades

Para que este proyecto tenga utilidad de cara a los usuarios se definen una serie de funcionalidades básicas necesarias y un conjunto de funcionalidades extendidas.

3.3.1 Funcionalidades básicas

- **Operación de consulta**
Es imprescindible que los usuarios puedan realizar consultas de los registros que tenga la aplicación para todos los tipos de datos: datos personales, sitios y corporaciones.
- **Operación de creación**
Bajo la premisa de ser un sitio colaborativo, es necesario que los usuarios sean capaces de crear nuevos registros para todos los datos. De esta forma las aportaciones de un usuario repercutirán en el resto de los usuarios del servicio.
- **Operación de modificación**
Dado que son los propios usuarios los que van poblando la BBDD, será necesaria la posibilidad de modificar los datos introducidos por otro usuario, ya sea por un error o para mejorarlos.
- **Operación de eliminación**
La operación de eliminación es fundamental en un sistema vivo donde pueden aparecer y desaparecer servicios.
- **Búsqueda**
Los usuarios tienen que ser capaces de poder buscar un dato, sitio o corporación dentro de la BBDD de una forma ágil y rápida.
- **Perfil**
Esta funcionalidad principal permite a un usuario ir gestionando un listado de los sitios que está utilizando y/o está registrado. De esta forma tendrá en un solo sitio toda la información sobre la exposición de sus datos en internet.
- **Estadísticas**
Se define esta funcionalidad como un conjunto de datos estadísticos sacados de la BBDD del servicio, que permitirán a los usuarios entender cómo se están gestionando los datos personales en internet, así como identificar los grandes y pequeños actores de este mercado.

3.3.2 Funcionalidades Extendidas

- **Incorporación automática de sitios**
Con el objetivo de ser un servicio atractivo para los usuarios y lo más sencillo posible, se plantea una funcionalidad que de forma automática recorra internet como una araña indexando los sitios y los datos asociados que sea posible.
- **Rastreo del perfil**
Con la misma premisa que la funcionalidad anterior, esta funcionalidad tiene por objetivo generar mecanismos que permitan automatizar y facilitar la elaboración de su perfil al usuario.
- **Notificaciones**
Disponer de un servicio de notificación, con diferentes alertas es un valor añadido a los usuarios.
- **Personalización**

Añadir personalización en las múltiples funcionalidades anteriores permitirá al usuario tener una experiencia más personal y obtener unos resultados más ajustados.

Teniendo en cuenta estas funcionalidades, se ha diseñado un mapa web del servicio que se está construyendo en este proyecto.

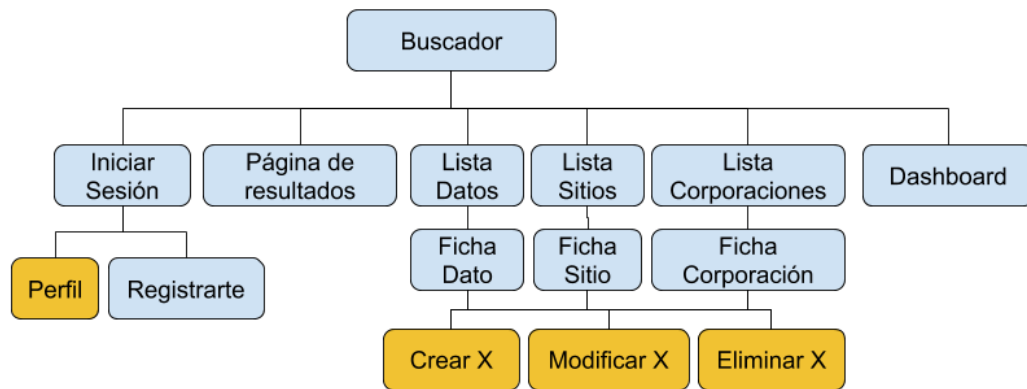


Ilustración 5 - Mapa web

En la ilustración anterior se ven todas las páginas accesibles públicamente en azul, mientras que las amarillas solamente estarán disponibles para usuarios registrados.

3.4 Experiencia de usuario

El principal objetivo en la experiencia de usuario es que esta sea lo más sencilla y fluida posible.

Los diseños de cada una de las páginas web por las que puede navegar el usuario es lo más simple, presentando únicamente los datos necesarios para la funcionalidad concreta.

Con el fin de tener claro el diseño, se han generado unos wireframes que permiten visualizar el resultado a implementar.

- Página de Inicio, el buscador. Esta página que será la que se encuentren los usuarios al acceder al servicio, será muy simple, siguiendo la filosofía de Google, donde únicamente se puede realizar una acción y hay una pequeña introducción al servicio.

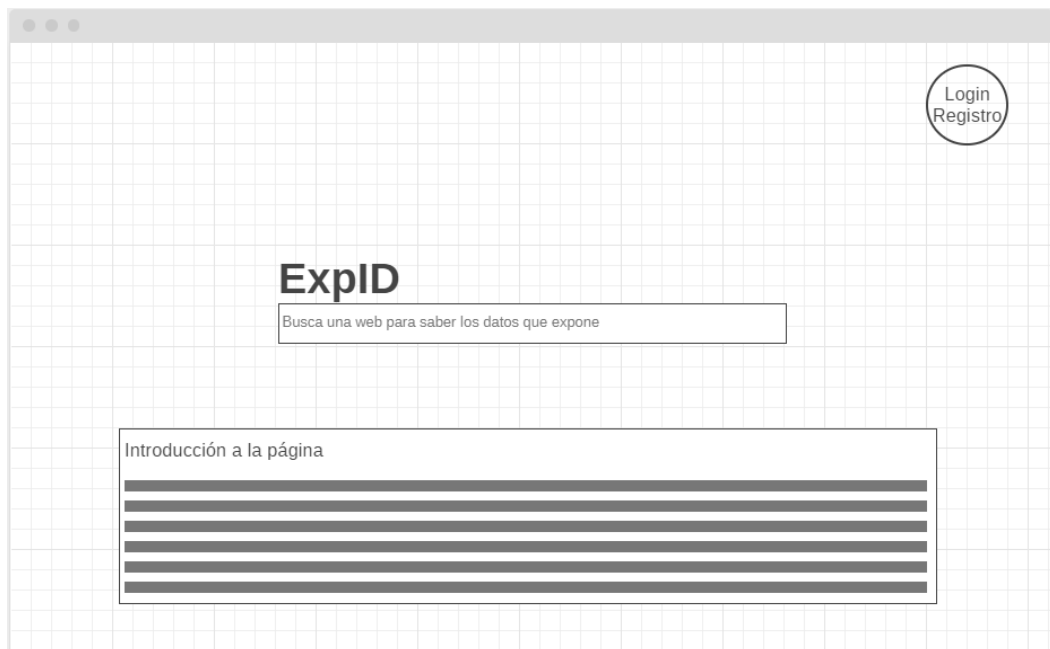


Ilustración 6 - Wireframe página inicio

- Página de un sitio. Esta es la representación en detalle de la ficha de un sitio. Donde se muestran sus características, así como los datos que recoge dicho sitio.

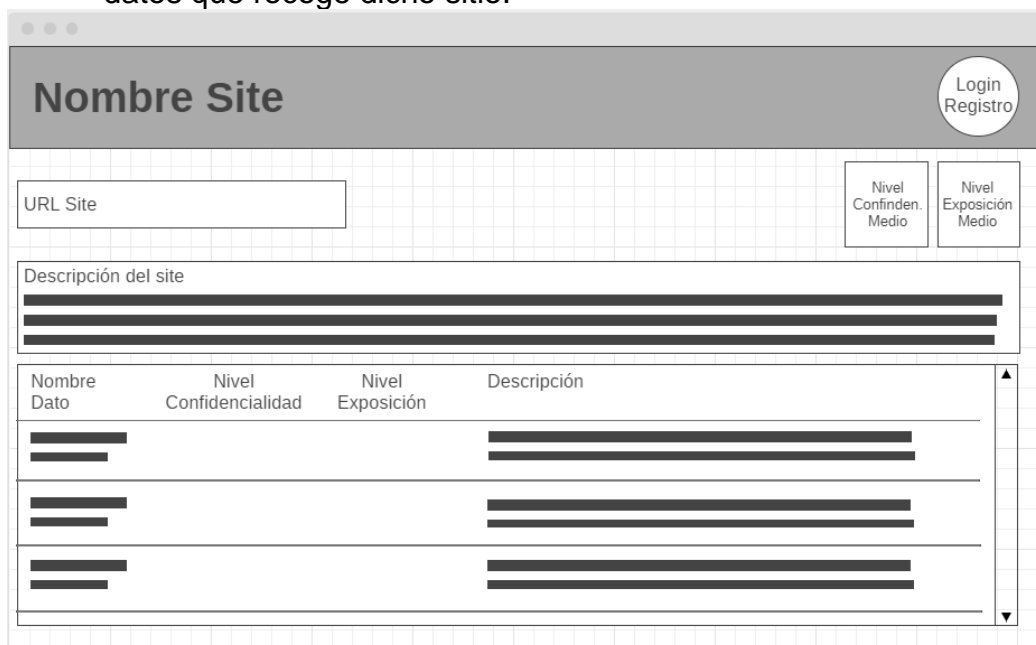


Ilustración 7 - Wireframe página detalle sitio

- Página perfil de usuario. Otra de las funcionalidades diferenciadoras de este proyecto es la opción de que los usuarios se creen un perfil que representa su huella en internet. El diseño esquemático de dicha página se muestra en la Ilustración 6.

