UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA INFORMÁTICA EN TECNOLOGÍAS DE LA INFORMACIÓN

"DESARROLLO DE UNA APLICACIÓN WEB PARA REALIZAR WEB SCRAPING SOBRE TABLAS DE DATOS"

TRABAJO FIN DE GRADO

Febrero - 21

AUTOR: Jorge Sánchez Bonillo
DIRECTOR/ES: Jesús Javier Rodríguez Sala

RESUMEN

En la actualidad la cantidad de datos que hay en la web es incalculable y muchos de estos datos están incrustados en tablas formateadas en páginas web. En muchas ocasiones es necesario extraer estos datos para investigaciones, archivo u otras necesidades. En la actualidad hay diferentes herramientas que permiten la extracción de datos, pero muchas de ellas están muy limitadas a la hora de modificar o visualizar datos de diferentes maneras.

Esta aplicación viene a suplir algunas de esas necesidades que no cubren las herramientas actuales, a través de una aplicación web con una interfaz sencilla y un lenguaje claro y conciso, de modo que pueda hacer uso de la misma cualquier usuario sin conocimientos técnicos previos.

La aplicación ha sido desarrollada principalmente mediante el lenguaje de programación Python y haciendo uso del framework Flask, además de otros lenguajes como puede ser HTML o CSS. Las tecnologías utilizadas son actualmente punteras en el mercado laboral y ha sido una de las razones por las que decidí realizar el proyecto con estas tecnologías.

En cuanto al desarrollo de la aplicación, te permite raspar las tablas de datos incrustadas en una página web, seleccionar la tabla entre todas las disponibles y realizar modificaciones o visualizar información relativa a ella antes de descargar la tabla en un fichero con formato CSV.

AGRADECIMIENTOS

Este proyecto pone el punto y final a mi etapa de formación en la Universidad Miguel Hernandez. Lo primero agradecer a todos los docentes que han hecho posible superar esta etapa y demostrarme lo bello que es el mundo de la informática. En especial, agradecer a Jesús Javier Rodríguez Sala por brindarme la oportunidad de ser su alumno en este proyecto y guiarme de la mejor manera posible en esta última etapa formativa.

Tampoco puedo olvidarme de todos los compañeros que me han acompañado durante esta etapa y me han ayudado a superar todos los obstáculos que se han interpuesto en el camino. Me llevo de esta etapa muy buenos amigos, gracias Mathias, Aaron, Jose Juan y Omar por todos los buenos momentos vividos.

Por último, gracias a ti, Paula, gracias por darme fuerzas para seguir en los momentos más duros, esta etapa sin tu fuerza no habría sido posible.

A todos vosotros, gracias.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Capítulo 1
Introducción

1.1.- ORÍGENES DE DATOS ONLINE

En 1989 con la consolidación por parte de Tim Berners Lee del lenguaje para marcado de hipertextos (HTML), el protocolo de transferencia de hipertextos (HTTP), y un programa llamado WorldWideWeb[1] browser, se empiezan a crear las primeras páginas web, ligadas a perfiles académicos, científicos o gubernamentales. En 1993 la web se empezó a desvincular de dichos perfiles al levantar el gobierno de EEUU la prohibición que tenía sobre su uso comercial. Ese mismo año, el CERN entregó de forma gratuita todas las tecnologías desarrolladas y la web pasó a ser de dominio público [2]. Ese año, el número de sitios web existentes era de unos 100, cuatro años más tarde, en 1997, se estima que había más de doscientos mil [3]. En la actualidad se calcula que pueden haber más de 1.800 millones de páginas web [4].

Internet se ha convertido en la mayor enciclopedia del mundo, en muy poco tiempo se tiene acceso a gran cantidad de datos de muy diversas fuentes. Los datos abiertos son datos que cualquiera puede utilizar, reutilizar y redistribuir de forma libre y, como mucho, se encuentran sujetos a las cláusulas de reconocimiento de su origen y/o autoría y a compartirlos de la misma manera que se publican [5].

Estos datos se pueden clasificar en función de sus usos y aplicaciones potenciales, según este criterio, algunas de las categorías de datos abiertos que se pueden encontrar disponibles on-line son [6]:

- Cultural: datos sobre obras o bienes culturales.
- Ciencia: datos generados por la investigación científica en sus diversas ramas (física, biología, medicina, etc.).
- Finanzas: datos sobre mercados financieros (acciones, bonos, divisas etc.), o sobre las cuentas de gobiernos u otras administraciones.
- Estadísticas: datos recopilados y generados por los centros de estudios estadísticos (censo, indicadores socioeconómicos, etc.).
- Tiempo: datos generados y utilizados para tratar de entender la meteorología e intentar predecirla.
- Medio ambiente: datos relacionados con el entorno natural, climatología, niveles de contaminación, calidad de agua de ríos y mares, etc.
- Transporte: datos provenientes de horarios, rutas, etc.

Los datos también se pueden clasificar en función de su fuente u origen del que provienen, es decir, del organismo o entidad responsable de su generación y difusión, a groso modo, se pueden identificar los siguientes orígenes [7]:

- Origen gubernamental o institucional
- Origen privado (empresas, ONGs, etc.)
- Origen personal (ciudadanos que ceden sus datos)

La mayor parte de los datos disponibles en Internet son de origen gubernamental o institucional. Estas instituciones abren sus datos a los ciudadanos para transmitir transparencia o bien para que sean usados a modo de estudio para poder mejorar cualquier ámbito del día a día.

Para que un conjunto de datos sea considerado como "*abierto*", debe cumplir una serie de condiciones [8]:

- Gratuidad: los datos deben poder ser obtenidos de manera gratuita, no deben tener ningún tipo de licencia privativa cómo copyright etc.

- Formato abierto: no deben estar en un formato propietario que dificulte o impida su lectura y utilización (CSV, XML, JSON, etc.).
- Accesibilidad: deben poder ser accesibles por todo tipo de personas, incluidas aquellas que tienen algún tipo de discapacidad.
- No discriminatorio: se debe poder acceder a los datos sin necesidad de aportar ninguna identificación, como usuario y contraseña o email.

Podemos obtener datos online de forma abierta de distintas fuentes o diferentes maneras. Algunos portales que disponen de datos abiertos son:

- Gobierno de España: https://datos.gob.es
- Banco Mundial de Datos: https://datos.bancomundial.org
- Universidad de Alicante: https://datos.ua.es
- Organización Europea para la Investigación Nuclear (CERN): http://opendata.cern.ch
- CIVIO: https://civio.es [9]
- Novagob: https://novagob.org [9]

Como se ha indicado más arriba, el portal de datos abiertos del Gobierno de España se encuentra en la dirección https://datos.gob.es. En dicho portal, a fecha 3 de enero de 2021, hay disponibles un total de 29.397 conjuntos de datos (141.664 distribuciones). En la tabla 1.1 se muestran las categorías y formatos más numerosos dentro de este portal:

Tabla 1.1: Categorías y formatos de los conjuntos de datos de datos.gov.es

CATEGORÍAS		FORMATOS	
Sector público:	6.170	CSV:	14.433
Medio ambiente:	5.990	XLS:	10.458
Sociedad y bienestar:	4.390	JSON:	9.499
Economía:	3.770	HTML:	6.649
Demografía:	3.552	XLSX:	4.910
Cultura y ocio:	2.686	PDF:	4.270
Educación:	2.269	XML:	2.740

La mayoría de los formatos disponibles están listos para ser manejados de forma automática o semiautomática por aplicaciones informáticas para tratamiento de datos, de entre ellos, en la actualidad, los que más habitualmente se utilizan para este fin son los formatos CSV y JSON, seguidos de XML.

El formato CSV (*Comma Separated Values*) es el más utilizado para el intercambio de datos ya que es compatible con la mayoría de aplicaciones, esto hace que sea el más popular entre todos los formatos, se usa para intercambiar datos entre sistemas en texto plano. Normalmente, contiene una fila de encabezado para nombrar las columnas, y a continuación, en las siguientes filas, van los valores (los propios datos) que se quieren

intercambiar. Para separar los valores, se suele utilizar comas (,), aunque también es posible encontrar ficheros de este tipo en los que el separador sea otro carácter como el punto y coma (;), el espacio o el tabulador (a estos últimos en los que el separador es el tabulador también se les llama "*ficheros TSV*", acrónimo de *Tab-Separated Values*). También es posible (aunque menos frecuente) utilizar algún tipo de separador personalizado o convenido entre el origen y el destino de los datos [10].

Otro formato muy conocido es JSON, se trata de un formato de notación de objetos JavaScript que se representa como pares de clave y valor en formato semiestructurado. Su principal utilidad, y por lo que se ha hecho popular, es por su uso en el intercambio de datos en línea. Debido a los servicios web basados en REST, JSON está incluido de forma nativa en la mayoría de lenguajes de programación basados en web [10].

Otro formato muy utilizado y similar al anterior es XML, este lenguaje permite codificar información mediante un conjunto de reglas de modo que sea legible por un ordenador y a su vez por un ser humano. Igual que JSON, se utiliza principalmente para transmitir información entre los servicios web y APIs REST. Una de las principales características de XML es el soporte a Unicode, esto permite escribir la información en cualquier idioma del mundo y que pueda ser leída del mismo modo. [11]

1.2.- WEB SCRAPING

A través de Internet se puede acceder a infinidad de datos, los cuales no siempre están disponibles en un fichero con un formato manejable y listo para descargar (como se ha descrito anteriormente), también hay otras muchas formas de presentar la información.

Semestre 2

Asignatura			Créditos		
Código	Nombre	Tipo	tot	teor	prác
2822	TRABAJO FIN DE GRADO	Trabajo Fin Grado	12	6	6
2816	GESTIÓN FINANCIERA PARA LAS TECNOLOGÍAS DE LA INFORMACIÓN	Optativa	6	3	3
2817	CREACIÓN DE EMPRESAS DE BASE TECNOLÓGICA	Optativa	6	3	3
2818	TÉCNICAS ÁGILES DE DESARROLLO DE SOFTWARE	Optativa	6	3	3
2819	DESARROLLO DE SERVICIOS WEB	Optativa	6	3	3
2820	HERRAMIENTAS DE SOFTWARE LIBRE	Optativa	6	3	3

Figura 1.1.- Captura tabla con las asignaturas de 4º Curso/2º Semestre (Informática/UMH)

La manera más usada de incluir datos en la mayoría de las páginas web es con el uso de tablas, es decir, los datos se disponen en forma de filas y columnas, organizando la información de modo que sea fácil, rápida y sencilla de entender para cualquier tipo de público. Las tablas están compuestas normalmente por una primera fila, denominada fila de cabecera, que tiene la finalidad de mostrar los nombres de la columna (ver ejemplos figuras 1.1 y 1.2).

Datos históricos GOOG

Plazo: Diario | Descargar datos | 03/12/2020 - 03/01/2021

Fecha	Último	Apertura	Máximo	Mínimo	Vol.	% var.
31.12.2020	1.751,88	1.735,42	1.758,93	1.735,42	1,01M	0,71%
30.12.2020	1.739,52	1.762,98	1.762,98	1.725,70	1,31M	-1,09%
29.12.2020	1.758,72	1.789,35	1.790,26	1.756,37	1,30M	-0,98%
28.12.2020	1.776,09	1.748,53	1.790,47	1.748,53	1,39M	2,14%
24.12.2020	1.738,85	1.735,00	1.746,00	1.729,11	346,75K	0,37%
23.12.2020	1.732,38	1.728,11	1.747,05	1.726,48	1,03M	0,52%
22.12.2020	1.723,50	1.734,54	1.736,70	1.712,67	938,35K	-0,91%
21.12.2020	1.739,37	1.712,01	1.740,52	1.699,54	1,83M	0,48%
18.12.2020	1.731,01	1.754,18	1.755,11	1.720,22	4,02M	-0,97%
17.12.2020	1.747,90	1.769,95	1.771,40	1.738,72	1,62M	-0,86%
16.12.2020	1.763,00	1.772,88	1.773,00	1.756,08	1,44M	-0,27%
15.12.2020	1.767,77	1.767,46	1.768,74	1.750,00	1,48M	0,44%
14.12.2020	1.760,06	1.778,58	1.796,36	1.757,51	1,60M	-1,22%
11.12.2020	1.781,77	1.763,06	1.784,45	1.760,00	1,22M	0,36%
10.12.2020	1.775,33	1.769,11	1.780,93	1.745,15	1,36M	-0,49%
09.12.2020	1.784,13	1.818,89	1.834,27	1.768,22	1,51M	-1,89%
08.12.2020	1.818,55	1.809,28	1.821,12	1.796,63	1,10M	-0,05%
07.12.2020	1.819,48	1.820,42	1.831,71	1.805,78	1,32M	-0,47%
04.12.2020	1.827,99	1.824,52	1.833,16	1.816,99	1,38M	0,07%
03.12.2020	1.826,77	1.824,01	1.846,35	1.823,05	1,23M	-0,06%

Máximo: **1.846,35** Mínimo: **1.699,54** Diferencia: **146,81** Promedio: **1.768,20** % var.: **-4,16**

Figura 1.2.- Captura cotización de Google en diciembre de 2020
(fuente: https://es.investing.com/)

Se denomina web scraping (del inglés scrape: arañar/raspar) a cualquier acción dirigida a extraer contenido de forma automática de una o varias páginas web. Su principal función es obtener mucha información sin teclear ninguna palabra ni estar realizando tediosas operaciones de selección, copiado y pegado de contenido. Gracias a los algoritmos de búsqueda, es posible "*raspar*" datos de multitud de páginas webs y extraer aquellos que se puedan necesitar [12].

Al software utilizado para scrapear se le suele llamar bot, spider (araña) o crawler [13]. En la actualidad hay muchas maneras de realizar web scraping, desde herramientas para aquellos usuarios que no tienen conocimiento de programación, hasta librerías para aquellos usuarios profesionales que se dedican a la programación de arañas.