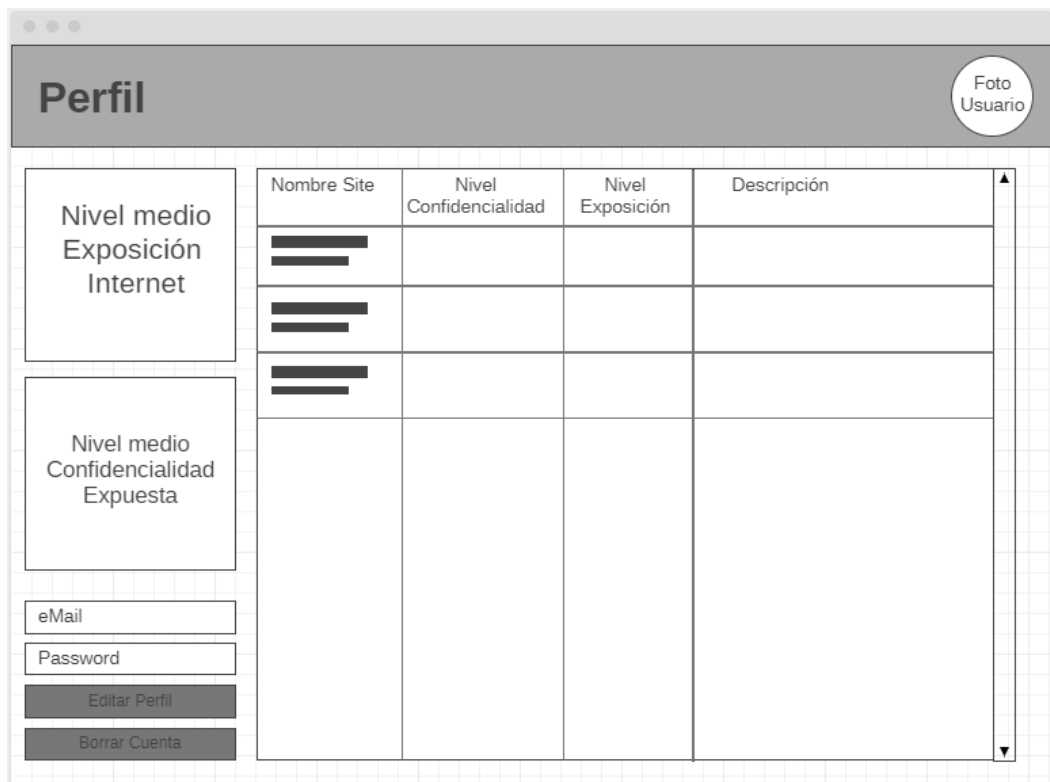


Ilustración 8 - Wireframe página perfil

- Página del dashboard. En esta página se van a mostrar las estadísticas de los datos de la aplicación, y del usuario autenticado. Es por tanto una de las páginas que más datos va a contener y ha de tener un diseño sencillo para que los usuarios puedan aprovechar al máximo los datos mostrados.

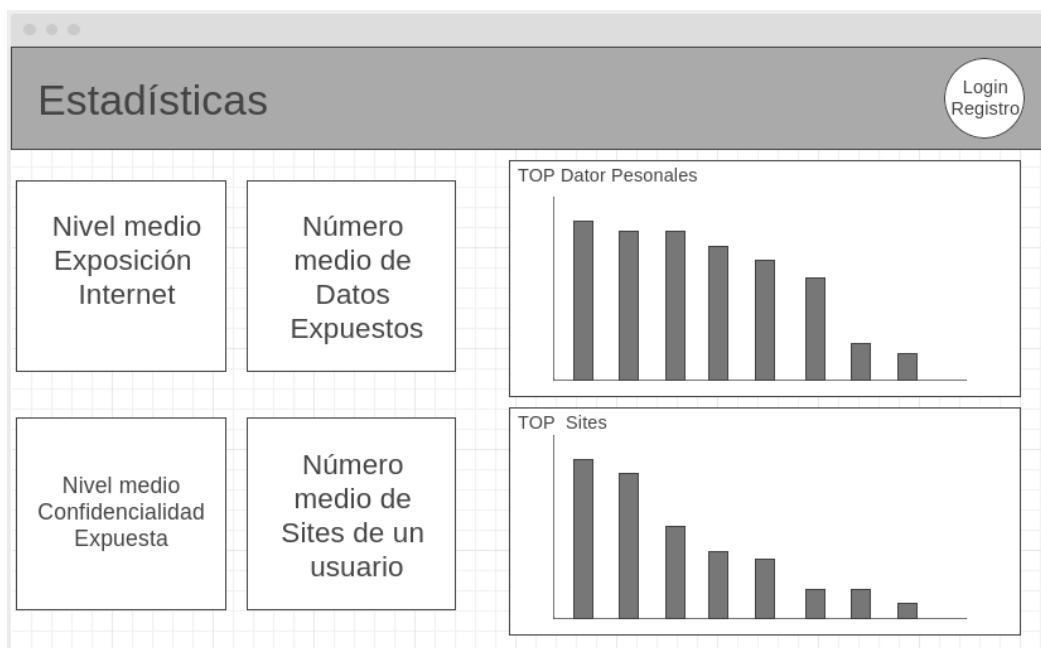


Ilustración 9 - Wireframe página dashboard

4 Implementación

4.1 Selección del lenguaje de programación

Para el correcto desarrollo de este proyecto es necesario hacer uso de lenguajes de programación que permitan el desarrollo web.

Los lenguajes de programación actuales se pueden clasificar por diversas características, y estas a su vez acotan la idoneidad del uso de estos lenguajes de programación.

La primera clasificación que podemos hacer son los lenguajes del lado del cliente, enfocados principalmente en la visualización e interfaz del usuario, y los lenguajes del lado del servidor, centrados en la lógica y los datos del servicio.

En este proyecto, dado que la web requiere lógica y gestión de datos, es necesario el uso de diversos lenguajes de programación, algunos del lado del cliente y alguno del lado del servidor.

Los lenguajes de programación más utilizados del lado del cliente y que se complementan entre sí son: css, html5 y javascript [11]. Tanto para css como javascript existen un conjunto de frameworks y librerías que aportan funcionalidad y velocidad de programación por disponer de funciones ya implementadas.

Entre los lenguajes del lado del servidor se abre un mayor debate de cuál es el mejor lenguaje. La elección de este lenguaje de programación suele estar condicionada por diversas variables, entre las que se encuentra las preferencias del programador. En la siguiente tabla, se observa la popularidad de los diferentes lenguajes según las estadísticas de 4 sitios de programación [12]:

RedMonk	Stack Overflow	SlashData	TIOBE Index 7/19
JavaScript	JavaScript	JavaScript	Java
Java	HTML/CSS	Python	C
Python	SQL	Java	Python
PHP	Python	C#	C++
Tie: C++/C#	Java	C/C++	C#
	Bash/Shell/Powershell	PHP	Visual Basic.NET
CSS	C#	Visual tools	JavaScript
Ruby	PHP	Swift	PHP
C	TypeScript	Ruby	SQL
TypeScript	C++	Kotlin	Objective C

Ilustración 10 - Lenguajes de programación más populares

En lo que a este proyecto se refiere, la selección del lenguaje de programación del lado del servidor ha sido totalmente pragmática. Lo poco que he programado en los últimos años ha sido con Python, y es el lenguaje con el que tengo más herramientas y fluidez para poder desarrollar con éxito este proyecto.

4.2 Selección Framework de desarrollo

Existen diferentes frameworks de desarrollo web para Python: Django, Flask, Web2py y Pyramid, y muchos otros menos conocidos.

En la selección del framework he primado la comunidad que tiene por detrás el framework, así como ciertas características principales que facilitan el desarrollo de la aplicación.

	Django	Flask	Web2py	Pyramid
Experiencia	●	●	●	●
Comunidad	●	●	●	●
Documentación	●	●	●	●
Funcionalidad de caja	●	●	●	●
Soporte BBDD	Principales BBDD de forma nativa	Necesita de otros módulos: SQLAlchemy, CouchDB, MongoDB...	Tiene scripts para generar el SQL de 10 motores de BBDD	Necesita de otros módulos: SQLAlchemy, CouchDB, MongoDB...

Ilustración 11 - Tabla comparativa Frameworks Web Python

Hay otros criterios como velocidad y rendimiento del framework que para el desarrollo de este trabajo final de máster no son importantes, pero que se deberán tener en cuenta si se quisiera salir a producción.

Así que el framework que voy a utilizar en el desarrollo de este TFM es Django, en su versión 3.

4.3 Funcionamiento y estructura del framework Django

Muchos frameworks de desarrollo web se basan en la arquitectura de software MVC (Modelo Vista Controlador) [13].

Esta arquitectura separa en tres capas la parte de los datos de la lógica de negocio, así como de la parte de gestionar los eventos y las comunicaciones.

Las capas se interrelacionan entre sí de la siguiente forma:

- El Modelo: es la capa que se encarga de gestionar los datos. Recibe las peticiones del controlador y sirve los datos necesarios a la capa de la vista.
- El controlador: es quien recibe las peticiones y las trata, realizando las peticiones oportunas al modelo. A veces también interactúa con la vista, si son peticiones de interfaz y no de los datos, por ejemplo, hacer un scroll en la página.
- La vista: se encarga de mostrar de una forma concreta la información proporcionada por el modelo al usuario.

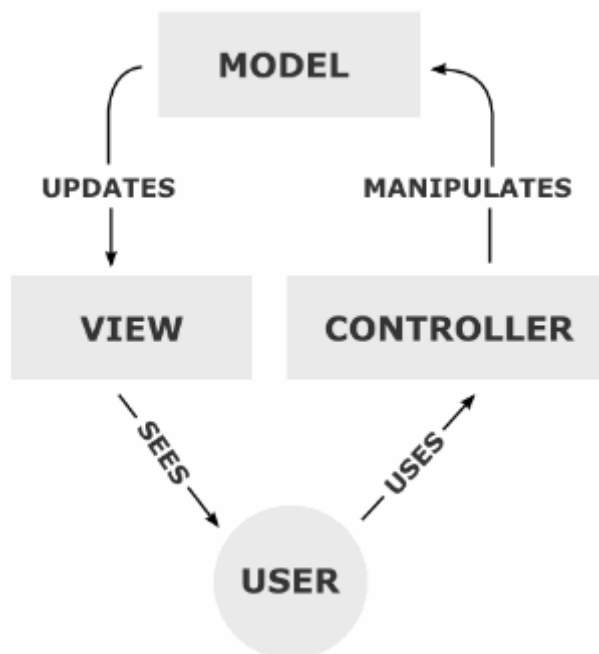


Ilustración 12 - Arquitectura MVC (Foto de la Wikipedia)

Django ha realizado su interpretación de esta arquitectura, y ha redefinido los términos. Según sus autores Django sigue una arquitectura MTV (Modelo Template Vista) [14].

Según su punto de vista, la definición de Vista es la capa que se encarga de gestionar los datos que se van a visualizar. Mientras que el Template será la capa que indique cómo visualizarlos. En esta definición de arquitectura, la capa del controlador quedaría definida por el framework

en sí mismo, que se encarga de gestionar las peticiones y hacerlas llegar a la vista correspondiente.

4.3.1 Estructura de ficheros de un proyecto Django

Django estructura el código en 2 partes. La capa superior que es el proyecto, que representa el conjunto del sitio web, aunando todas las funcionalidades que se desarrollen o incorporen en las diferentes aplicaciones.

En este TFM, se ha generado un proyecto llamado expID. En la siguiente imagen se puede observar la estructura de ficheros que componen el proyecto.

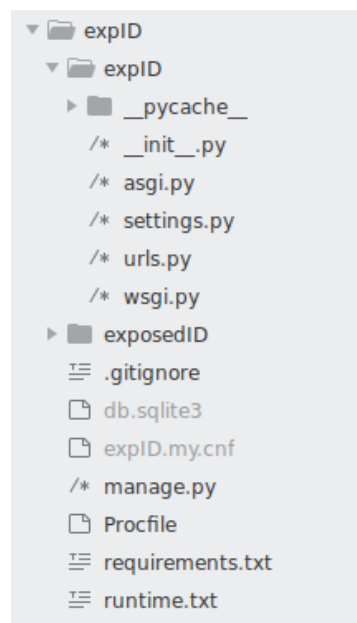


Ilustración 13 - Estructura del proyecto Django de este TFM

Los ficheros dentro de la subcarpeta expID son los que definen el proyecto, los valores de configuración (settings.py), como arrancarla desde servidores web compatibles con asgi (asgi.py) o con wsgi (wsgi.py) así como la definición de las URN principales que gestiona este proyecto.

Las aplicaciones son la otra parte. Un proyecto puede contener 1 o múltiples aplicaciones. Estas aplicaciones son funcionalidades autocontenidas que como pasa con las funciones se pueden reaprovechar entre proyectos. El más claro ejemplo de aplicación es la parte *Admin* de Django, ya que es una aplicación que se incorpora al proyecto para evitarle al desarrollador tener que escribir las funcionalidades de gestión básicas de usuarios, roles y tratamiento de forma genérica de los datos definidos en el resto de las apps del proyecto.

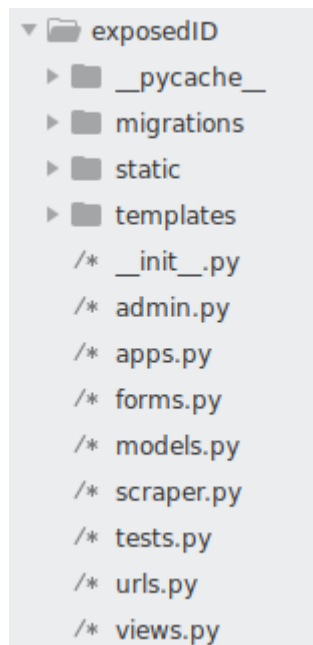


Ilustración 14 - Estructura de la aplicación de este TFM

La función de los principales ficheros es:

- Migrations: en esta carpeta se encuentran todos los ficheros que genera el propio framework para trasladar la definición de los modelos de datos a la estructura de tablas de la base de datos.
- Templates: es la carpeta donde se encuentran las diferentes plantillas para construir las páginas web del servicio.
- Forms.py: es el fichero donde se recoge la definición y tratamiento de los formularios necesarios para el servicio.
- Models.py: es el fichero donde están definidos todos los datos que gestiona expID, según se ha definido en el modelo de datos.
- Views.py: es el fichero donde se definen como tratar las peticiones de los usuarios, y que datos son necesarios para responder a ellas.
- Urls.py: aquí se definen las URN que gestiona esta aplicación dentro del proyecto, y se relacionan estas con las diferentes funciones definidas en views.py.

4.3.2 Repositorio de código

Se ha utilizado la herramienta Git para hacer un control de versiones del código fuente que va codificando.

El código también ha sido subido a GitHub, <https://github.com/giyevm/expID>, con dos objetivos principales:

1. Tener una copia de respaldo en remoto.
2. Tener la posibilidad de recibir colaboraciones futuras para seguir desarrollando el proyecto, ampliando sus funcionalidades por parte de otros usuarios.

4.4 Implementación del modelo de datos

El motor de BBDD que voy a utilizar es MySQL por su popularidad, buen rendimiento, y la cantidad de documentación que se puede encontrar para resolver problemas, frente a otra de las BBDD más utilizadas hoy en día de software libre que es PostgreSQL, que consume más recursos, aunque tiene sus ventajas en cuanto a robustez y potencia con gran concurrencia y volumen de datos. [15]

Dado que Django permite abstraer el motor de la BBDD del desarrollo en el framework, me he decantado por instalar MySQL por los motivos anteriores, sobre todo por la documentación, ya que ante cualquier problema me va a ser más sencillo solucionarlo y este proyecto no tiene como foco construir una aplicación que sea eficiente y muy escalable, el principal objetivo es construir la aplicación. Para pasar a producción la aplicación se podría modificar Django para utilizar en producción una BBDD PostgreSQL, aun habiendo realizado todo el desarrollo con MySQL, si se considerará que en un análisis en profundidad de las consultas que se vayan a realizar a BBDD PostgreSQL va a tener una ventaja competitiva respecto a poner en producción un MySQL.

La implementación del modelo de datos definido se ha materializado como diversas tablas en la BBDD, cada tabla implementando un tipo de dato concreto.

Las relaciones N:N requieren de tablas pivote que relacionen las entradas de ambas tablas. A parte de su función básica, nos permite añadir campos adicionales que afectan a cada una de las relaciones, y que he aprovechado para mapear ciertas características extra de los datos personales.

A continuación, muestro como han quedado definidas las tablas:

- Data
 - data_name [string]: nombre del dato personal, ej. DNI
 - confidentiality [int]: valor de la confidencialidad del dato
 - exposed [int]: valor de la exposición del dato
 - category [int]: valor de la categoría del dato personal
 - description [text]: descripción del dato personal.
 - created [date]: fecha de creación del dato personal
 - last_update [date]: fecha de la última modificación en el registro
- Pivote Site - Data

No voy a definir los campos básicos que se utilizan en las tablas pivote, solamente los campos extra que he añadido.

 - shadow [bool]: indica si el dato personal se recoge de forma fácilmente identificable para el usuario o no.
 - logged [bool]: indica si el dato personal se recoge solamente cuando se está registrado o también al navegar como invitado en la web.