Para aquellos usuarios que desean hacer web scraping pero no tienen conocimientos técnicos, hay diversas herramientas que pueden realizar el raspado de datos, como pueden ser import.io o TableCapture. Estas herramientas suelen estar asociadas a un coste por uso o bien a un uso gratuito de forma limitada. Para aquellos usuarios con capacidades más técnicas, disponen de librerías para realizar la extracción de datos, librerías como Beautiful Soup o Scrapy. La mayoría de las librerías son gratuitas pero requieren el uso de conocimientos de programación. Las herramientas y librerías mencionadas anteriormente, serán tratadas más en profundidad en el capítulo 2.

A continuación se va a detallar los principales usos del web scraping [13]:

- Agregadores de contenido: una de las principales aplicaciones originarias, se usó para reunir noticias u ofertas interesantes en un único sitio web.
- Reputación online: para la realización de estudios sobre la reputación en foros especializados, comentarios en productos o noticias, plataformas de reviews etc.
- Caza de tendencias: tras el estudio de la reputación online, hacer uso del scraping para saber sobre lo que se va a estar hablando en los próximos meses y aprovechar para realizar campañas de marketing sobre ello.
- Optimización de precios: el scrapeo continuo de diferentes páginas webs de compras para poder obtener en tiempo real cual es el mejor precio de mercado de un producto. O en el caso contrario, para poder modificar los precios de venta acorde a los precios de los competidores.
- Monitorización de la competencia: poder alertarnos y controlar las actualizaciones que realizan nuestro competidores en relación a los catálogos, ofertas, actualizaciones de la web etc.
- Optimización e-commerce: a través del scraping poder escoger cual es la mejor imagen a mostrar de un producto destacado o buscar cuál puede ser un nicho libre en el mercado, etc

Un claro ejemplo de uso podría ser que queremos extraer el título de 400 páginas que tienen el mismo formato y se encuentran alojadas dentro del mismo sitio web. Sabemos que el título se encuentra alojado dentro de una etiqueta <*h1*> y que esta está dentro de una etiqueta <*div*> con el atributo *class=header*, o directamente, dentro de una etiqueta <*header*>. Una aplicación araña lo que haría es navegar por todas las páginas web y obtener el valor que está dentro de la etiqueta. A su vez, todas estas etiquetas se pueden guardar en un fichero .csv para poder disponer de un listado de páginas web con sus

correspondientes títulos. Este resultado se puede obtener en apenas unos minutos, cosa que si se hiciera de forma manual, se tardaría muchas más horas [12].

Otro caso de uso de web scraping sería la obtención de todos los valores que aparecen por ejemplo en la tabla 1.2. (ver más arriba) Esta tabla hace relación a la cotización en bolsa de Google, si queremos extraer esos datos y guardarlos en un fichero para un posterior estudio de esta información, se puede crear una araña que extraiga dicho contenido y lo almacene en un fichero, de forma que en apenas unos minutos tendríamos esos valores de forma correcta (sin errores debidos a la extracción manual de datos), en cambio, si un usuario quisiera extraer esos mismos datos por medios no automatizados, no solo tardaría más tiempo, sino que además, existe la posibilidad cometer equivocaciones en la transcripción, error que haría cometer nuevos errores en nuestro estudio.

1.3.- JUSTIFICACIÓN DEL PROYECTO

Como ya se ha comentado, hoy en día se puede acceder a infinidad de datos on-line, pero no siempre esos datos son fáciles de extraer para ser analizados. Una de las mayores dificultades con la que se enfrentan los usuarios al intentar descargar información de Internet, es que no siempre los datos están disponibles para hacer dicha descarga. Muchas veces estos datos se encuentran en formatos que no se pueden descargar fácilmente, como puede ser el caso de la información presentada en forma de tablas, a la hora de obtener esta información, la tarea resulta compleja o, en algunos casos, muy larga y tediosa, ya que la forma más inmediata para hacerlo (para quien no tiene conocimientos técnicos) es mediante operaciones de "*copy-paste*" o, en algunos casos, copiando los valores de uno en uno. Por lo tanto, la tarea de extraer estos datos, con la finalidad de poder procesarlos y/o generar alguna gráfica, para poder incluir dichos resultados en algún estudio o informe, resulta compleja.

Son varios los perfiles de potenciales usuarios de una herramienta que facilite la tarea de poder realizar web scraping sobre datos presentados en tablas en las páginas de un determinado sitio web:

- **Estudiantes**: A nivel personal, una de las dificultades que he tenido durante la carrera a la hora de investigar por Internet y acceder a datos es que muchos de estos no estaban accesibles para descargar y modificar como yo quiera. Frecuentemente, me he encontrado con tablas, en las que, si quería descargar e insertar en el informe, tan solo podía hacerlo mediante una captura de pantalla y posteriormente adjuntar dicha imagen. También me ha ocurrido el caso de querer modificar algunos datos de la tabla para la mejora de la legibilidad y no poder hacerlo ya que esos datos no eran fácilmente personalizables.

- **Periodistas**: He tenido la ocasión de hablar con varios periodistas sobre esta cuestión, ellos también tienen el mismo problema a la hora de acceder a datos. Me han comentado que muchas veces buscaban datos en Internet para poder adjuntar a los artículos, o para poder dar una explicación más gráfica sobre el problema que estaban analizando, y se encontraban que para hacer algunos gráficos, no podían extraer fácilmente la información de la página web, sino que tenían que copiar cada dato y, posteriormente, introducirlos de forma manual para la realización del gráfico.

- **Estadísticos/Investigadores**: Del mismo modo que le ocurre a los periodistas, muchos estadísticos o investigadores, tienen ese mismo problema, la información que disponen para realizar sus investigaciones, está incrustada en tablas web de modo que no pueden acceder fácilmente a ella.

1.4.- OBJETIVOS

En este cuarto apartado, se van a explicar los principales objetivos a cumplir en la realización del proyecto, así como los objetivos secundarios y personales que me he propuesto.

1.4.1.- Objetivos principales

El objetivo principal del proyecto es, dada una URL, poder obtener todas las tablas HTML disponibles en dicha dirección, así como dar la posibilidad de seleccionar una de ellas y poder trabajar con los datos, ya sea viendo estadísticas, poder modificar datos, eliminar aquellos que no estén bien (que presenten valores nulos o incorrectos) y, finalmente, permitir la posibilidad de guardarlos en un fichero con un formato estándar, que sea legible por cualquier aplicación informática. Dado que los datos que se pretende obtener estarán originalmente en formato tabular, se utilizará el formato de fichero CSV para realizar la descarga de los datos extraídos.

Además de poder obtener las tablas de una URL, para aprovechar mejor las opciones de procesamiento de datos de la aplicación, también debe ser posible cargar datos de un fichero CSV en la aplicación, y trabajar de la misma manera que se hace desde una tabla obtenida desde una URL.

Para poder hacer uso de la aplicación, el usuario deberá estar previamente logueado mediante un usuario y una contraseña. También se realizará una traza de logs, de modo que

se sepa que ha hecho cada usuario en cada momento, así como los cambios que ha ido efectuado, dicha traza e información sobre el uso de la aplicación no será pública, sólo podrá ser vista por el administrador de la aplicación, de modo que quedará protegida la información que se obtiene del usuario.

Entre los objetivos principales, también está presente la realización de control de errores, de modo que el usuario no pueda realizar acciones incorrectas como introducir una URL de forma errónea, acceder a información que no tiene permiso o, que la aplicación falle por cualquier otra circunstancia. El sistema será capaz de detectar el error, capturarlo y mostrar al usuario un aviso con el problema detectado.

Finalmente, se deberá proporcionar un manual con dos finalidades, por un lado indicar como realizar la instalación y puesta en funcionamiento de esta aplicación, para que un usuario interesado pueda descargar el software necesario, instalarlo y configurarlo y así, poder usarlo. El manual también incluirá la información técnica necesaria para, en un futuro, poder realizar un mantenimiento de la aplicación, corregir bugs y ampliar su funcionalidad.

1.4.2.- Objetivos secundarios

Entre los objetivos secundarios en la realización del proyecto, está la realización de una interfaz sencilla e intuitiva, de modo que cualquier persona con pocos conocimientos informáticos pueda usar la aplicación. Todo ello con un lenguaje conciso y directo y sin sobrecargar mucho la interfaz para que el usuario no se vea sobrepasado por un exceso de información y su experiencia de uso sea fluida.

1.4.3.- Objetivos personales

En cuanto a los objetivos personales que me he marcado completar, el primero es llevar a cabo la planificación necesaria para el desarrollo de una aplicación web, con su posterior informe y documentación, y el hacer frente a todos los inconvenientes que puedan ocurrir en el transcurso del proyecto.

A nivel más técnico me propongo aplicar los cuatro años de carrera a un proyecto que, aunque de carácter académico, resuelve una problemática real, aplicar las técnicas aprendidas y los lenguajes herramientas utilizados durante la carrera.

También quiero aprender un nuevo lenguaje de programación, Python; nuevo, no porque sea novedoso o moderno, sino porque no se ha estudiado durante la carrera y, en la actualidad, es uno de los lenguajes más demandados. Este aprendizaje de Python incluye

aprender todo lo que lo envuelve, desde la nueva sintaxis, hasta las librerías estándar más interesantes que me permitan desarrollar código los más rápido posible, así como la interpretación de la documentación de las principales funciones y librerías.

También está entre mis objetivos, la posibilidad publicar tanto la aplicación como la documentación técnica, de modo que cualquier persona pueda hacer uso de ella y ayudar en la obtención de datos de la Web.

Capítulo 2 Antecedentes y estado de la cuestión

En este capítulo se va a realizar un estudio sobre las diferentes aplicaciones, librerías o frameworks que hay en la actualidad para la realización de web scraping o "raspado de datos en la web". Se va a dividir en dos apartados, en primer lugar se verán aplicaciones de scraping y a continuación, librerías y frameworks de scraping. Cada uno de estos apartados estará dividido en cuatro subapartados con las herramientas más relevantes y populares que se han encontrado sobre cada una de estas categorías. Finalmente, a modo de resumen, se valorarán las herramientas estudiadas.

2.1.- APLICACIONES DE SCRAPING

En la actualidad, la extracción de datos de Internet es muy importante para muchas empresas que lo aplican para tareas como la realización de estudios de mercado, estudio de la competencia, o extracción de datos para fines de investigación. A pesar de que los lenguajes de programación y sus librerías poco a poco se van simplificando para que

cualquier persona pueda hacer uso de ellas, no siempre todos tienen la capacidad de escribir código. Esto ha derivado en que muchas empresas hayan desarrollado una idea de negocio de forma que permiten a cualquier persona sin conocimientos técnicos en la extracción de datos, la posibilidad de hacerlo mediante alguna de estas aplicación. Estas empresas tratan de simplificar lo máximo posible las herramientas para que la extracción sea lo más sencilla posible para sus clientes.

A continuación se van a detallar, de entre la gran cantidad de aplicaciones que hay en el mercado, aquellas más populares o que tienen alguna característica que destaca por encima de las demás.

2.1.1.- import.io

Import.io es una compañía fundada en 2012 por Matthew Painter, Andrew Fogg y David White [14], su principal función es ofrecer al usuario la extracción de datos web no estructurados y convertirlos en tablas de datos estructurados [15]. La aplicación consiste en un servicio web en el que hay que introducir la dirección URL que contiene la información que deseas extraer y, a través del método "Point-and-click" (apuntar con el ratón y clicar), señalar los elementos deseados para convertirlos en un formato de datos estructurados.

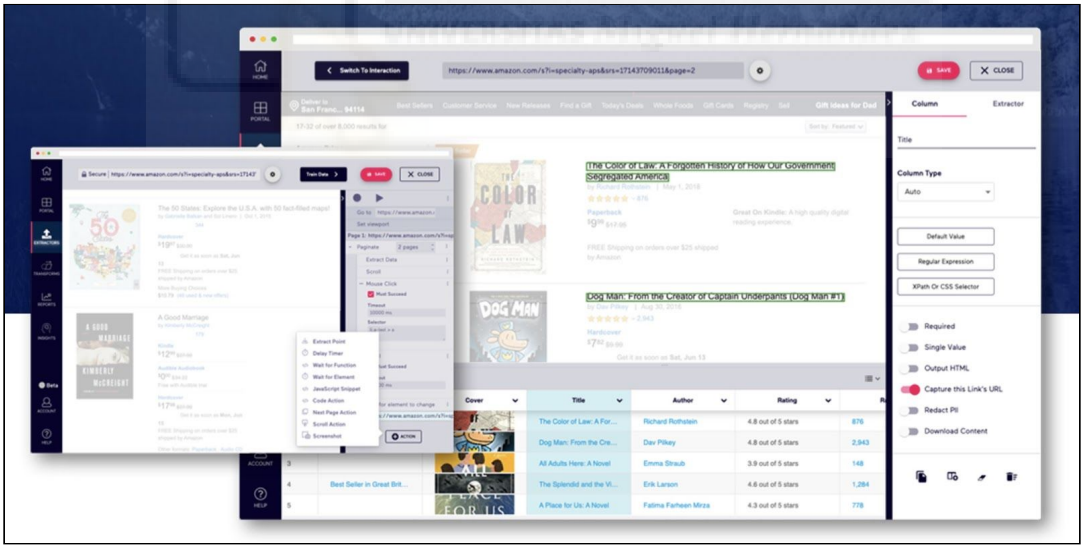


Figura 2.1: Captura de pantalla de la web import.io

Actualmente no acepta el registro de nuevos usuarios para realizar pruebas, pero según se ha podido comprobar en un artículo del portal "*schollofdata*" [16], una vez introducida la URL de nuestro interés, la aplicación muestra una tabla con todos los datos disponibles en dicha página. También da la opción de eliminar columnas o renombrar las existentes. Se ha comprobado que no solo permite la descarga de la página actual, sino que en caso de que la información deseada esté en varias páginas, la aplicación ofrece la opción de buscar en

varias páginas y poder descargar los datos de todas ellas. Una vez seleccionada la información que desea el usuario, esta se puede descargar en diferentes formatos, también ofrece la posibilidad de integrar dichos datos en una API, así como exportarlos a diferentes plataformas (como por ejemplo puede ser Google Sheets, entre otras).