- inferred [bool]: indica si el origen del dato está inferido de otros datos o lo recopila el sitio de forma directa.
- optional [bool]: indica si el usuario está obligado o no a informar de este dato.
- Site
 - site_name [string]: nombre del site
 - url [string]: dirección web que tenga el site, si la tiene
 - description [text]: descripción del sitio
 - cooperation [foreignkey]: apunta a la empresa matriz de quién depende este site
 - count_data [int]: número de datos personales que tiene este site
 - conf_average [float]: media del valor de los campos de Confidencialidad de los datos personales que mapea
 - exposed_average [float]: media del valor de la Exposición de los datos personales que agrupa el site
 - created [date]: fecha de creación del site
 - last_update [date]: fecha de la última modificación en el registro
- Corp
 - corp_name [string]: nombre de la corporación o empresa matriz.
 - conf_average [float]: media de los valores de Confidencialidad de los sites que tiene asociados
 - exposed_average [float]: media de los valores de Exposición de los sites asociados
 - created [date]: fecha de creación del registro
 - last_update [date]: fecha de la última modificación en el registro
- Perfil (Usuario)
 - login [string]: nombre de usuario
 - password [string]: contraseña del usuario
 - count_sites [int]: contador del número de sites asociados a su perfil
 - email [string]: email del usuario
 - conf_average [float]: media de los valores de Confidencialidad de los sites que tiene asociados
 - exposed_average [float]: media de los valores de Exposición de los sites que tiene el usuario
 - created [date]: fecha de creación del usuario
 - last_update [date]: fecha de la última modificación en el registro

4.5 Permisos de los usuarios

Se han definido 3 roles para controlar las capacidades a las que pueden acceder los usuarios:

1. **Recolector:** rol básico que se asigna a todo usuario recién registrado. Con este rol, el usuario puede añadir nuevos registros de todos los tipos de datos (dato personal, sitio y corporación)

pero no tiene permisos para hacer ninguna modificación o eliminación de datos.

2. **Editor:** con este rol, un usuario puede además de añadir nuevos datos, modificar los ya existentes en el servicio.
3. **Inquisidor:** los usuarios que tienen este rol pueden eliminar registros de la BBDD.

En la implementación actual del servicio, se asigna de forma automática el rol *Recolector* a todo usuario que se registre, pero no se ha generado ningún mecanismo para gestionar los otros dos roles, por parte de los usuarios, solamente se puede realizar su gestión por parte de la interfaz de administrador de Django.

4.6 Enrutador de la aplicación

El framework Django, requiere de un enrutador que se encarga de procesar las peticiones y hacerlas llegar según la URL que se solicita a la función pertinente.

En este desarrollo he generado un enrutador específico para la aplicación del TFM, permitiendo que en un futuro se pudiera distribuir esta como una aplicación de Django y no interfiera con otras posibles aplicaciones instaladas en el proyecto.

Cada función de la aplicación puede estar compuesta por una o varias URLs, que pueden llegar como peticiones de diferentes tipos HTML, como GET o POST. Se ha utilizado la sentencia *url* del framework para definir a que función de la capa view del framework se envía la información, delegando el tratamiento del verbo html utilizado a la propia función.

En el [Anexo 9.1](#) se pueden observar todas las urls implementadas.

4.7 Desarrollo operaciones con Datos

En este y los siguientes puntos de desarrollo de las diferentes operaciones y funciones del servicio *Exposed ID*, se exponen las decisiones tomadas a la hora de desarrollarlas sin entrar en el detalle técnico de cómo se han materializado esas decisiones en las líneas de código.

Los registros de los datos personales son la base de este proyecto, pero las operaciones con este tipo de datos son de las más sencillas.

Las principales decisiones durante el desarrollo han sido:

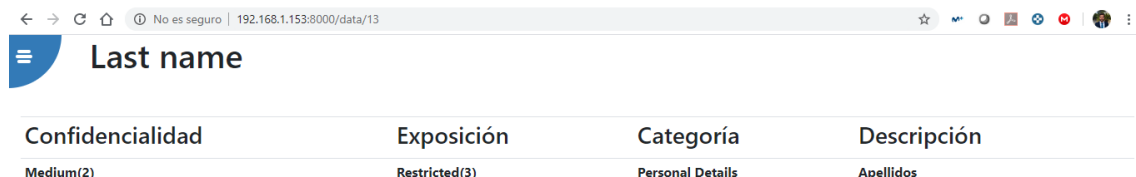
- *Valores estáticos:* en los datos personales, las tres variables principales que los describen están codificadas como un conjunto de valores estáticos. Se ha codificado así para limitar las opciones de los usuarios y de esta forma acotar el error que se puede

producir cuando un usuario clasifica un dato. Siempre existe la posibilidad de que un mismo dato dos usuarios lo habrían clasificado de forma diferente, pero al tener un conjunto finito de valores posibles con sus definiciones condicionamos la respuesta de los usuarios disminuyendo la posibilidad de que la clasificación de dos usuarios diferentes fuera diferente.

- *No se pueden borrar*: he tomado la decisión de que una vez introducido un dato personal en la BBDD no existe la posibilidad de que un usuario del servicio pueda eliminar este dato. De esta forma se evita que puedan desaparecer datos que ya fueron mapeados en algún sitio facilitando su incorporación en nuevos sitios.
- *Actualización Sitios*: cuando se almacena un dato se comprueba si el dato está relacionado con algún sitio y de ser así se dispara su actualización para reflejar el posible cambio del valor de alguno de los atributos confidencialidad y exposición medias.

La interfaz web del usuario con los datos personales la he estructurado en 2 pantallas:

1. Detalles de los datos personales. En esta pantalla se puede observar todos los atributos que conforman un dato personal, y en caso de tener los permisos correspondientes realizar su modificación.



The screenshot shows a web browser window with the address bar displaying "No es seguro | 192.168.1.153:8000/data/13". The page title is "Last name". Below the title is a table with four columns: "Confidencialidad", "Exposición", "Categoría", and "Descripción". The table contains one row of data: "Medium(2)", "Restricted(3)", "Personal Details", and "Apellidos".

Confidencialidad	Exposición	Categoría	Descripción
Medium(2)	Restricted(3)	Personal Details	Apellidos

Este dato lo recogen 2 sitios.

Ilustración 15 – Captura interfaz ExpID: Detalle Dato Personal

2. Formularios de creación/actualización: se utiliza un mismo formulario para la creación o modificación de los datos personales.

← → ↻ 🏠 ⓘ No es seguro | 192.168.1.153:8000/data/create/

Nuevo Dato Personal

Data name:

Confidentiality: Confidentiality level

Exposed: Exposed level

Description:

Category: Data Category

Ilustración 16 - Captura interfaz ExpID: Formulario Dato Personal

3. Listado de datos personales. Esta pantalla permite al usuario navegar por todos los datos personales que tiene registrados el servicio.

← → ↻ 🏠 ⓘ No es seguro | 192.168.1.153:8000/datas/

Lista de Datos Personales

Personal Details	
Dato Personal	Descripción
Apellido/s	Apellidos de la persona
Email	Correo electrónico
Last name	Apellidos
Middle Name	Nombre intermedio
Móvil	Teléfono móvil
Nombre	Nombre de pila de una persona
Teléfono	Teléfono fijo
Financial Details	
Dato Personal	Descripción
Cuenta bancaria	Cuenta bancaria
Tarjeta de crédito	Datos de la tarjeta de crédito
Profiling Details	
Dato Personal	Descripción

Ilustración 17 - Captura interfaz ExpID: Lista de Datos Personales

4.8 Desarrollo operaciones con Sitios

Las principales decisiones tomadas en la implementación de las operaciones con los Sitios (Sites) son:

- *Actualización de medias:* los valores de los atributos de un sitio que contienen las medias de exposición y confidencialidad se actualizan cada vez que se modifica un dato que se guarda el registro en la BBDD.

La implementación se ha realizado modificando la función guardar del modelo de datos, de forma que siempre que sea llamada esta actualice el valor de estos campos, de esta forma desde otros puntos del código se puede actualizar estos valores, como por ejemplo cuando se guarda un dato, ya que puede afectar al valor de estas medias.

- *Actualización Corporación:* cuando se almacena un Site se comprueba si está relacionado con alguna corporación y de ser así se dispara su actualización para reflejar el posible cambio del valor de alguno de los atributos confidencialidad y exposición medias.

La interfaz de usuario para tratar con este tipo de datos sigue la misma línea que para los datos personales, pero con un gran cambio en el formulario de los sitios.

1. Detalles de los sitios: se ven todas las características del sitio, junto con un listado de los datos que recoge y las características de como los recopila.

Amazon (Amazon Co)

NIVEL CONFIDENCIALIDAD: 4.2 NIVEL EXPOSICIÓN: 3.0 URL: <https://amazon.com> Descripción: Marketplace de compras

Datos Personales

Nombre	Categoría	Confidencialidad	Exposición	Recogido Oculto	Usuarios Logados	Inferido	Opcional
Apellido/s	Personal Details	Restricted	Medium	👍	👎	👍	👎
Móvil	Personal Details	Restricted	High	👍	👎	👍	👎
Nombre	Personal Details	Top Secret	High	👍	👎	👍	👎
Teléfono	Personal Details	Confidential	High	👍	👎	👍	👎
Tarjeta de crédito	Financial Details	Secret	Critical	👍	👎	👍	👎

Ilustración 18 - Captura Interfaz ExpID: Detalles de un Sitio

2. Formularios: se ha modificado el formulario por defecto para la introducción de un Site, para que refleje y permita introducir los atributos que definen la relación de un sitio con sus datos asociados. Esto se ha realizado utilizando conjuntos de formularios (formsets) con su variante en línea (inlineFormSet).