En la actualidad el método de trabajo con la aplicación es mediante una suscripción de pago. Para conocer los precios y las condiciones del servicio hay que contactar con la propia empresa, ya que no ofrecen información al respecto en la web.

2.1.2.- Table Capture

Esta segunda aplicación de scraping es también muy conocida en el mundo de "data scraping". Table Capture está basada en un plugin para Google Chrome o Mozilla y para poder hacer uso de ella debemos instalarla en uno de estos dos navegadores. Table Capture da la posibilidad de capturar fácilmente tablas HTML para usarlas en una hoja de cálculo, importa el contenido de la tabla contenida en una página web a los formatos más habituales de hoja de cálculo (Microsoft Excel, Office 365, Open Office, Google Sheets, etc.) [17].

Para usar la aplicación se debe hacer clic derecho encima de la tabla que queremos extraer y dentro del cuadro nos saldrá una opción para extraer los datos de la tabla seleccionada.

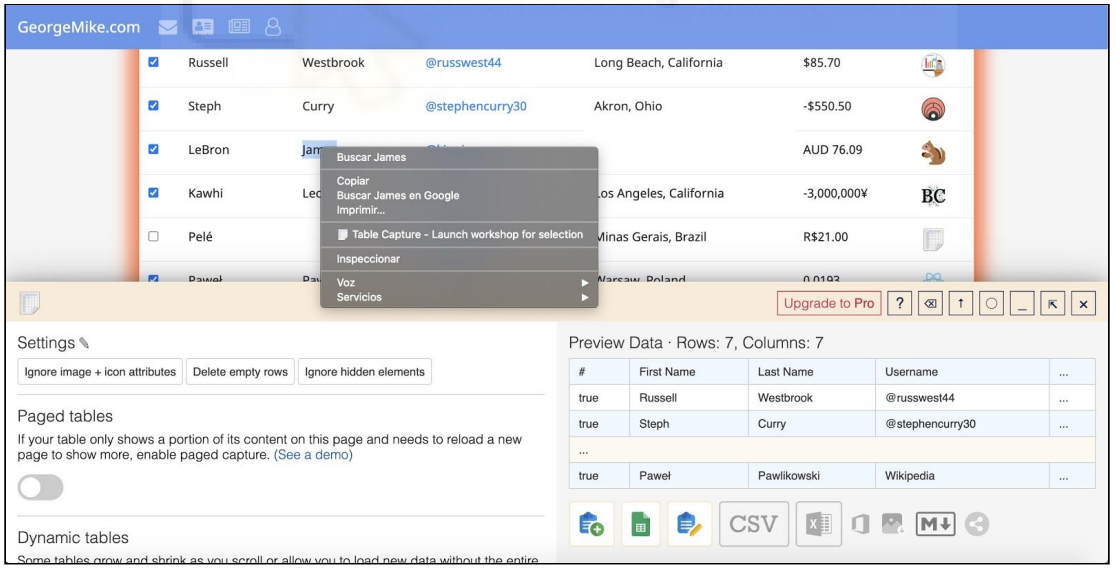


Figura 2.2.- Aplicación Table capture en funcionamiento

Al activar este plugin, se abre una ventana en la parte inferior, la cual estará dividida en dos mitades, la mitad izquierda para la configuración y la parte derecha para la visualización de la tabla que se quiere capturar. La aplicación no permite modificar ni eliminar ninguna columna, ni hacer ninguna operación de edición sobre los datos antes de descargar los datos, solamente permite su descarga tal cual están en la tabla. Da la opción

de descargar los datos en fichero Google Sheets o copiarlos al portapapeles. En caso de querer guardar la información en formato CSV o Excel, establecer el delimitar deseado para el fichero csv, eliminar la publicidad implícita o extraer datos de tablas más complejas, es necesario suscribir un plan PRO, que tiene un coste de 8$ al año.

2.1.3.- Octoparse

Octoparse es otra herramienta para extracción de datos web. Extrae automáticamente datos de páginas y sitios web públicamente visibles en Internet. No es necesario escribir código, por tanto, cualquier usuario sin conocimientos técnicos, puede extraer fácilmente una gran cantidad de información de diversas webs a través de esta aplicación. Los datos que se pueden obtener son archivos HTML, archivos de texto e imágenes como PNG y GIF. Además de guardar datos en el formato de archivo original basado en escenarios comerciales, puede convertirlos a un formato especificado por el usuario y descargarlos. Hay una variedad de posibles formatos de salida, como CSV, Excel, HTML, JSON y bases de datos (MySQL, SQL Server, Oracle) [18].

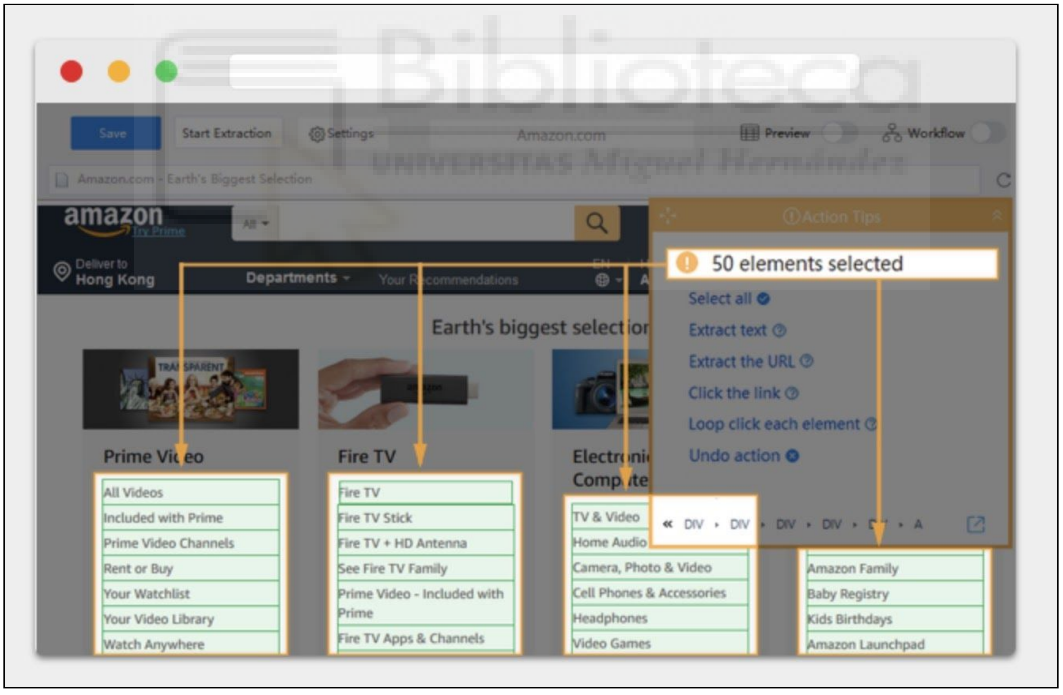


Figura 2.3.- Octoparse en funcionamiento

Entre las funciones más interesantes que tiene esta aplicación está la de dar la posibilidad de programar la fecha y hora específica en la que se desea raspar una determinada web, también realiza una rotación de IP automáticas para evitar ser bloqueado por la web origen de los datos, también puede raspar datos de elementos desplegables así como de páginas accesibles solo después de haber iniciado sesión o páginas que se cargan poco a poco, dinámicamente, utilizando AJAX.

Está basado en el método "point-to-click" (apunta-y-clica), lo que facilita la tarea de raspado a cualquier tipo de usuario, así como la integración de diferentes plantillas para la extracción de datos de webs tales como Amazon, eBay, Groupon, Alibaba, Twitter, Instagram, etc.

Para el uso de esta aplicación, la empresa desarrolladora ofrece un plan gratuito pero sin muchas de las opciones disponibles, como por ejemplo la rotación de IP o las plantillas. Para hacer uso de estas opciones hay que contratar el plan de pago Estándar a un precio de 75$/mes. Para usos más intensivos, la empresa ofrece la posibilidad de planes personalizados desde 4.899$/ año.

2.1.4.- 80legs

80legs es una herramienta de rastreo web, igual que las herramientas anteriores, permite la configuración de requisitos personalizados. Está diseñada sobre todo para la obtención de grandes cantidades de datos, es decir, de millones de registros. Tiene la posibilidad de recolectar, además de los datos, links, keywords, metadatos, etc., así como poder luego almacenar dichos datos en diferentes ficheros en la nube o descargarlos para el usuario en su computadora local.

Una vez introducidos los parámetros necesarios para indicar el origen de datos, la aplicación se encarga de raspar todos estos datos utilizando diversas técnicas para que no pueda ser bloqueado por las paginas web, utiliza técnicas como la inclusión de URLs duplicadas, múltiples IPs, así como limitar la velocidad de rastreo, etc.

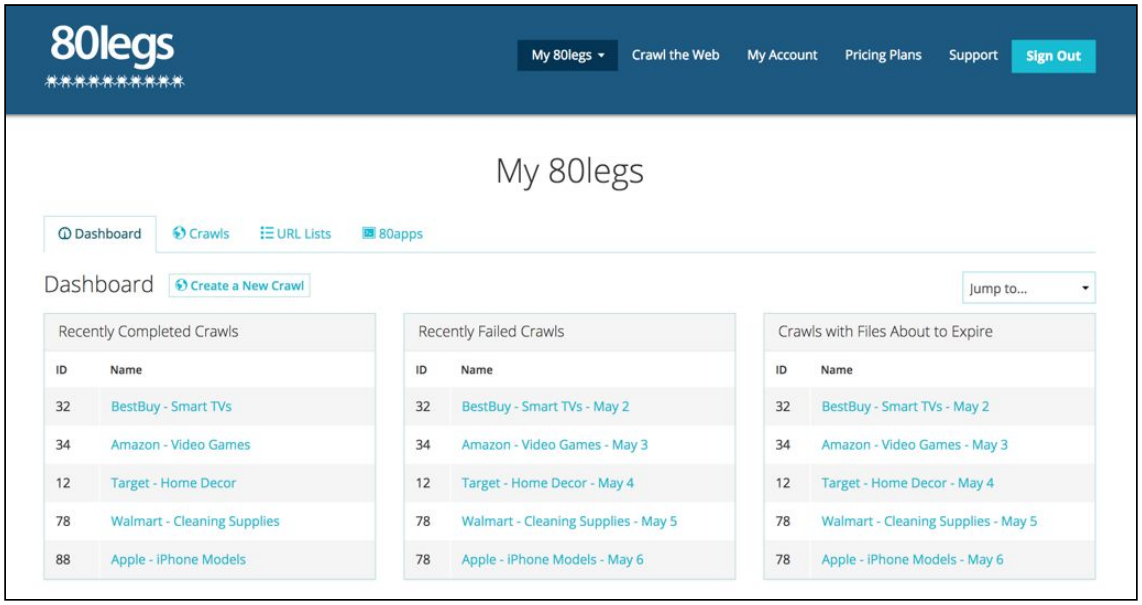


Figura 2.4.- Captura pantalla de la aplicación 80legs [19]

Además, está optimizado de modo que está conectado a una base de datos altamente estructurada denominada "*Datafiniti*", la cual tiene almacenados millones de registros de páginas webs y permite que cuando se quiera acceder a los datos de una página web, si esta está scrapeada por la base de datos, no tiene que volver a realizar el raspado de datos, en vez de eso, puede acceder a la base de datos, permitiendo obtener dichos datos de una forma más rápida y segura [20].

En cuanto a los planes ofrecidos por el fabricante para utilizar esta aplicación, existe un plan gratuito que, a pesar de serlo, exige introducir un número de tarjeta de crédito; este plan permite raspar hasta 10.000 URL's. Además, dispone de 5 planes de pago que van desde los 29$/mes y 100.000 URL's, hasta el plan personalizado para cada empresa con 10.000.000 URL's [21].

2.2.- LIBRERÍAS Y FRAMEWORKS DE SCRAPING

Las aplicaciones de scraping en ocasiones están muy limitadas en lo que se refiere al ámbito profesional y no permiten muchas veces realizar operaciones más complejas o elaboradas que se salen de su repertorio de opciones. Por ello, resulta interesante considerar la alternativa de desarrollar un programa ad hoc para aquello que se desea extraer, si bien, esta tarea es mucho más compleja, y no todas las personas están cualificadas para realizar la extracción de datos desarrollando sus propios algoritmos de scraping, ya que requiere conocimientos de programación que no suelen tener los usuarios convencionales de las herramientas descritas en el apartado anterior. Las personas que sí están cualificadas para este trabajo son los programadores, estos profesionales trabajan directamente con muchas de las librerías creadas para desarrollar, y son los responsables de la implementación de herramientas como las mencionadas en el apartado anterior, pero también serán capaces de diseñar y ejecutar tareas más personalizadas para trabajos de extracción de datos más específicos.

En el mundo de las librerías y frameworks de scraping, igual que pasa en el ámbito de las aplicaciones, hay muchas herramientas a disposición de los desarrolladores, pero no todas tienen detrás una amplia comunidad de usuarios, o una documentación de calidad que las haga populares, y que además, faciliten la tarea de codificar algoritmos y permitan al programador resolver las dudas que le puedan surgir cuando está escribiendo el código de sus aplicaciones. Por ello, en los siguientes subapartados, se van a detallar las tres principales librerías para el raspado de datos, así como uno de los frameworks para populares para esta tarea; en cada apartado también se detallaran algunos ejemplos para ilustrar el funcionamiento de estas herramientas.

2.2.1.- Requests

Requests es una librería de Python que permite enviar solicitudes HTTP automáticamente (sin intervenir manualmente con un navegador web), haciendo que la integración con los servicios web sea mucho más fácil [22]. Según la propia página web, esta librería es utilizada por grandes empresas como Google, Amazon, Paypal entre otras muchas, además de algunas organizaciones Federales de los Estados Unidos de América, que la utilizan de forma interna. Ha sido descargada más de 8.000.000 veces desde PyPI [23].

Al navegar por Internet, constantemente se hacen peticiones HTTP get y/o post a los servidores (lo más frecuente al navegar son las peticiones get). Para ver una página web, el navegador solicita un archivo HTML al servidor web que aloja dicha página para que este pueda representarlo y mostrarlo de manera adecuada. La librería requests emula este tipo de peticiones, obteniendo el contenido de una página web, gracias a esta funcionalidad, esta librería es muy utilizada en el scraping de datos, ya que normalmente lo que se hace es guardar la página web en formato HTML y sobre esa web, trabajar el raspado de datos.

Para usar esta librería lo primero que hay que hacer es instalarla en nuestro equipo con el comando pip (package installer for Python). Una vez instalada correctamente hay que insertar en el código las instrucciones que se muestran en la figura 2.5.

```
# línea de comando (consola) para instalar request
    C:\path> pip install requests

# Código Python para descargar una web
    import requests
    response = requests.get('https://www.umh.es')
```

Figura 2.5.- Instalación de request y descarga de la página www.umh.es

Además de las peticiones get, de forma análoga, requests permite realizar peticiones post indicando en la instrucción correspondiente, además de la URL de la página solicitada, los parámetros que serán adjuntados y enviados en el cuerpo de la petición al servidor. También es posible usar requests para realizar más tipos de peticiones HTTP (por ejemplo: put, delete, etc.) [24].

```
# Código Python para importar librería request
# y hacer peticiones post, put y delete
  import requests
  resp=requests.post('https://www.umh.es', data={'key':'value'})
  resp=requests.put('https://www.umh.es', data={'key':'value'})
  resp=requests.delete('https://www.umh.es/delete')
```

Figura 2.6.- Código para realizar peticiones post, put y delete

2.2.2.- Beautiful Soup

Beautiful Soup es otra librería de Python para analizar documentos HTML (incluyendo los que tienen un marcado incorrecto), y suele utilizarse conjuntamente con la librería anterior para realizar scraping. Beautiful Soup crea un árbol con todos los elementos del documento que puede ser utilizado para extraer información [25]. La figura 2.7 muestra como instalar, importar y un ejemplo de uso de esta librería en el que se extrae el título de la página web de la Universidad Miguel Hernández.

```
# línea de comando (consola) para instalar
beautifulSoup
    C:\path> pip install beautifulsoup4

# Código Python para descargar una web
    import requests
    from bs4 import BeautifulSoup
    response = requests.get('https://www.umh.es')
    soup = BeautifulSoup(response.text,'html.parser')
    # obtiene el título de la web descargada
    soup.title

# Salida por pantalla
    <title> Universidad Miguel Hernández </title>
```

Figura 2.7.- Instalación de beautifulsoup4 y descarga del título de una página web

Esta librería proporciona todos los métodos necesarios para el raspado de un fichero HTML. Se suele trabajar conjuntamente con la librería requests, esta obtiene el documento HTML y Beautiful Soup se encarga de extraer la información. Para el uso adecuado de la librería hay que tener nociones de HTML y CSS para saber como acceder a ciertas partes del documento. Si no se tienen se tienen estos conocimientos de HTML, se puede hacer uso de la herramienta mediante una ruta de selector o una ruta XPath.

También se pueden realizar búsquedas o filtros, para ello la librería dispone de métodos para buscar elementos en el árbol del documento HTML. Dos de los principales métodos son "find_all()" y "find()", ambos métodos buscan entre los hijos o descendientes un objeto de tipo tag (como pueden ser div, h1, span, etc) y recopila todos aquellos que cumplan una serie de condiciones [26].

Esta librería es especialmente útil para descargar webs estáticas, ya que hay ciertas web que trabajan con frameworks Javascript y se cargan de forma dinámica mediante la ejecución de dicho código en el navegador. Con estas herramientas se obtendrá el script que se ejecuta en el navegador, y no el contenido HTML.

2.2.3.- Selenium

La librería Selenium puede tomar el control de un navegador web. Es una herramienta bastante potente que permite interactuar con el navegador web como lo haría un humano [27]. Para hacer uso de esta librería, igual que las dos anteriores, primero es necesario instalarla en el sistema. Esta librería tiene un requisito adicional, además de instalar la propia librería, hay que instalar los drivers que permiten controlar al navegador, estos drivers son válidos para Chrome, Edge, Firefox y Safari [28].

Esta biblioteca está pensada sobre todo para la realización de test y validar que todo funcione de forma correcta. También se puede usar para realizar acciones que deben seguir un determinado procedimiento, como por ejemplo, introducir un login automáticamente. En la figura 2.8 se muestra como hacer un login automático en Facebook [29].

```
# línea de comando (consola) para instalar selenium
  C:\path> pip install selenium

# Código Python para descargar una web
  from selenium import webdriver
  from selenium.webdriver.common.keys import Keys

# Ruta donde guardamos chromedriver.exe
  bro = webdriver.Chrome('C:\path\chromedriver.exe')
  bro.get('https://facebook.com')
  time.sleep[5]

# Datos que extraemos de la página inspeccionando elemento
  username = bro.find element by id("email")
  password = bro.find_element_by_id("pass")

# Cambiar las credenciales
  username.send keys("tu correo")
  password.send_keys("tu_contraseña")

# Emula el hacer click en "Iniciar Sesión"
  log=bro.find_element_by_xpath("//*[@type='submit']")
  log.submit()
```

Figura 2.8.- Instalación de selenium y acceso automático al login de Facebook

Con esta librería se solventa el problema que tienen las herramientas para webs estáticas como Beautiful Soup, ya que da igual que la página web se cargue dinámicamente con un script Javascript, puesto que esta librería recopila los datos una vez se ha cargado la página y puede hacer esperar un tiempo desde que se inicia la carga para confirmar que la página se ha cargado correctamente. Esta funcionalidad de control del navegador permite hacer cosas como estudiar la compatibilidad de un sitio web con varios navegadores, de forma

que todos ellos hagan las mismas pruebas, de la misma manera, y siguiendo los mismos pasos para confirmar que el funcionamiento es correcto en todos los navegadores.

2.2.4.- Scrapy

A diferencia de las tres herramientas anteriores, Scrapy no es una librería sino un framework, uno de los más versátiles, utilizados para la creación de arañas web. Su arquitectura basada en Pipelines, Schedulers, Spiders y Downloaders permite al desarrollador tener un control completo sobre todo el proceso de Scraping [30]. Aunque el objetivo principal de Scrapy es extraer datos de páginas web, también se puede utilizar para extraer datos mediante API, para obtener la estructura de la web o simplemente como un extractor general. Para poder hacer uso de esta herramienta se deberá descargar e instalarla desde su web oficial [31]. También existe el servicio en la nube Scrapy Cloud [32], donde se pueden ejecutar procesos de descarga y almacenamiento de webs.

Scrapy también dispone de una tubería integrada para procesar los datos extraídos. La apertura de páginas en Scrapy se realiza de forma asincrónica, es decir, se pueden descargar varias páginas al mismo tiempo. Por ello, Scrapy es una buena opción para proyectos que necesitan procesar una gran cantidad de páginas.

A continuación se va a ilustrar con un ejemplo (como en los casos anteriores) como extraer el título de la web de la Universidad Miguel Hernández. Primero se ejecuta el comando scrapy en el terminal, con la URL de la página a descargar (figura 2.9), esto inicia el proceso de descarga, y a continuación abre otra consola en la que ya se pueden introducir los comandos que exploran y extraen información del documento descargado (figura 2.10).

```
# línea de comando (consola) para ejecutar scrapy
    C:\path> scrapy shell 'https://www.umh.es'
```

Figura 2.9.- Ejecución de scrapy en la consola

```
# línea de comando (consola) para obtener el título
    C:\path> response.css(«title::text»).extract()

# salida del comando anterior
    # ['Universidad Miguel Hernández']
```

Figura 2.10.- Extracción del título de una página web

2.3.- VALORACIÓN

Para concluir, debido a mi conocimiento de lenguajes y librerías de Python, considero mejor abordar el problema haciendo uso de esas librerías y frameworks, me van a permitir poder personalizar más la aplicación final, que en el caso de usar tecnologías clásicas para desarrollo de páginas webs. Además, las librerías son gratuitas y con ellas no hay ningún límite en cuanto a la cantidad de dominios o páginas que se pueden raspar datos.

El principal problema de raspar los datos de tablas se va abordar haciendo uso de las librerías Request y Beautiful Soup. Con la librería Request se realizarán las peticiones al servidor y se recopilará cierta información. Con la librería Beautifulsoup se podrán raspar los datos y obtener solo las tablas que hay en la página web.

Con estas dos librerías se espera poder abordar el problema de modo que se pueda raspar cualquier tabla, independiente del formato y estilo que presenten. En el caso de haber datos vacíos, también será posible reconocerlos y representarlos sin ningún problema.

Capítulo 3 Hipótesis de trabajo

3.1.- TECNOLOGÍAS DEL LADO DEL CLIENTE

En el desarrollo de aplicaciones web, los lenguajes de programación del lado del cliente son utilizados para la creación tanto de contenidos estáticos, como contenidos dinámicos cuando se utilizan metodologías como AJAX, y el propio cliente solicita y procesa datos obtenidos desde el servidor. Lo más usual es la inclusión de estos datos en un documentos HTML. El funcionamiento de este lado es sencillo, cuando el servidor envía los datos al cliente, este último solo tiene que presentar el resultado en el navegador web. A veces, los datos contienen instrucciones de cómo reaccionar ante cierta acción llevada a cabo por el usuario, como puede ser el caso de un clic en un botón. Al ser ejecutado en el navegador web, el usuario es capaz de ver el código fuente de lo que se está mostrando, a diferencia de lo que ocurre en el lado del servidor, cuyo código ya no es accesible para el cliente [33].

3.1.1.- HTML

HTML es un lenguaje creado en 1991 por Tim Berners-Lee, el considerado padre de la World Wide Web (WWW). En el momento de su creación, el lenguaje tan solo tenía una pocas etiquetas utilizadas para escribir documentos. La mayor revolución que tuvo este lenguaje fue la posibilidad de crear enlaces entre documentos, esto permitió que se pudiera navegar entre las páginas [34].

La primera versión de HTML contenía una serie de etiquetas que tenían que comenzar con "<" seguido del nombre de la etiqueta y cerrando como el signo ">". A su vez, estas etiquetas pueden contener parámetros denominados atributos. Algunas de las primeras etiqueta fueron [35]:

- <TITLE> … </TITLE>: se refiere al título de la página.
- <A NAME=xxx HREF=xxx> … </A>: esta etiqueta se utiliza para enlazar o apuntar a otras páginas.
- <PLAINTEXT>: esta etiqueta, actualmente en desuso, se empleó para representar todo el texto siguiente, hasta el final del fichero, como texto plano sin formato.
- <P>: indica un nuevo párrafo.
- <H1>, <H2>, <H3>, <H4>, <H5>, <H6>: seis niveles de encabezamiento.
- <UL>, <LI>: se utilizan para la creación de listas de ítems.

La organización reguladora de los estándares de la web es W3C (World Wide Web Consortium) y es la encargada de la publicación de las diferentes versiones de HTML. Se han publicado versiones como 3.2 en enero de 1997, 4.0 en abril de 1998. La última versión, HTML5, se publicó en 2007 [34].

3.1.2.- CSS (Cascading Style Sheets)

CSS es un lenguaje de estilo que tiene como objetivo definir el formato y maquetación en los documentos HTML. Mientras HTML define el contenido, CSS formatea dicho contenido. Abarca opciones como las fuentes, márgenes, altura, anchura, colores, imágenes de fondo entre otras muchas más opciones [36]. En la actualidad CSS forma parte del W3C, que se ha encargado de realizar las versiones CSS2 y CSS3. Actualmente está en desarrollo la versión CSS4, pero aún no se ha lanzado como versión oficial [37].

Hay varias formas para aplicar los estilos CSS a un documento HTML [38]:

- En línea: Utilizando el atributo "style" para definir los estilos de un elemento HTML dentro de la propia etiqueta HTML.

- Interno: definiendo estilos para ciertos elementos del documento entre las etiquetas <style>...</style> en el propio documento HTML.

- Externo: enlazando con la etiqueta <link … /> un fichero de definición de estilos, externo al documento HTML, pero incluido por este con esta etiqueta.

3.1.3.- JavaScript

JavaScript (denominado también como Javascript o JS) es un lenguaje de programación ligero, interpretado, y con funciones de primera clase (las propias funciones pueden ser pasadas como argumento a otras funciones). Este lenguaje es especialmente conocido por ser un lenguaje de scripting para páginas webs [39] (actualmente también tiene otros usos).

Este lenguaje corre en el lado del cliente web y permite implementar complejas secuencias en páginas webs, animaciones de gráficos 2D/3D, actualizaciones de contenido, mapas interactivos etc. Se puede incluir código JS en una web de varias formas [40]:

- Insertando el código Javascript entre las etiquetas <script>...</script> en propio documento HTML.

- Usando las mismas etiquetas <script src="codigo.js"></script> y el atributo “src” para enlazar un fichero externo con código Javascript.