Site name:

S url:

Description:

Corporation:

Data: Shadow: Loged: Inferred: Optional: Delete:

Data: Shadow: Loged: Inferred: Optional: Delete:

Data: Shadow: Loged: Inferred: Optional: Delete:

Data: Shadow: Loged: Inferred: Optional: Delete:

Data: Shadow: Loged: Inferred: Optional: Delete:

Data: Shadow: Loged: Inferred: Optional: Delete:

[Cancel](#)

Ilustración 19 - Captura interfaz expID: Formulario de Sitio

3. Lista de Sitios: se muestra una lista paginada de todos los Sitios que contiene el servicio. Mostrando las características principales, y las posibles acciones a realizar.

Sitio (Cooperación)	Confidencialidad	Exposición	
Amazon (Amazon Co)	4.2	3.0	
El mundo	6.0	3.0	
ExpID	6.0	3.0	
Facebook	4.0	2.7	
girinst	0.0	0.0	
jaki	0.0	0.0	
LinkedIn	3.0	2.0	
Prueb7	3.0	3.0	
Prueba10 (CoorpPrueba)	0.0	0.0	
Prueba11 (CoorpPrueba)	3.0	2.0	
Prueba4 (CoorpPrueba)	6.0	4.0	
prueba8 (Amazon Co)	5.0	3.5	

Page 1 of 2. [next](#)

Ilustración 20 - Captura Interfaz expID: Lista de Sitios

La actualización de las medias de los sitios se ha codificado modificando el método que tiene por defecto para guardar una entrada en la BBDD. Esta función también llama a la función guardar de la corporación para actualizar también las medias de ese registro.

```

125     def save(self, *args, **kwargs):
126         #do_something()
127         super(Site, self).save(*args, **kwargs) # Guardamos el registro para generar la relación Many2Many con los datos
128         #do_something_else()
129         s=Site.objects.filter(id = self.id).annotate(conf_avg=Avg('datas__confidentiality'), exposed_avg=Avg('datas__exposed'))
130         if s[0].conf_avg is not None and s[0].exposed_avg is not None:
131             self.conf_average = s[0].conf_avg
132             self.exposed_average = s[0].exposed_avg
133             #Site.objects.filter(id = self.id).update(conf_average = s[0].conf_avg, exposed_average = s[0].exposed_avg)
134             super(Site, self).save(*args, **kwargs) # Guardamos la actualización de los valores medios de confidencialidad y e
135         if self.corporation:
136             Corporation.objects.get(id=self.corporation.id).save()

```

Ilustración 21 - Código parcial: Función save del modelo Site

4.9 Desarrollo operaciones con Corporaciones

El desarrollo realizado para las corporaciones es similar a los realizados para los datos personales y los sitios.

La principal decisión tomada, propiciada por la limitación del tiempo del TFM es que, a diferencia de los sitios, cuando se da de alta una corporación no se pueden introducir a la vez los sitios que la componen, y esta relación se ha de establecer indicando en cada sitio a que corporación pertenece.

En cuanto a la interfaz gráfica del usuario es continuista con lo desarrollado para los otros tipos de datos.

- Vista de detalle de una corporación.

Amazon Co

Nivel Confidencialidad: 4.2

Nivel Exposición: 3.2

Sitios			
Nombre	Confidencialidad	Exposición	Descripción
Amazon	4.2	3.0	Marketplace de compras
prueba8	5.0	3.5	Red social

Ilustración 22 - Captura Interfaz expID: Detalle de una Corporación

- Formulario de entrada/modificación.

Formulario Corporación

Corp name:

Submit Cancel

Ilustración 23 - Captura Interfaz expID: Formulario Corporación

- Lista de corporaciones.

Corporación	Confidencialidad	Exposición
Amazon Co	4.2	3.2
CoopPrueba	3.0	2.0
Twitter Corp	3.5	2.8

Ilustración 24 - Captura Interfaz expID: Lista de Corporaciones

Al igual que para los registros de los Sites, en las corporaciones también se ha sobrescrito la función guardar para que se actualicen las medias de exposición y confidencialidad cuando se guarda un registro.

```

31     def save(self, *args, **kwargs):
32         super(Corporation, self).save(*args, **kwargs) # Guardamos el registro para generar la relación Many2Many con los datos
33         s=Corporation.objects.filter(id = self.id).annotate(conf_avg=Avg('corporations__conf_average'), exposed_avg=Avg('corporations__exposed_avg'))
34         if s[0].conf_avg is not None and s[0].exposed_avg is not None:
35             self.conf_average = s[0].conf_avg
36             self.exposed_average = s[0].exposed_avg
37         #Corporation.objects.filter(id = self.id).update(conf_average = s[0].conf_avg, exposed_average = s[0].exposed_avg)
38         super(Corporation, self).save(*args, **kwargs) # Guardamos la actualización de los valores medios de confidencialidad

```

Ilustración 25 - Código parcial: Función save del modelo Corporación

4.10 Desarrollo operaciones del perfil y usuarios

Esta es una de las características que más valor le aporta al usuario, porque le permite construir su perfil de datos expuestos en internet. Para desarrollarlo, he tomado las siguientes decisiones:

- *Incorporación sencilla de Sitios al perfil:* he desarrollado un sencillo sistema que permite mediante 2 acciones o asociar o desasociar un sitio al perfil de un usuario. La incorporación de los sitios a un perfil se puede realizar desde diferentes puntos del servicio expID, mientras que eliminar esta asociación solamente se puede realizar desde la página del perfil del usuario.
- *Métricas condicionadas:* los valores de exposición y confidencialidad medios que se presentan en el perfil del usuario tienen en cuenta si los datos que recoge un Sitio son opcionales o no, haciendo que los datos opcionales tengan un peso de la mitad.

He tomado esta decisión, ante la imposibilidad temporal de desarrollar un sistema más personalizado. De esta forma la métrica mostrada en el perfil se ajusta un poco y será más fiel a la realidad para los usuarios que no introducen todos los datos opcionales de un Sitio. Es verdad que, al realizar este ajuste, estas métricas van a ser una referencia para todos los usuarios en lugar de ser la métrica exacta de su exposición y confidencialidad.

En cambio, en los Sitios se calcula la media de estos atributos con el mismo peso para todos los datos ya que el objetivo del sitio es conseguir todos los datos y de esta forma se les

muestra a los usuarios el valor máximo posible, lo que sería el peor escenario para ellos.

La interfaz gráfica implementada para esta funcionalidad tiene los siguientes elementos:

1. Página del perfil: aquí se muestra toda la información relacionada con el perfil del usuario.

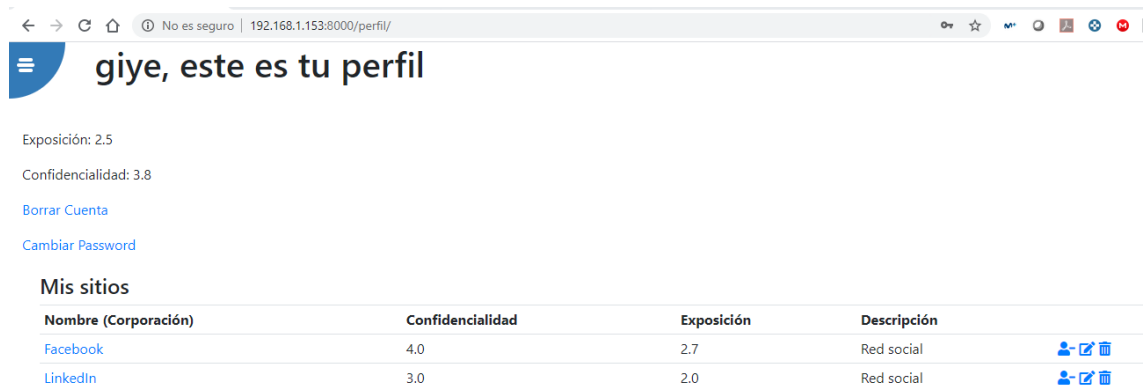


Ilustración 26 - Captura Interfaz expID: Página Perfil usuario

2. Formulario de registro: es la forma de crear una cuenta en el servicio expID.

Usuario:

Email:

Password:

Confirma Passowrd:

[¿Has olvidado tu password?](#)

Ilustración 27 - Captura Interfaz expID: Formulario de registro de usuario

4.11 Desarrollo del dashboard

El dashboard constituye otra de las funcionalidades que más valor le aportan al usuario.

Desde este cuadro de mando, el usuario va a poder ver las estadísticas de todos los datos que tiene registrados el servicio Exposed ID, así como métricas globales, con las que poder compararse respecto al resto de usuarios.

Las métricas que se han implementado para todos los tipos de datos (datos personales, sitios, corporaciones y perfiles) son:

- *Conteo de registros*: se ha realizado un conteo de los registros que hay de ese tipo en la BBDD. Esta métrica le permite al usuario hacerse una idea de si el resto de las métricas se han calculado sobre un conjunto amplio de datos o no.
- *Máximo y mínimo*: se han extraído los valores máximo y mínimo de los atributos confidencialidad y exposición de todos los registros, para mostrarle al usuario en que horquilla se mueven los datos registrados en el sistema.
- *Media y Desviación Estándar*: se han calculado estos valores para el conjunto de datos de un mismo tipo. Con lo que el usuario puede comprender en qué valores se mueven la mayoría de los registros almacenados.

Para los Datos Personales también se han implementado los siguientes gráficos:

- *Datos por categoría*: Donde se muestra el porcentaje de datos existentes en el servicio de cada categoría.

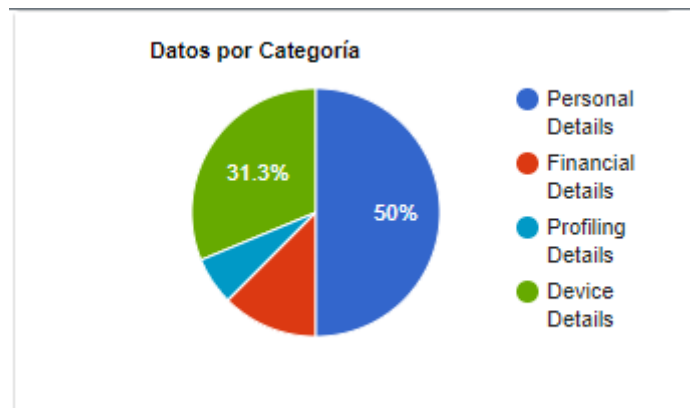


Ilustración 28 - Dashboard: gráfica de Datos por categoría

- *Como se recogen*: indica en porcentaje como se están recogiendo todos los datos que se tienen almacenados por parte de los Sitios.

Recogida de datos

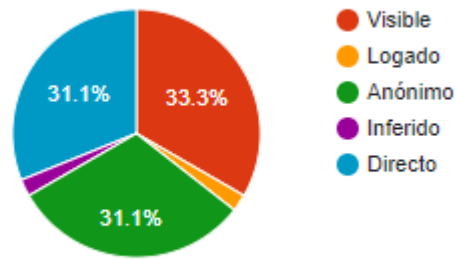


Ilustración 29 - Dashboard: gráfica “Recogida de datos”

- *Top 10 datos*: es una gráfica que muestra los 10 datos más recopilados por los sitios, y de qué forma se están recogiendo.

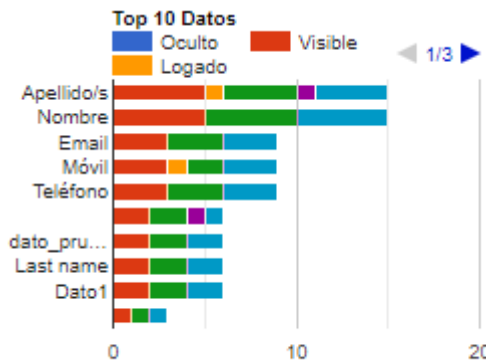


Ilustración 30 - Dabshoard: gráfica “Top 10 Datos”

Respecto a los Sitios, también se han incorporado 2 gráficas extra:

- *Resumen de sitios*: de una forma visual, muestra todos los sitios registrados, con el tamaño del círculo indica si recoge muchos datos personales, y con el color si es un Sitio muy utilizado por los usuarios, y por tanto lo tienen muchos en su perfil.

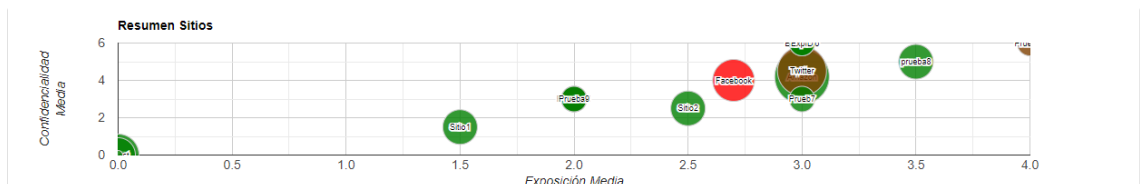


Ilustración 31 - Dashboard: gráfica “Resumen de Sitios”

- *Top Sitio por tipo de Dato*: es una tabla en la que se enlazan los Sitios que más y menos datos de una categoría tienen.

TOP Sitio por tipo de Dato

Categoría	Sitio MAX(#Datos)	Sitio MIN(#Datos)
Datos Personales	Facebook(6)	-
Datos Financieros	Twitter(1)	-
Datos Identificativos	-	-
Datos de Empleo	-	-
Datos de Localización	-	-
Datos de Perfilado	Prueba4(1)	-
Datos Médicos	-	-
Datos de Dispositivos	Sitio1(2)	-

Ilustración 32 - Dashboard: tabla "TOP Sitio por tipo de Dato"

En el perfil se ha incorporado una gráfica extra:

- Mi perfil vs la media: en esta gráfica se muestran las principales características del perfil del usuario, respecto a la media de los perfiles, teniendo en cuenta los umbrales de una desviación estándar, para que el usuario sepa si su perfil se encuentra dentro de los valores de la mayoría de los usuarios o en algún extremo.

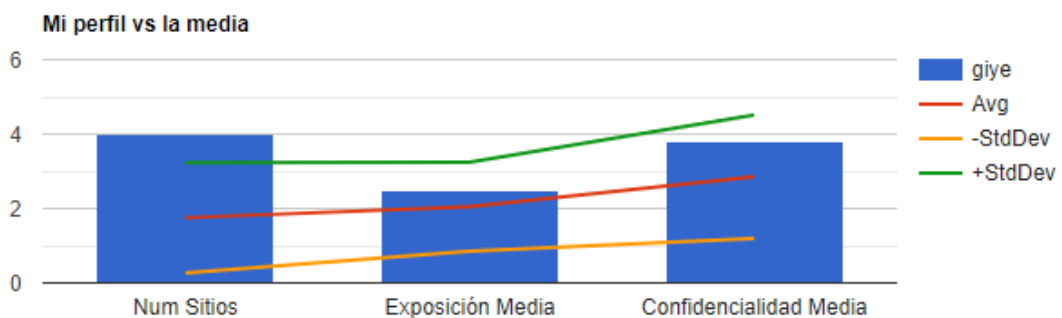


Ilustración 33 - Dashboard: gráfica "Mi perfil vs la media"

La parte más importante de este desarrollo es preparar correctamente los datos que se envían desde la vista al template, con el objetivo de poder reaprovechar el código lo máximo posible.

Los datos se han preparado incluyéndolos en una sola variable de contexto, que luego dentro del template se procesa y se envía a los diferentes subtemplates necesarios.

1. Buscador:

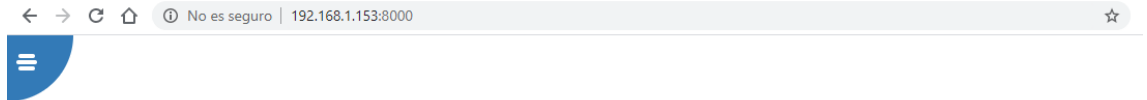
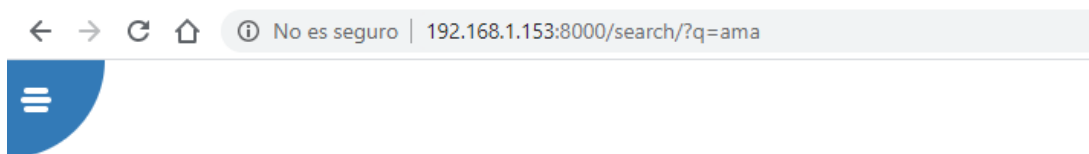


Ilustración 36 - Captura Interfaz expID: Página del buscador

2. Página de resultados:



Sitios encontrados

- [Amazon \(https://amazon.com\)](https://amazon.com)

Coorporaciones encontradas

- [Amazon Co](#)

Ilustración 37 - Captura Interfaz expID: Página de resultados del buscador

4.13 Desarrollo del Scrapper web

Esta funcionalidad facilita la incorporación de nuevos Sitios en el servicio expID, haciendo que el usuario no tenga que realizar el alta de forma manual.

Las decisiones tomadas para su implementación son:

- *Análisis URL indicada:* solamente se va a analizar la página de la URL que se proporciona por el usuario, por lo que es necesario que el usuario introduzca la URL del registro concretamente.
- *Análisis de los campos Input:* los campos que se buscan dentro de la página para identificar qué datos personales están recopilando son las etiquetas HTML del tipo input. Si la página en cuestión utiliza alguna otra implementación que no haga uso de estas etiquetas para solicitar los datos al usuario, este scrapper no podrá mapear nada.

En la implementación realizada, se buscan estas etiquetas en toda la página, por lo que si hay más campos input en la página aparte de los que componen el formulario de entrada, estos serán analizados, y en caso de encontrarse una coincidencia se incluirán dentro de la ficha del Sitio que se crea.

- *Comparativa a nivel de nombre:* el análisis de los campos input encontrados se hace extrayendo su atributo "ID" o "Name" y este se compara literalmente contra el nombre de todos los Datos registrados en la BBDD del servicio. La única transformación que se realiza para esta comparación es que se elimina cualquier carácter especial y espacios. A parte de realizar una comparación con ambos nombres convertidos en minúsculas.

Este algoritmo de comparación es capaz de procesar correctamente muchas páginas de registro, pero existen páginas que utilizan abreviaciones en estos campos, u otro tipo de identificación y eso va a hacer que con este análisis no se puedan identificar los datos personales que están recopilando.