- Insertando fragmentos de código JavaScript en línea, dentro de las etiquetas HTML, asociando dicho código a ciertos atributos de dichas etiquetas relacionados con eventos que el usuario puede realizar como “onclick”, “ondblclick”, “onmouseover”, etc.

3.1.4.- Bootstrap [41]

Bootstrap es un framework CSS desarrollado por Twitter a mediados de 2010. Se ha convertido en unos de los frameworks front-end más famosos en el mundo del open source. Fue conocido como “Twitter Blueprint” y sirvió como guías de estilo para la herramientas desarrolladas por la empresa. En agosto de 2011, se publicó de forma libre y continua hasta hoy. Desde su liberación se han publicado 20 actualizaciones, además de 2 reescrituras mayores v2 y v3. Con la versión 2 se añadió la funcionalidad de que el framework fuera “responsive” (adaptativo), y con la versión 3 se amplió esta característica para hacerla viable en dispositivos móviles. La última versión publicada es la 4, de nuevo se ha reescrito el proyecto de modo que se han incluido las nuevas propiedades de CSS así

como nuevas dependencias y tecnologías para los navegadores web más modernos [16]. Para hacer uso de este framework en una página web hay dos opciones:

- Descargar los ficheros CSS compilados y minificados, e incluirlos en el código de la página web con la etiqueta <link… />.

- Usar BootstrapDNS para, con tan solo una línea de código, incluir el framework en el proyecto, dicha línea es la que se muestra en la figura 3.1, y habrá que ponerla en la sección <head>...</head> de la página web.

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/
css/bootstrap.min.css"
integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0
E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous"/>
```

Figura 3.1.- Código para incluir Bootstrap 4 en una web

3.1.5.- AJAX

AJAX (Asynchronous JavaScript and XML) es una metodología que utiliza llamadas asíncronas al servidor a través del objeto XMLHttpRequest de JavaScript, y utiliza XML como formato para el intercambio de datos. Estas dos tecnologías trabajando conjuntamente permiten que las aplicaciones webs procesen las solicitudes que realizan al servidor en segundo plano, es decir, de forma asíncrona [42].

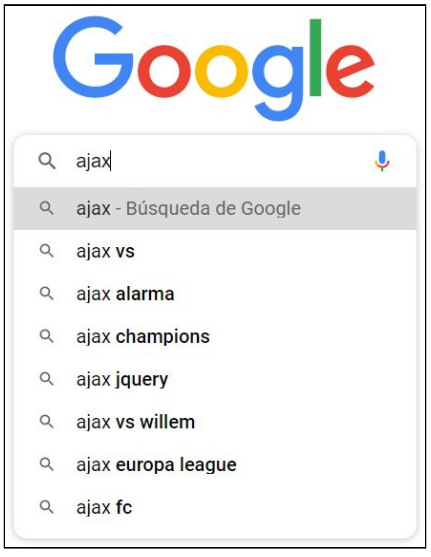


Figura 3.2.- Ejemplo de búsqueda en Google, el desplegable se genera usando AJAX

El principal valor que aporta el uso de AJAX consiste en hacer que las aplicaciones web se comporten de forma similar a como lo hacen las aplicaciones de escritorio, es decir, enviar y recibir información del servidor sin la necesidad de tener que recargar los elementos estáticos de una página (al revés de como lo hacían las aplicaciones web clásicas).

3.1.6.- JSON [43]

JSON es un lenguaje de ligero formato para el intercambio de datos. Se trata de un lenguaje simple de interpretar y de generar para las máquinas, de la misma manera que es fácil de leer para los humanos. El formato de JSON, es independiente del lenguaje que se use, y ampliamente reconocido por las comunidades como ser un lenguaje ideal para el intercambio de datos.

JSON está construido por dos estructuras fundamentales:

- El objeto: Es una colección de pares clave/valor, donde la clave es un string y "valor" es cualquier tipo de dato válido de JSON. Todos los elementos de un objeto (la colección) irán entre llaves ({...}) y separados por comas (,); entre una clave y su valor asociado se pondrá el carácter "dos puntos" (:) como separador.

- El array: Consiste en una lista ordenada de valores, cada uno de los cuales será, con en el caso anterior, cualquier tipo de dato válido de JSON. Los elementos de un array van entre corchetes ([...]) y separados por comas.

Tanto los objetos como los arrays de JSON son considerados "valores" por la propia sintaxis, por lo que es posible anidar objetos dentro de arrays y viceversa.

```
[
  {
    "name": "Molecule Man",
    "age": 29,
    "powers": ["Radiation resistance","Turning tiny"]
  },
  {
    "name": "Madame Uppercut",
    "age": 39,
    "powers": ["Super strength","Damage resistance"]
  }
]
```

Figura 3.3.- Ejemplo datos en formato JSON [44]

3.1.7.- JQuery

JQuery es una librería de JavaScript creada por John Resign y lanzada por primera vez en 2006. Se utiliza principalmente para manejar de forma sencilla eventos, animaciones y, en general, manipular documentos. Es compatible con la mayoría de navegadores y se ha convertido en una de las librerías de JavaScript más usada y con una gran comunidad de desarrolladores que, a lo largo del tiempo, han ido generando infinidad de plugins para esta librería (hablaremos de algunos de ellos en los siguientes apartados) [45][46]. La última versión publicada es la 3.5.1 y para incorporarla a un proyecto, se debe de incluir en la sección <head>...</head> de nuestro fichero html la instrucción de la figura 3.4.

```
<script
      src="https://code.jquery.com/jquery-3.5.1.min.js">
</script>
```

Figura 3.4.- Código para incluir JQuery 3.5.1 en una web

3.1.8.- DataTables

DataTables es un plugin "open source" de JQuery que permite introducir en la web tablas con las que poder realizar funciones tales como buscar, ordenar o paginar los resultados de forma rápida y sencilla. En definitiva, permite agregar funciones avanzadas a cualquier tabla HTML [47]. Entre las opciones más avanzadas de este plugin se tiene [48]:

- Paginación, creación de una barra de navegación de páginas.
- Búsqueda instantánea filtrando elementos.
- Ordenamiento por columnas, y también por múltiples columnas a la vez.
- Compatible con diferentes orígenes de datos (DOM, JavaScript, AJAX)
- Personalizable incluyendo temas de DataTables, Bootstrap, Foundation etc.
- Responsive, también a dispositivo móvil.
- Se traduce fácilmente a otros idiomas.

```
<link rel="stylesheet"
href="https://cdn.datatables.net/1.10.23/css/jquery.da
taTables.min.css">

<link rel="stylesheet"
href="https://cdn.datatables.net/1.10.23/js/jquery.dat
aTables.min.js">
```

Figura 3.5.- Código para incluir DataTables en una web

Para hacer uso de este plugin hay que incluir dos ficheros en la web (ver figura 3.5) que son los que incorporan la funcionalidad que se ha mencionado previamente. Para convertir una tabla HTML ordinaria en una tabla dotada con el estilo y la funcionalidad descrita es necesario que dicha tabla disponga de la propiedad "id" definida (por ejemplo algo como esto: <table id="tab1">...</table>) y ejecutar el script que se muestra en la figura 3.6.

```
$(document).ready( function () {
    $('#tab1').DataTable();
} );
```

Figura 3.6.- Código para dar funcionalidad DataTables a una tabla HTML

Figura 3.7.- Ejemplo de aplicación de DataTable [49]

3.1.9.- FileInput

FileInput es un plugin, construido especialmente para su uso con Bootstrap, para configurar un selector/cargador de ficheros avanzado de manera simple. Mejora la funcionalidad de un selector "input", ofreciendo la posibilidad de previsualizar una gran cantidad de ficheros y otros archivos como videos, fotos o audios. Además permite la inclusión de AJAX para la subida de archivos, funcionalidad "drag-and-drop" (arrastrar y

soltar ficheros), así como la posibilidad de incluir una barra de progreso y visualizar el tamaño del fichero o ficheros seleccionados (entre otras muchas opciones). Su instalación es algo compleja, aunque en la referencia bibliográfica se describen los pasos necesarios de forma detallada [49].

Para incluir un elemento FileInput en una aplicación web, basta con declarar un objeto <input> de HTML como se indica el la figura 3.8.

```
<input id="nomInput" name="nomInput" type="file"
class="file" data-browse-on-zone-click="true">
```

Figura 3.8.- Código correspondiente al <input> con FileInput

La figura 3.9 muestra la llamada JavaScript necesaria para activar el elemento <input> y dotarlo de toda la funcionalidad de FileInput.

```
$("#nomInput").fileinput();
```

Figura 3.9.-Activación de FileInput a partir de un elemento <input>

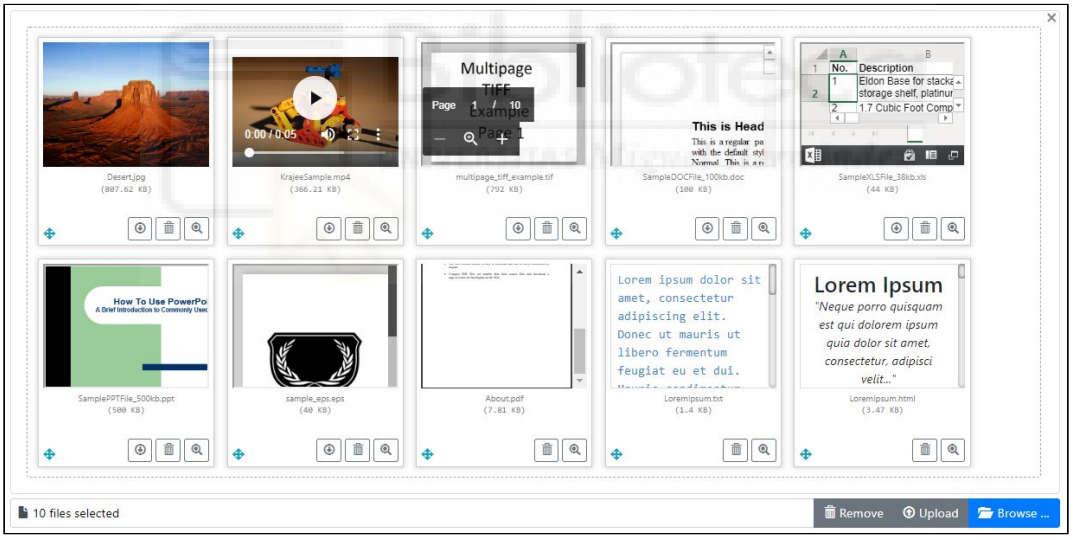


Figura 3.10.- Ejemplo de visualización de FileInput [50]

3.2.- TECNOLOGÍAS EN EL LADO DEL SERVIDOR

Las tecnologías del lado del servidor son las encargadas de dar respuesta a las solicitudes realizadas por el cliente (solicitar un recurso, extraer datos, procesarlos e integrarlos en el proyecto web), y todo ello de manera opaca para el usuario. En los primeros tiempos de la red, la programación en el servidor se realizaba exclusivamente con programas o scripts en lenguajes como Perl, líneas de comando (CMD) o C. Estas tecnologías de servidor han ido evolucionando, y en la actualidad existen diversos lenguajes de programación para este

propósito como pueden ser PHP, el lenguaje por excelencia para servidores web, o Python, Java, ASP.NET entre otros [33].

3.2.1.- Python

Python es un lenguaje de programación interpretado, interactivo y orientado a objetos, admite diferentes paradigmas de programación como pueden ser el orientado a objetos, procedural o funcional. Es un lenguaje potente (amplia variedad de paquetes y funcionalidades) y con una sintaxis sencilla. Además, es portable, corre en la mayoría de sistemas (Unix/Linux, MacOS y Windows). Es una tecnología "open source" y desde la versión 2.1 pertenece a la organización independiente sin ánimo de lucro "Python Software Foundation", que es la encargada de hacer avanzar la tecnología y publicitar su uso [51].

El autor creador de este lenguaje fue Guido van Rossum, que empezó a trabajar en Python a finales de 1980 en el Instituto Nacional de Matemáticas y Computación en los Países Bajos. El autor se basó en un proyecto anterior llamado "ABC", recogió las partes que más le gustan de él, e introdujo algunas características adicionales que le parecieron adecuadas para desenvolverse más cómodamente. En Python hay disponibles dos maneras de escribir módulos, directamente en el propio lenguaje, o escribiendo el módulo en C, e incorporándolo al proyecto que se esté desarrollando [52].

Algunas de las características más importantes del lenguaje son [53]:

- Fuertemente tipado (con funciones específicas para hacer "casting").
- Tipado dinámico.
- Filosofía "Incluye baterías" (librería estándar incluida rica y versátil)
- "*Zen*" de Python, 20 principios de software para el diseño del lenguaje:
    - Bello es mejor que feo.
    - Explícito es mejor que implícito.
    - Simple es mejor que complejo.
    - La legibilidad cuenta.
    - Los errores nunca deberían dejarse pasar silenciosamente.
    - (etc...)

```
print("Hola Mundo");
```

Figura 3.11.- Ejemplo "hola mundo" en Python (fichero: "holamundo.py")

Se puede ejecutar un programa en Python desde la línea de comandos llamado al intérprete de lenguaje (normalmente denominado '*python*' o '*py*') seguido del nombre del fichero que contiene el programa o script que se quiera ejecutar, que tendrá extensión '.*py*'.