- *Edición manual posterior al análisis:* dado que el código del scrapper puede fallar a la hora de identificar los campos, o que el usuario sea consciente de otros datos personales que recoge el sitio aparte de los del registro. Cuando el scrapper acaba de realizar el análisis y generar la entrada en la BBDD para el sitio indicado, redirige al usuario a la página de modificación de dicho registro por si desea realizar alguna aportación manual, siempre que tenga permisos para editar registros, si no se le mostrará lo creado de forma automática.

A nivel de la interfaz web, solo hay una pantalla nueva que tiene el siguiente diseño:



← → ↻ 🏠 ⓘ No es seguro | 192.168.1.153:8000/analize/

Analizador de Sitios

;RECUERDA! Pega la URL completa de la página de REGISTRO del sitio que quieres analizar

url:

Ilustración 38 - Captura Interfaz expID: Scrapper (Analizador de Sitios)

Su desarrollo se ha realizado haciendo uso de la librería *BeautifulSoup4* [16], para el análisis de documentos.

Con esta librería se extrae de la página descargada todos los elementos HTML *input* que tiene, para luego revisar si su atributo *id* o *name*

coinciden con alguno de los datos personales que están registrados en la BBDD.

También se extraen los datos del nombre del sitio y su URL, para crear la entrada del site en la BBDD.

```
337     soup = BeautifulSoup(html_content, "lxml")
338     datos_entrada = soup.find_all("input")
339     print(soup)
340     print(datos_entrada)
341     nombre_sitio = urlparse(url).netloc.split(".")
342     if len(nombre_sitio)>2:
343         if nombre_sitio[0] != "www":
344             nombre_sitio = nombre_sitio[0]+"."+nombre_sitio[1]
345         else:
346             nombre_sitio = nombre_sitio[1]
347     else:
348         nombre_sitio = nombre_sitio[0]
349
350     sitio,created = Site.objects.get_or_create(
351         site_name=nombre_sitio,
352         s_url=urlparse(url).scheme+'://'+urlparse(url).netloc,)
353     if created == True:
354         dbdatos = Data.objects.all()
355         for dato in datos_entrada:
356             print(dato.get("id")+" -- "+dato.get("name"))
357             if dato.get("id") != None:
358                 for dbdato in dbdatos:
359                     if compare(dato.get("id") , dbdato.data_name):
360                         sitio.datas.add(dbdato)
361                         break
362             elif dato.get("name") != None:
363                 for dbdato in dbdatos:
364                     if compare(dato.get("name") , dbdato.data_name):
365                         sitio.datas.add(dbdato)
366                         break
```

Ilustración 39 - Código parcial: Análisis de sitios (scraper web)

4.14 Desarrollo del Menú

El menú es una pieza fundamental de todo servicio, ya que guía al usuario en las diferentes funcionalidades que puede utilizar. Por esta razón, se ha desarrollado un menú sencillo y minimalista.

Este menú consta únicamente de 9 posibilidades que se presentan en este orden:

1. *Home*: Enlace que te lleva a la página principal del servicio, donde se encuentra el buscador.
2. *Login/Logout*: esta opción del menú cambia de icono y permite abrir o cerrar la sesión de usuario.
3. *Perfil*: aquí el usuario podrá acceder a su perfil de usuario.
4. *Listado de Corporaciones*
5. *Listado de Sitios*
6. *Listado de Datos Personales*
7. *Dashboard*

8. *Analizador de webs*: este enlace te lleva a la funcionalidad del scrapper web para poder incorporar de forma automática un sitio y los datos que recoge en el registro.
9. *Información del sitio*

Visualmente el menú se ha codificado de la siguiente forma:



Ilustración 40 - Captura Interfaz expID: Menú

5 Limitaciones y trabajo futuro

5.1 Limitaciones de la implementación realizada

El trabajo realizado presenta las funcionalidades básicas para poder validar el objetivo principal.

Una aplicación con la que los usuarios puedan generar su perfil de la exposición de sus datos en internet. Para llegar a lograr este objetivo, se han desarrollado diversas funcionalidades, las cuales por un compromiso de coste y complejidad no se han podido desarrollar completamente y presentan ciertas limitaciones.

5.1.1 Gestión de los roles de los usuarios.

Como se ha descrito en el apartado [4.4](#) se han creado 3 roles diferentes, que permiten realizar diferentes acciones sobre los tipos de datos.

Lo que no se ha podido desarrollar es unos procesos automáticos o basados también en roles que permitan que un usuario progrese y cambie de rol, permitiéndole mayor funcionalidad en el servicio. Ahora mismo los usuarios una vez registrados están limitados a interactuar siempre con el servicio con el rol de *Recolector*, y la única opción disponible para gestionar los roles de cada usuario es que un administrador de Django modifique estos permisos en la zona de administración del framework o directamente en la BBDD.

Una posible solución futura sería implementar un sistema similar a la Wikipedia, donde todo el mundo puede modificar ciertos contenidos, pero con un sistema de aprobación. También integrado con un sistema de puntos que permita en función de las aportaciones realizadas al servicio ir promocionando en los diferentes roles.

5.1.2 Sinónimos y traducciones de los datos

La implementación de la BBDD en cuanto a la recolección de los datos está centrada en un solo idioma, lo que limita *extracción automática de datos*.

En el apartado [4.11](#) se ha detallado como se realiza la extracción de los datos de una página web de forma automática. Al comparar el campo ID con el nombre de los datos en la BBDD, ahora mismo estamos limitados a que ese valor exista literalmente (excepto los signos de puntuación y capitalización del texto) como una entrada, lo que obliga al servicio a tener un registro duplicado en la BBDD por cada traducción o tipo sinónimo del tipo de dato que se quiera ser capaz de mapear.

Por ejemplo: para que el sistema sea capaz de identificar el/los apellidos de una persona en castellano e inglés deberíamos disponer de los siguientes registros en la tabla de datos.

- Apellido
- Apellidos
- Primer Apellido
- Segundo Apellido
- Last Name
- Family Name
- Middle Name
- ...

La solución pasa por buscar modificar el modelo de datos, de forma que se puedan asociar diferentes nombres a un mismo dato.

5.1.3 Limitaciones por idioma en el scrapper

Esta limitación comparte características con la limitación anterior [5.1.2 Sinónimos y traducciones de los datos](#). Si en el anterior se genera un problema a la hora de mantener los datos para poder tener una web accesible en diferentes idiomas para todos los posibles usuarios.

La limitación del scrapper viene por el idioma en el que se haya codificado los atributos en la web a analizar. En el desarrollo actual, va a buscar literalmente contra todos los datos introducidos en la BBDD, pero para una web escrita en un idioma que no utilicen caracteres latinos se producirían errores.

La solución está en generar un sistema de traducciones de los datos y sus posibles variaciones (5.1.2) que sea accesible por el scrapper para poder identificar los campos de los formularios de entrada.

5.1.4 Localización de datos

El proceso de procesado de una web de registro para la extracción automática de los datos que se recogen por la compañía para dar de alta a un usuario es bastante básico y requiere que en la web se haya utilizado el nombre completo del dato como atributo id del campo input.

Si no se dan estas condiciones entonces el scrapper implementado no es capaz de identificar el campo y no lo mapea.

La solución pasa por analizar más campos y atributos presentes en los formularios de entrada, generando un scrapper con una lógica más compleja, pudiendo llegar a incorporar técnicas de Machine Learning o Inteligencia Artificial.

5.1.5 UX

Se puede mejorar toda la interfaz de usuario del servicio desarrollado en este proyecto.

5.1.6 Categoría de datos personales

Los datos personales se pueden clasificar en diferentes categorías de datos personales. Actualmente esas categorías son valores estáticos definidos en el código de la aplicación, por lo que cualquier inclusión o modificación de estas tiene que hacerse con una nueva entrega del software.

Una mejora sería modificar el código para que sea un campo editable más, que permita una evolución de las categorías por parte de los usuarios del servicio.

5.1.7 Formularios de Corporaciones

El formulario de creación o modificación de una Corporación solo permite introducir sus datos concretos, que en este caso solo es el nombre de la corporación. No se permite gestionar desde este extremo de la relación los sitios que forman parte de la misma corporación.

La solución consistiría en realizar los InlineFormSet correspondiente para poder gestionar los Sitios desde el propio formulario de la Corporación.

5.1.8 Edición datos del scrapper

Con la implementación actual, los datos que recopila el scrapper los almacena directamente en la BBDD y si el usuario tiene permisos, se le redirige a la página de modificación al acabar para que ajuste o complete lo que considere. En el caso de no tener permisos, entonces simplemente se le muestra el registro creado.

Esto supone una limitación respecto a la creación manual que todo usuario tiene por defecto, ya que en esta opción si pueden completar todo lo que quieran la entrada del Sitio.

La solución consiste en poder mostrar al usuario los datos recogidos antes de crear la entrada en la BBDD y así modificarla a su antojo, para después guardarla.

5.2 Trabajo futuro

Además de desarrollar las soluciones a las limitaciones del desarrollo realizado, se proponen ciertas funcionalidades que aportarían mucho valor a los usuarios.