```
C:\path> python holamundo.py
```

Figura 3.12.- Ejecución de programa en la consola desde línea de comandos

En el capítulo 2 se analizaron varias librerías y frameworks para hacer scraping, entre ellos "Request" y "Beautiful Soup", que son dos librerías de Python que se utilizan en este proyecto. A continuación se van a describir otras librerías de Python, no directamente relacionadas con la funcionalidad de scraping, pero que también se han utilizado en este proyecto.

3.2.1.1.- Pandas

El desarrollo de Pandas lo inició AQR Capital Management en 2008, y a finales de ese mismo año ya hubo una versión open source disponible. Hasta 2015 ha sido mantenido por la comunidad de software, y a partir de ese año Pandas se convirtió en un NumFOCUS project, es decir, pasó a ser parte de esta asociación, asegurando el desarrollo correcto y el mantenimiento del proyecto [54].

Pandas es una librería para Python que proporciona estructuras de datos diseñadas para trabajar con datos relacionales o etiquetados. Su principal objetivo es la construcción de un bloque de alto nivel para el análisis de datos en tiempo real. Pandas acepta los siguientes tipos de datos [55]:

- Datos organizados en columnas heterogéneas como en una consulta SQL o una hoja de Excel.
- Series de datos ordenados o no ordenados como puede ser una estructura con clave/valor.
- Datos estructurados en matrices con etiquetas en filas y columnas.
- Cualquier otra forma de datos estadísticos.

Una ventaja de Pandas es que los datos no deben estar etiquetados o estructurados de una manera específica para ser colocados dentro de una estructura de pandas. Dichas estructuras pueden ser de los siguientes tipos [56]:

- Series: arrays unidimensionales con indexación, parecido a un diccionario.
- DataFrame: estructuras de datos similares a una tabla de una base de datos.
- Panel, Panel4D y PanelND: estructuras de datos de más de 2 dimensiones (muy poco utilizadas).

Para hacer uso de esta librería hay que importarla en el script o código del programa. Se suele usar el nombre "pd" como un convenio de la comunidad de desarrolladores como alias abreviado para nombrar esta librería.

```
# importación de la librería pandas
import pandas as pd
```

Figura 3.13.- Importación de la librería Pandas

3.2.1.2 Conector MySQL

El conector MySQL permite a un programa en Python acceder a bases de datos MySQL usando una API. Entre otras, el conector MySQL tiene las siguientes características[57]:

- Realiza conversión de valores entre Python y MySQL (p.e.: fecha y hora).
- Compresión de protocolo, comprime el flujo de datos entre cliente y servidor.
- Conexiones usando socket TCP/IP.
- Permite conexiones seguras utilizando SSL.
- Controlador autónomo, no requiere la biblioteca de cliente MySQL.

La figura 3.14 muestra como importar el conector y como hacer un uso básico del mismo en un programa en Python [58] [59].

```
# Comando de consola para instalar mysql.connector
C:\path> pip install mysql.connector

# Uso del conector en Python
import mysql.connector
cnx = mysql.connector.connect(user='nom', password='pass',
      host='127.0.0.1', database='tfg')
...
cnx.close()
```

Figura 3.14.- Instalación del conector, crear y cerrar conexión con base de datos

3.2.1.3 MatPlotLib [60]

Matplotlib es una librería para la generación de gráficos en 2D para Python. Tiene incorporados valores predeterminados así como funciones que permitirán la creación de gráficas de una manera sencilla, como por ejemplo, la función "*plot()*". Para generar una gráfica tan solo hay que definir los datos, dependiendo del tipo de gráfica que se quiera visualizar, y realizar una llamada a la función específica que hace dicha gráfica, pasando como parámetros ciertos valores como el color de línea, el grosor, etc. El único requisito fundamental de esta librería es que los datos deben estar estructurados bajo la librería NumPy [61].

```
# Comando de consola) para instalar matplotlib
C:\path> pip install matplotlib

# Código Python para descargar una web
import matplotlib.pyplot as plt
a = [1, 2, 3, 4]
b = [11, 22, 33, 44]
plt.plot(a, b, color='blue', linewidth=3, label='línea')
plt.legend()
plt.show()
```

Figura 3.15.- Instalación y ejemplo de uso de matplotlib

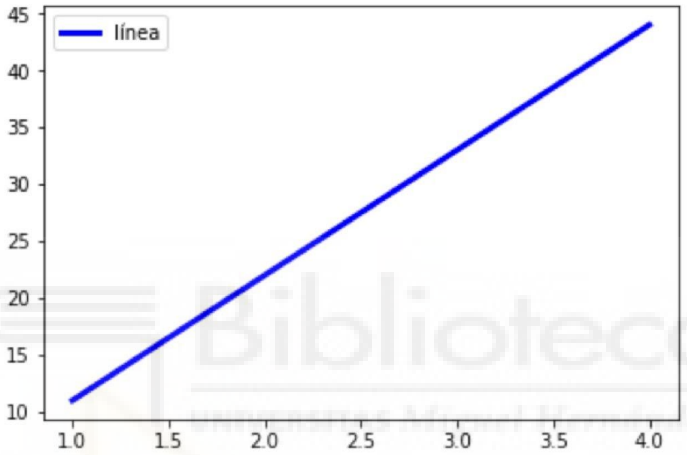


Figura 3.16.- Gráfica resultado del código de la figura 3.15 [62]

3.2.1.4 Time [63]

Time es una librería estándar para Python que ya viene incluida con la instalación básica (no es necesario instalarla aparte), y proporciona un conjunto de funciones para el trabajo con fechas y horas. Además de poder obtener la hora del huso horario que se desee, se pueden convertir fechas y horas, validar o aplicar formatos específicos o incluso detener una ejecución durante un tiempo determinado.

A continuación se desarrolla un ejemplo de como la fecha 4-3-2017 19:10:01 se puede especificar de tres modos distintos:

- Numero: 1488651001.7188754 segundos (desde 1 de enero de 1970 hasta la fecha)
- Cadena: “Sat Mar 4 19:10:01 2017”.
- Objeto struct_time: time.struct_time(tm_year=2017, tm_mon=3, tm_day=4 tm_hour=18, tm_min=10, tm_wday=5, tm_yday=63, tm_isdst=0).

Este módulo tiene la restricción de ser un sistema de 32 bits y por lo tanto sólo admite fechas desde 1 de enero de 1970 a las 0 horas hasta el año 2038. Para poder obtener una fecha/hora hay que llamar al módulo "time" y pasar como argumentos los datos que se quiere obtener (ver ejemplo figura 3.17).

```
import time

print("gmtime():" , time.gmtime())
cad = time.strftime("%d-%m-%Y %H:%M", time.gmtime())
print("cad:", cad)

# salida del programa
gmtime(): time.struct_time(tm_year=2021, tm_mon=2,
          tm_mday=7, tm_hour=10, tm_min=3, tm_sec=38,
          tm_wday=6, tm_yday=38, tm_isdst=0)
cad: 07-02-2021 10:03
```

Figura 3.17.- Ejemplo uso de la librería time

3.2.1.5 base64 [64]

Esta librería también está disponible con la instalación básica de Python (no hay que instalarla aparte), y permite codificar y decodificar en el formato "base64". Convierte una cadena en bytes y de esta manera se tiene un sistema para que no se corrompa la información mientras está siendo transferida o procesada. El formato base64 es un formato que se compone de 26 letras en mayúscula, 26 en minúscula, los números del 0 al 9 y los símbolos "+" y "/". A cada valor le corresponde una combinación en binario así como un índice (se puede consultar la tabla de combinaciones en [65]).

```
import base64

cadena = "Base64 en Python"
bytes = cadena.encode("ascii")
bytes_b64 = base64.b64encode(bytes)
cadena_b64 = bytes_b64.decode("ascii")
print("Cadena original  :", cadena)
print("Cadena codificada:", cadena_b64)

# salida del programa
Cadena original  : Base64 en Python
Cadena codificada: QmFzZTY0IGVuIFB5dGhvbg==
```

Figura 3.18.- Ejemplo de uso de la librería base64

3.2.2.- Base de datos MySQL [66]

MySQL es un sistema gestor de bases de datos relacionales de código abierto cuyo modelo es cliente/servidor. Fue fundado en 1994 por la empresa sueca MySQL AB, más tarde Sun Microsystems la adquirió en 2008 hasta que finalmente, en 2010 Oracle compró Sun Microsystems y finalmente MySQL forma parte de Oracle.

Como se acaba de decir, MySQL trabaja sobre bases de datos relacionales, que son estructuras donde los datos están almacenados y organizados en tablas y cada tabla puede estar relacionada con otras mediante claves. A su vez, también utiliza el lenguaje SQL para definir estas estructuras, así como para manipular los datos que contienen.

El funcionamiento habitual de MySQL es muy sencillo:

- En el servidor se crea una base de datos definiendo las relaciones entre tablas.
- El cliente realiza una solicitud utilizando el lenguaje SQL.
- El servidor devuelve al cliente la información solicitada en la solicitud.

3.2.3.- El Framework Flask

Flask es denominado como un "*microframework*" para crear aplicaciones webs. En un framework convencional, en el propio proyecto ya está listo para utilizar un manejador de migraciones, de rutas, una estructura para la conexión a una base de datos, un gestor de autenticación etc. Lo interesante del concepto de microframework, como es el caso de Flask, no se tiene nada de todo esto, se trata de un lienzo en blanco en el que se irán añadiendo módulos y librerías según vayan haciendo falta. Una ventaja que tiene frente a un framework convencional es que solo se integra en el proyecto aquello que se necesita y por lo tanto se tiene más control sobre el código. Pero a su vez se pierde tiempo en la configuración de aquellos elementos que se necesitan [67].

Flask sigue el patrón MVC (Modelo-Vista-Controlador), se trata de un patrón que utiliza tres componentes (el modelo, la vista y el controlador), para separar la responsabilidad que tiene cada uno de ellos, de modo que si se realiza un cambio en alguno de ellos, no afecte al resto de componentes y, de esta manera, que haya una mayor independencia en la implementación de cada uno de ellos [68].

En este patrón, el modelo es la capa donde se trabajan con los datos, contiene mecanismos para el acceso a la información. Los datos, normalmente se encontrarán en una base de datos, por lo tanto esta capa será la encargada de realizar las sentencias de insertado, extracción, actualización o eliminación sobre la base de datos. La vista contiene el código

que va a producir la generación de la interfaz del usuario, en el caso de una aplicación web, el código HTML. Por último, el controlador es la capa que sirve de enlace entre las vistas, el modelo y el usuario. Responde a los mecanismos que requieren el resto de componentes para implementar las necesidades de la aplicación, se encarga de hacer de enlace entre las otras dos capas [69].

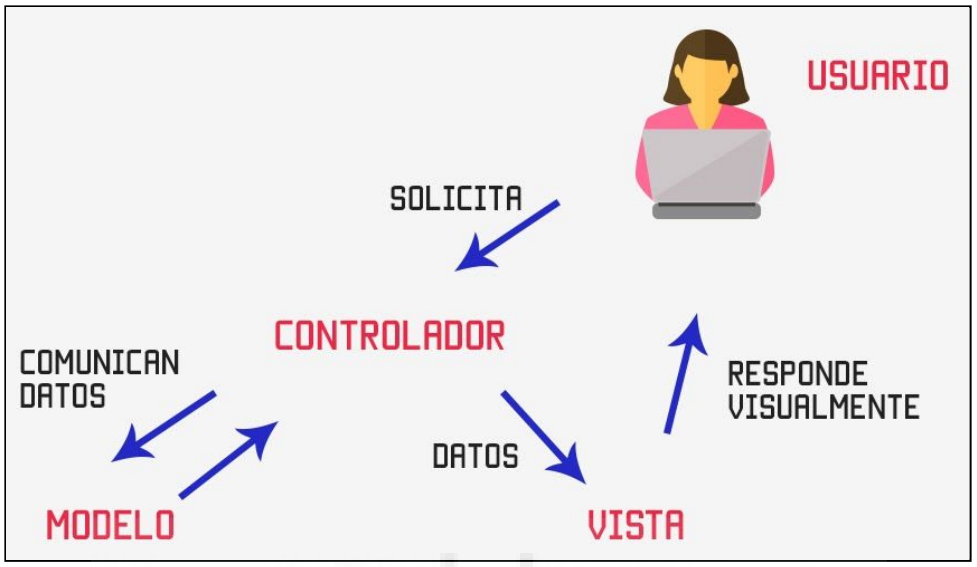


Figura 3.19.- Esquema del patrón MVC

Flask no es considerado un framework "full stack" puesto que no vienen instaladas todas las funcionalidades, a pesar de que se pueden instalar a posteriori, pero a pesar de eso tiene numerosas características interesantes[70]:

- Como "microframework" es muy válido para aquellas aplicaciones que no requieren muchas extensiones, y permite desarrollar código de forma rápida y ágil.
- Incluye un servidor web integrado en la propia herramienta, por lo que no es necesaria otra infraestructura con esa funcionalidad para ir probando la aplicación.
- Contiene un depurador de código y soporte de pruebas unitarias.
- Es compatible con Python3.
- Compatible con wsgi, protocolo utilizado por ciertos servidores webs, específico para servir a aplicaciones escritas en Python.
- Incorpora un controlador que recibe las peticiones realizadas por los usuarios y determina a qué recurso se está accediendo.
- Soporta de forma nativa el uso de cookies seguras.
- Soporta el uso de variables de sesión.
- A pesar de no tener integrado un ORM, se puede implementar con una extensión
- Permite la construcción de servicios webs como por ejemplo APIs REST o aplicaciones de contenido estático.
- Es Open Source y está bajo la licencia BSD.
- Dispone de una gran comunidad y buena documentación en la web.

En la figura 3.20 se ilustra como hacer un simple “Hola Mundo” con Flask, en 9 líneas de código se tiene implementado este programa así como su configuración del servidor (en otros frameworks sería mucho más costoso) [67]:

```
# Aplicación “Hola mundo” con Falsk
from flask import Flask
app = Flask(__name__)

app.route('/')
def holaMundo():
     return 'Hola Mundo'

if __name__ == '__main__'
     app.run()
```

Figura 3.20.- Ejemplo de “Hola Mundo” en Flask

Flask permite trabajar utilizando un entorno virtual o “virtualenv”. Se trata de una herramienta usada para crear un entorno Python aislado, este entorno no comparte bibliotecas con otros entornos virtuales o con las bibliotecas instaladas en el propio ordenador o servidor, tiene sus propios directorios [71].

Como se ha comentado anteriormente, Flask es un lienzo en blanco en el cual se van incluyendo librerías, a continuación se van a detallar aquellas librerías que se han utilizado en el proyecto

3.2.3.1. - Librería render_template [72]

Render_template es una función para Flask para generar una salida desde una plantilla que está en una carpeta con todas las plantillas. Su principal uso es el de completar la plantilla con los datos que se han generado y de esta manera generar una salida lo más completa posible. Las figuras 3.21 y 3.22 muestran, respectivamente, el código en Python y HTML de un ejemplo de uso:

```
# Ejemplo de direccionamiento con render_template
from flask import render_template

return render_template("ejemplo.html",
                       titulo="Titulo del ejemplo",
                       contenido="Contenido del ejemplo")
```

Figura 3.20.- Ejemplo de render_template

```
<html>
      <head>
            <title>{{ titulo }}</title>
      </head>
      <body>
            <p>{{ contenido }}</p>
      </body>
</html>
```

Figura 3.21.- Código correspondiente al ejemplo de la figura 3.20

3.2.3.2. - Librería request [73]

Los clientes web realizan peticiones al servidor, esté a su vez lo almacena en un objeto llamado request. Con esta librería se puede saber el tipo de petición realizada desde el cliente, ya sea POST, GET, etc. Para poder acceder al tipo de petición, hay que escribir un script como el de la figura 3.22, que devolverá el tipo de petición realizada ("request.method").

```
import requests
response=requests.get('https://www.umh.es')
```

Figura 3.22.- Ejemplo request

En el caso de una petición GET, los argumentos son del tipo clave:valor, para poder acceder a dicha información hay que hacer como en la figura 3.23.

```
import requests
resp=requests.post('https://www.umh.es',
                   data={'key':'value'})
resp=requests.put('https://www.umh.es',
                  data={'key':'value'})
resp=requests.delete('https://www.umh.es/delete')
```

Figura 3.23.- Ejemplo request para una llamada GET con parámetros

3.2.3.3. - Librería send_file [74]

Flask, además de devolver respuestas del tipo HTML, también puede dar otro tipo de respuestas, como por ejemplo, un fichero para su descarga. Para devolver el contenido de un fichero se necesita la librería send_file así como su función implícita send_file. Se debe pasar como parámetro el fichero que se desea devolver (véase la figura 3.24).

```
# Ejemplo de envío de ficheros con send_file
from flask import send_file

return send_file("data.csv")
```

Figura 3.24.- Ejemplo del envío del fichero “data.csv”

3.2.3.4. - Librería session [75]

Esta librería permite crear variables de sesión y de esta manera almacenar información entre cada una de las peticiones que se realicen a la aplicación. El funcionamiento de esta librería consiste en la creación de una cookie con la información que se desee almacenar dentro de ella. Su funcionamiento es muy sencillo como se puede ver en el ejemplo de la figura 3.25.