5.2.1 Scrapper Autónomo

Esta funcionalidad consiste en disponer de un scrapper autónomo que vaya recorriendo internet en busca de formularios de registro para ir indexando sitios y los datos que recopilan en el registro.

5.2.2 Scrapper Ampliado

Por medio de otras técnicas más complejas de aprendizaje automático, se dotaría al scrapper autónomo de las capacidades de interpretar datos personales que recogen los sitios web para incorporarlos a la ficha del sitio

5.2.3 Perfil detallado

En la versión actual el usuario es capaz de incorporar a su perfil qué sitios utiliza, pero es incapaz de indicar si en dichos sitios ha creado una cuenta, o ha introducido los datos opcionales.

Por eso, esta funcionalidad consistiría en modificar toda la parte del perfil para proporcionar la capacidad a los usuarios de poder detallar la relación que tienen con cada sitio que incorporan a su perfil indicando:

- Si han creado cuenta o solamente han navegado como invitado
- Qué datos personales han introducido y cuales no

Toda esta información afectará a los cálculos de sus KPIs de exposición y confidencialidad, haciendo que sea más ajustada y personal.

5.2.4 Scrapper de perfil

El objetivo tras esta funcionalidad es facilitar al usuario la construcción y mantenimiento de su perfil.

Consiste en implementar un elemento, plugin o sistema, que permita analizar el historial de navegación de los diferentes navegadores del usuario y que incorporen en su perfil de forma automática las páginas que ha visitado.

5.2.5 Sistema de alertas

Integrar el servicio de Exposed ID con otros servicios o webs donde se identifican ataques a compañías que han comprometido datos de estas, de forma que se pueda avisar a los usuarios que tuvieran esa empresa en su perfil sean avisados de que sus datos personales han sido comprometidos, y si existe el detalle indicar cuales.

5.2.6 Dashboar interactivo

Actualmente esta sección del servicio muestra ciertas estadísticas de forma estática, con unos filtros y varemos predefinidos en el momento de codificación. Aportaría mucho valor a los usuarios ser capaces dentro de ciertos límites poder filtrar y gestionar los datos que se muestran en el dashboard para adaptarlo a sus necesidades.

5.2.7 Buscador vitaminado

La finalidad es disponer de un buscador potente, que facilite al usuario encontrar la información que está buscando.

Actualmente, el buscador hace una sencilla búsqueda de texto en los campos de nombre y descripción de los diferentes tipos de datos, pero se debería incrementar esta capacidad a poder buscar por múltiples criterios, así como buscar en ciertos atributos de los diferentes tipos de datos, aplicar filtros a la búsqueda, etc.

6 Conclusiones

Cada vez la sociedad está más conectada y digitalizada, lo que provoca una exposición cada vez mayor de sus datos personales. Aunque los gobiernos son conscientes de ello, y están empezando a aplicar medidas para proteger a los usuarios, la mayoría de estos no tienen una visión global de lo que realmente están exponiendo ni dónde.

Este trabajo permite a los usuarios construir esa visión global de su identidad personal expuesta en internet, y donde está siendo expuesta.

Se han cumplido los objetivos marcados al inicio del proyecto. El usuario dispone de un marco de referencia con el evaluar los diferentes sitios, conocer fácilmente si recopilan muchos datos y la tipología de estos, así como los niveles de exposición que está asumiendo de forma inconsciente.

La metodología adoptada para el desarrollo del TFM, inspirada en métodos Agile ha aportado grandes ventajas, ya que ha permitido realizar ajustes de la planificación entre los diferentes Sprints. Esta libertad de planificación ha permitido reevaluar la viabilidad y razón de las funcionalidades inicialmente propuestas, modificándolas o reemplazándolas para poder disponer al final del proyecto de un producto más completo y adecuado.

Por las limitaciones inherentes a la naturaleza del TFM, el servicio Exposed ID aquí desarrollado se puede considerar un MPV (Mínimo Producto Viable) al que le queda mucho recorrido para estar finalizado, pero que en su estado actual ya presta las capacidades mínimas buscadas.

7 Glosario

- Exposed ID: nombre del servicio desarrollado en este TFM, también expresado como expID.
- IA: Inteligencia Artificial
- CRUD: acrónimo de Create, Read, Update y Delete. Las operaciones básicas que se pueden realizar con un objeto.
- MPV: acrónimo de Mínimo Producto Viable.
- URI: Uniform Resource Identifier. Es el identificador único de un recurso en internet. Se compone de la mezcla de URL+URN.
- URL: Uniform Resource Locator. Sirve para nombrar recursos en internet.
- URN: Uniform Resource Name, es la parte que identifica a un recurso sin localizarlo, es decir sin indicar el servidor en el que se encuentra. Es la parte de la URI que se encuentra a continuación del nombre de dominio (URL).
- BBDD: Base de Datos
- Sites: término utilizado como tipo de dato en el TFM, para registrar en el servicio expID todos los registros de aplicaciones, sitios web o empresas que recogen datos personales.
- Github: plataforma de repositorios remotos para el control de versiones con el protocolo git.

8 Bibliografía

- [1] - [The WIRED Guide to Your Personal Data \(and Who Is Using It\)](#) (6/5/2020)
- [2] - [RGPD - REGLAMENTO \(UE\) 2016/679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO de 27 de abril de 2016](#) (6/5/2020)
- [3] - Millard, C., & Hon, W. K. 2012. "Defining 'personal data' in e-social science. *Information, Communication & Society*", 15(1), 66-84.
- [4] - [¿Qué datos personales guarda cada empresa tecnológica? \[Infografía\]](#) (6/5/2020)
- [5] - [ToSDR \(Terms of Service, Didn't Read\)](#) (7/5/2020)
- [6] - [Guard](#) (7/5/2020)
- [7] - [Descubre qué datos recopilan las aplicaciones](#) (7/5/2020)
- [8] - [AppCensus](#) (7/5/2020)
- [9] - [Exodus Privacy](#) (7/5/2020)
- [10] - [CISSP Security Management and Practices](#) (7/5/2020)
- [11] - [Los mejores lenguajes de programación para cada tarea](#) (7/5/2020)
- [12] - [The Most Popular Programming Languages of 2019](#) (7/5/2020)
- [13] - [Definición Modelo – vista – controlador Wikipedia](#) (27/5/2020)
- [14] - [Definición de la arquitectura de Django](#) (27/5/2020)
- [15] - [Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL](#) - TESIS - López Herrera Patricia
- [16] - [Documentación Librería Beautiful Soup 4](#) (29/5/2020)

9 Anexos

9.1 URLs de la aplicación

En el framework Django se generan uno o varios ficheros encargados de gestionar las URLs de la aplicación. En el caso de este TFM, todas las URL de la aplicación Exposed ID se encuentran en un solo fichero, *urls.py*, dentro del paquete de la aplicación.

```
url(r'^datas/$', views.DataListView.as_view(), name='datas'),
url(r'^data/(?P<pk>\d+)$', views.DataDetailView.as_view(), name='data-detail'),
url(r'^data/create/$', views.DataCreate.as_view(), name='data_create'),
url(r'^data/(?P<pk>\d+)/update/$', views.DataUpdate.as_view(), name='data_update'),
url(r'^data/(?P<pk>\d+)/delete/$', views.DataDelete.as_view(), name='data_delete'),
#urls Sites
url(r'^sites/$', views.SiteListView.as_view(), name='sites'),
url(r'^site/(?P<pk>\d+)$', views.SiteDetailView.as_view(), name='site-detail'),
url(r'^site/create/$', views.SiteCreate.as_view(), name='site_create'),
url(r'^site/(?P<pk>\d+)/update/$', views.SiteUpdate.as_view(), name='site_update'),
url(r'^site/(?P<pk>\d+)/delete/$', views.SiteDelete.as_view(), name='site_delete'),
#urls Corporaciones
url(r'^corporations/$', views.CorporationListView.as_view(), name='corporations'),
url(r'^corporation/(?P<pk>\d+)$', views.CorporationDetailView.as_view(), name='corporation-detail'),
url(r'^corporation/create/$', views.CorporationCreate.as_view(), name='corporation_create'),
url(r'^corporation/(?P<pk>\d+)/update/$', views.CorporationUpdate.as_view(), name='corporation_update'),
url(r'^corporation/(?P<pk>\d+)/delete/$', views.CorporationDelete.as_view(), name='corporation_delete'),
#urls Perfil
path('accounts/', include('django.contrib.auth.urls')),
url(r'^register$', views.register, name='register'),
url(r'^perfil/$', views.PerfilView.as_view(), name='my-sites'),
url(r'^perfil/(?P<pk>\d+)/delete$', views.PerfilDelete.as_view(), name='perfil-delete'),
url(r'^perfil/(?P<pk>[-\w]+)/add$', views.add_site2profile, name='add2profile'),
url(r'^perfil/(?P<pk>[-\w]+)/del$', views.del_site2profile, name='del2profile'),
#otras urls
url(r'^$', views.index, name='index'),
url(r'^search/$', views.SearchResultsView.as_view(), name='search'),
url(r'^dashboard/$', views.dashboardView, name='dashboard'),
url(r'^info/$', views.infoView, name='info'),
url(r'^analize/$', views.scrapperView, name='scrapper'),
```

Ilustración 41 - Código: enrutador urls de explID