```
# creación de una variable de sesión
from flask import session

session["variable"] = "valor"
```

Figura 3.25.- Creación de una variable de sesion

3.3.- El IDE PyCharm

PyCharm, de la compañía JetBrains, es un IDE (Integrated Development Environment) para programar en Python, es decir, es una aplicación cuya principal función es la de integrar todas las funciones necesarias para un programador, con el fin de facilitar la tarea de desarrollar aplicaciones, en este caso, en Python, mejorando la productividad. Se distribuye bajo una licencia de código abierto [76] [77]. Algunas de las funcionalidades más destacadas que tiene Pycharm son[78]:

- Asistencia inteligente a la codificación: proporciona finalización de código inteligente, indicación de errores así como refactorizacion de codigo.

    - Editor de código inteligente: ofrece compatibilidad con lenguajes como Python, JavaScript, CSS, TypeScript, etc.
    - Navegación inteligente por el código: con un click se puede saltar a la clase, método o implementación.
    - Refactorización rápida y segura: permite eliminar, introducir variables entre otras opciones de manera segura y teniendo en cuenta el lenguaje.

- Herramientas de desarrollo integradas: herramientas preconfiguradas, ejecutores de pruebas, terminal integrado, herramientas para bases de datos, terminal SSH entre otras:

    - Depuración, pruebas y generación de perfiles: permite depurar código en Python y JavaScript así como una interfaz gráfica para la pruebas.
    - VCS, Despliegue y Desarrollo remoto: incluye interfaz unificada para Git, SVN, Mercurial entre otros sistemas de control de versiones. Permite ejecutar y depurar la aplicación en máquinas remotas.
    - Herramientas para bases de datos: permite acceder a bases de datos directamente desde el IDE y a su vez ejecutar código SQL.

- Desarrollo web: ofrece soporte para lenguajes de desarrollo web como HTML/CSS, AngularJS, Node.js entre otros:

    - Marcos de trabajo web Python: ofrece compatibilidad con Frameworks como Django o Flask entre otros.
    - JavaScript y HTML: Incluye depurador JavaScript de forma integrada en el IDE.
    - Live Edit: permite visualizar de forma instantánea en el navegador los efectos que se realizan sobre el código.

- Herramientas científicas: integra IPython Notebook así como la compatibilidad con Anaconda o paquetes científicos como Matplotlib o NumPy:

    - Consola Python interactiva: el propio IDE incluye una consola.
    - Soporte para pila científica: compatibilidad con bibliotecas científicas como Pandas, Numpy o Matplotlib entre otras, así como la visualización de gráficas.
    - Integración con Conda: permite crear ambientes Conda.

- IDE personalizable y multiplataforma: disponible en las plataformas más populares como Windows, MacOS o Linux, así como la posibilidad de personalización completa del IDE.

    - Interfaz de usuario personalizable: permite personalizar atajos así como la interfaz al gusto del programador.
    - Complementos: compatibilidad con más de 50 completos de IDE 's distintos así como la compatibilidad con complementos de Visual Studio Code.
    - IDE multiplataforma: disponible en las plataformas más usadas en la actualidad como Windows, MacOS y Linux.

Capítulo 4
Metodología y resultados

Todo proyecto de desarrollo de software está basado en algún ciclo de vida del software, en este capítulo se va a describir como ha sido la planificación del proyecto (ciclo de vida, etapas, temporización), y a continuación se describirán los detalles acerca de los requisitos, el diseño y desarrollo de la aplicación

4.1.- PLANIFICACIÓN DEL PROYECTO

El presente proyecto es un trabajo plenamente académico, la aplicación desarrollada no está destinada a ninguna empresa o institución concreta, sino que es el resultado de implementar una serie de requisitos proporcionados por mi director de proyecto, con quien, a través de sucesivas reuniones, se han ido determinando y puliendo las características de la aplicación final.

4.1.1.- Ciclo de vida

Se podría decir que para el desarrollo del presente proyecto se ha seguido un ciclo de vida parecido a SCRUM[79], el cual es una metodología de trabajo ágil cuyo objetivo es la entrega de versiones en cortos periodos de tiempo de forma que permita al cliente trabajar de manera conjunta con el equipo desarrollador de la aplicación y haya una retroalimentación entre el cliente y el responsable del proyecto. Entre las personas que intervienen en un proceso SCRUM existen tres roles fundamentales:

- Product Owner: se trata del responsable del proyecto, es el encargado de hablar de manera constante con el cliente y debe tener conocimientos sobre el negocio. Solo puede haber un rol de este tipo en el proyecto.

- Scrum Master: es el encargado de que las técnicas sean aplicadas de manera eficiente en el proyecto, también se encarga de resolver los posibles problemas ocurriendo en la fase *sprint*.

- Equipo de desarrollo: se trata de un equipo multifuncional y auto-organizado y se encargan de desarrollar el proyecto y las tareas propuestas por el *Product Owner*

En el caso particular de este proyecto, se puede decir que esta metodología se ha aplicado de forma laxa, ya que no se han definido plazos concretos y rigurosos para las reuniones ni había un equipo de desarrollo de varias personas que tuviera que estar coordinado (un TFG es un trabajo individual). Por tanto, se puede decir que el papel de "cliente" ha sido desempeñado por mi director de proyecto, mientras que los tres roles técnicos, orientados a la coordinación y desarrollo del proyecto los he desempeñado yo mismo (en el anexo 1 hay una descripción más detallada de como debe ser un proceso SCRUM y sus principales características).

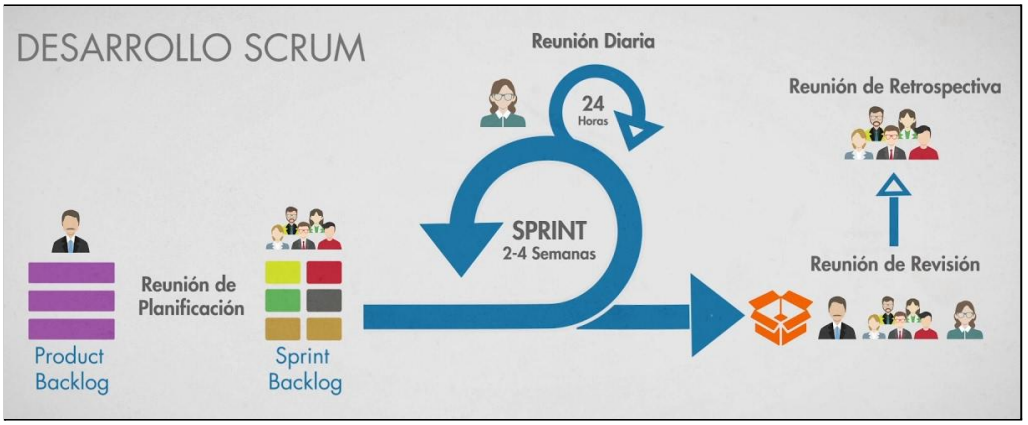


Figura 4.1 Esquema de la metodología SCRUM [80]

4.1.2.- Etapas y plazos del proyecto. Diagrama de Gantt

El proyecto se ha desarrollado en 20 semanas, entre finales de 2020 y principios de 2021. En el diagrama de Gantt de la figura 4.2 se pueden ver las etapas o tareas de que se ha compuesto el proyecto, y la duración o momentos en los cuales tuvieron lugar.

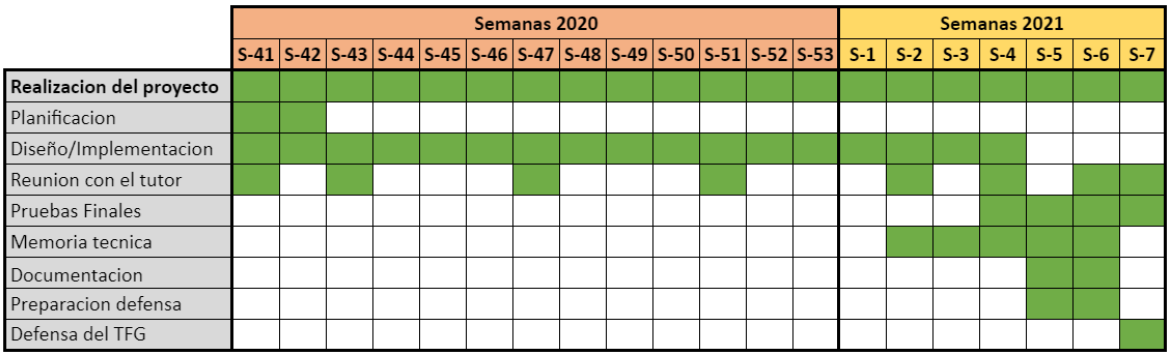


Figura 4.2: Diagrama de Gantt

La etapa concreta de "Diseño/Implementación", la más larga del proyecto, incluye todas las tareas relacionadas con la especificación, diseño y desarrollo del proyecto, a lo largo de dicha etapa, han habido diversos momentos en los que me he reunido con mi director de proyecto (ver etapa "Reuniones con el tutor") en los que le mostraba el trabajo realizado y tomaba nota de sus observaciones y sugerencias, así como las especificaciones que debía complir la aplicación final.

4.2.- CAPTURA DE REQUISITOS

4.2.1.- Jerarquía de actores

Esta aplicación va a funcionar con dos tipos de usuarios, un administrador y un usuario registrado.

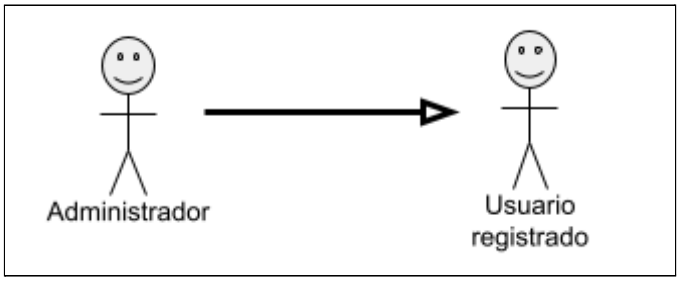


Figura 4.3.- Usuarios y relación de herencia

El usuario registrado es el usuario común de la aplicación, que podrá usar toda su funcionalidad para hacer scraping de tablas de datos incrustadas dentro de páginas web. El administrador también podrá hacer uso de toda esa funcionalidad (ver relación de herencia en la figura 4.3), pero además tendrá a su disposición ciertas funciones para gestionar el uso que se hace de la aplicación. En las tablas 4.1 y 4.2 se describen con un poco más de detalle estos roles (los casos de uso que ahora solo se enumeran por su código, se describen en detalle más adelante).

Tabla 4.1.- Rol "usuario registrado"

Actor	*Usuario registrado*
Descripción	El usuario registrado es el usuario principal que usa la aplicación, tendrá disponible las funcionalidades de buscar por URL o introducir fichero, así como los siguientes pasos en que se corresponden la aplicación. Podrá hacer uso de la aplicación siempre que su cuenta no esté baneada.
Casos de uso	C.U.1, C.U.2, C.U.3, C.U.4, C.U.5, C.U.6, C.U.7, C.U.8, C.U.9, C.U.10, C.U.11, C.U.12, C.U.13, C.U.14, C.U.15, C.U.16 (*) Todos estos casos de uso de describen detalladamente en el apartado 4.2.2

Tabla 4.2.- Rol "Administrador"

Actor	*Administrador*
Descripción	El administrador podrá hacer todo lo que puede el usuario registrado, pero además, tiene todos los permisos de la aplicación y puede realizar la gestión de usuarios así como la gestión de los logs.
Casos de uso	C.U.1, C.U.2, C.U.3, C.U.4, C.U.5, C.U.6, C.U.7, C.U.8, C.U.9, C.U.10, C.U.11, C.U.12, C.U.13, C.U.14, C.U.15, C.U.16, C.U.17, C.U.18, C.U.19, C.U.20, C.U.21, C.U.22, C.U.23, C.U.24, C.U.25, C.U.26 (*) Todos estos casos de uso de describen detalladamente en el apartado 4.2.3

4.2.2.- Actor: usuario registrado

Como ya se ha comentado, el usuario registrado es quien va a hacer el uso principal para el que se ha diseñado la aplicación, que no es otro que el de poder hacer scraping sobre tablas de datos en páginas web para, posteriormente, poder realizar ciertas tareas básicas de manipulación y preprocesamiento de dichos datos y descarga a un fichero CSV. El usuario también podrá subir a la aplicación los datos de un fichero CSV para manipularlos. En la figura 4.4 se muestra el diagrama de casos de uso de este actor y en las tablas sucesivas (desde la tabla 4.3 hasta la tabla 4.18) se describe con detalle cada uno de esos casos de uso.

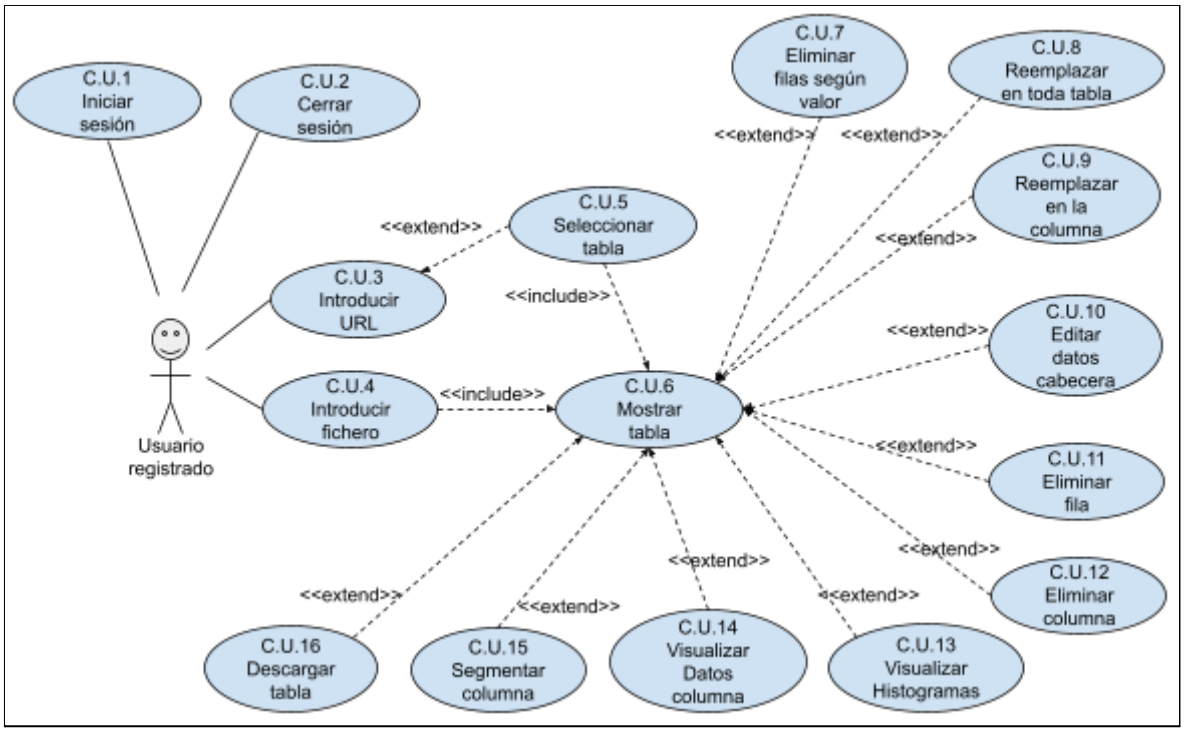


Figura 4.4.- Casos de uso del usuario registrado

Tabla 4.3.- C.U. 1: Iniciar sesión

C.U. 1	*Iniciar sesión*
Actores	Cualquier usuario
Descripción	El usuario inicia sesión en la aplicación, identificándose con su usuario y contraseña, para poder hacer uso de la misma.
Dependencias	
Precondición	
Secuencia normal	P1 - El usuario introduce sus credenciales (id y contraseña) en el formulario web de la página inicial (index) de la aplicación. P2 - Pulsar el botón “Acceso”.
Poscondición	Si el usuario y la contraseña es correcta, dará acceso a la parte privada de la aplicación, mostrando las opciones disponibles para el usuario dependiendo de su rol (usuario registrado o administrador).
Excepciones	P2- Si el usuario y la contraseña no son correctas se indica con un mensaje y se vuelve al formulario de identificación. P2- Si la cuenta del usuario ha sido baneada por el administrador, se mostrará una página informando de dicho estado y no podrá acceder a la parte privada de la aplicación.
Comentarios	

Tabla 4.4.- C.U. 2: Cerrar sesión

C.U. 2	*Cerrar sesión*
Actores	Usuario registrado (y administrador)
Descripción	El usuario cierra su sesión y sale de la parte privada de la aplicación, nuevamente al formulario de identificación.
Dependencias	
Precondición	El usuario previamente debe estar correctamente logueado.
Secuencia normal	P1 - Pulsar la opción “Cerrar sesión” del menú principal.
Poscondición	La aplicación cerrará la sesión del usuario y redirigirá a la página de login (C.U. 1).
Excepciones	
Comentarios	

Tabla 4.5.- C.U.3: Introducir URL

C.U. 3	*Introducir URL*
Actores	Usuario registrado (y administrador)
Descripción	El usuario introduce una URL para realizar el raspado de datos e identificar las tablas que contenga dicha dirección web..
Dependencias	C.U.1
Precondición	El usuario debe estar logueado previamente
Secuencia normal	P1 - Pulsar la línea de texto de “Introducir URL”. P2 - Introducir la URL. P3 - Pulsar el botón de “Enviar”. P4 - La aplicación muestra las tablas disponibles en la web para su selección por parte del usuario.
Poscondición	La aplicación realizará el raspado de la página introducida y mostrará las tablas disponibles en la página web de la URL indicada.
Excepciones	P3 - En caso de no introducir ninguna URL y pulsar el botón “Enviar”, se mostrará una alerta indicando que se debe completar el campo. P4 - Si la web indicada no contiene tablas, no se muestran opciones de selección.
Comentarios	

Tabla 4.6.- C.U. 4: Introducir fichero

C.U. 4	*Introducir fichero*
Actores	Usuario registrado (y administrador)
Descripción	Permite al usuario desde la página principal introducir un fichero “CSV” para realizar la visualización y modificación de datos así como la selección del separador del fichero.
Dependencias	C.U.1
Precondición	El usuario debe estar logueado previamente.
Secuencia normal	P1 - Hacer click en el apartado del fichero. P2 - Arrastrar el fichero dentro del recuadro. P3 - Seleccionar la opción de “Separador predeterminado” o “Separador personalizado”. P4 - Seleccionar o introducir el separador del fichero. P5 - Pulsar el botón “Enviar” (salta al C.U.6).
Poscondición	La aplicación mostrará el C.U.6 en el que permitirá realizar operaciones con la tabla
Excepciones	P3 - En caso de seleccionar un separador personalizado, se deberá introducir el separador o saldrá una alerta indicando que el campo debe ser rellenado. P4 - Solo permite ficheros con formato CSV, en caso de introducir otro tipo de ficheros, estos saldrán como “no disponible”.
Comentarios	El usuario puede seleccionar si es separador es predeterminado o personaliza, cuando se elija uno, el otro tipo quedará deshabilitado.

Tabla 4.7.- C.U. 5: Seleccionar tabla

C.U. 5	*Seleccionar tabla*
Actores	Usuario registrado (y administrador)
Descripción	La aplicación permite la selección de una tabla entre todas las tablas raspadas de la URL introducida en el C.U.3.
Dependencias	C.U.3
Precondición	El usuario debe haber introducido previamente una URL correcta que contenga tablas.
Secuencia normal	P1 - El usuario selecciona una tabla de las disponibles. P2 - Pulsar el botón de “Enviar” para pasar al siguiente paso (C.U.6).
Poscondición	La aplicación pasará al siguiente paso con la tabla seleccionada.
Excepciones	
Comentarios	En caso de no seleccionar una tabla no estará habilitado el botón de enviar.

Tabla 4.8.- C.U. 6: Mostrar tabla

C.U. 6	*Mostrar tabla*
Actores	Usuario registrado (y administrador)
Descripción	Permite al usuario realizar modificaciones en los datos que ha introducido previamente desde una tabla contenida en una URL o desde un fichero CSV.
Dependencias	C.U.4 o C.U.5
Precondición	Debe haber seleccionado una tabla o debe haber cargado un fichero correctamente (casos de uso 4 y 5 respectivamente).
Secuencia normal	P1 - La aplicación carga la tabla P2 - La aplicación muestra la tabla convertida con DataTables
Poscondición	La aplicación mostrará la tabla seleccionada junto con todas las opciones para realizar modificaciones sobre los datos que dicha tabla contiene.
Excepciones	
Comentarios	

Tabla 4.9.- C.U. 7: Eliminar fila según valor

C.U. 7	*Eliminar fila según valor*
Actores	Usuario registrado (y administrador)
Descripción	Permite eliminar todas aquellas filas que tienen el valor introducido en una de sus filas.
Dependencias	C.U.6
Precondición	Previamente, debe haberse cargado una tabla para la modificación de sus datos.
Secuencia normal	P1 - Pulsar la opción “Eliminar fila según valor” del menú. P2 - Introducir el valor a buscar en las filas. P3 - Pulsar el botón “Enviar”.
Poscondición	La aplicación eliminará aquellas filas que contengan en una de sus celdas el valor introducido.
Excepciones	P3 - Si el valor introducido no se encuentra en ninguna de las celdas de la tabla, no se borra ninguna columna.
Comentarios	Si el usuario no introduce ningún dato, se eliminará aquellas filas que tengan datos vacíos. Si el usuario hace clic en “Cerrar” o pulsa fuera de la ventana, se cerrará la ventana manteniendo los datos inalterados.

Tabla 4.10.- C.U. 8: Reemplazar en toda la tabla

C.U. 8	*Reemplazar en toda la tabla*
Actores	Usuario registrado (y administrador)
Descripción	Permite reemplazar en toda la tabla valores, de forma que reemplazará el valor a buscar por el valor a reemplazar.
Dependencias	C.U.6
Precondición	Previamente, debe haberse cargado una tabla para la modificación de sus datos.
Secuencia normal	P1 - Pulsar la opción “Reemplazar en toda tabla” del menú. P2 - Introducir el valor a buscar. P3 - Introducir el valor a reemplazar. P4 - Pulsar el botón “Enviar”.
Poscondición	La aplicación buscará y reemplazará en toda la tabla el valor introducido a buscar por el valor introducido a reemplazar
Excepciones	P4 - Si el valor introducido en P2 no se encuentra en ninguna de las celdas de la tabla, no ocurrirá ningún reemplazo.
Comentarios	En caso de no introducir ningún valor (tanto en P2 como en P3) se interpretará como cadena vacía; buscará o reemplazará valores vacíos. Si el usuario hace clic en “Cerrar” o pulsa fuera de la ventana se cerrará la ventana manteniendo los datos.

Tabla 4.11.- C.U. 9: Reemplazar en la columna

C.U. 9	*Reemplazar en la columna*
Actores	Usuario registrado (y administrador)
Descripción	Permite al usuario reemplazar valores en la columna seleccionada.
Dependencias	C.U.6
Precondición	Previamente, debe haberse cargado una tabla para la modificación de sus datos.
Secuencia normal	P1 - Pulsar la opción “Info” de la columna que se quiere reemplazar. P2 - Introducir el valor a buscar. P3 - Introducir el valor a reemplazar. P4 - Pulsar el botón “Enviar”.
Poscondición	La aplicación buscará y reemplazará en la columna el valor introducido a buscar por el valor introducido a reemplazar-
Excepciones	P4 - Si el valor introducido en P2 no se encuentra en ninguna de las celdas de la tabla, no ocurrirá ningún reemplazo.
Comentarios	En caso de no introducir ningún valor (tanto en P2 como en P3) se interpretará como cadena vacía; buscará o reemplazará valores vacíos. Si el usuario hace clic en “Cerrar” o pulsa fuera de la ventana se cerrará la ventana manteniendo los datos.

Tabla 4.12.- C.U. 10: Editar datos cabecera

C.U. 10	*Editar datos cabecera*
Actores	Usuario registrado (y administrador)
Descripción	Permite al usuario modificar los nombres de la cabecera de la tabla de datos.
Dependencias	C.U.6
Precondición	Previamente debe haber una tabla cargada.
Secuencia normal	P1 - Pulsar la opción “Editar head” de la última columna de la cabecera. P2 - Introducir el nuevo valor de la columna que se desea modificar. P3 - Pulsar el botón “Enviar”.
Poscondición	La aplicación modifica los nombres de la cabecera.
Excepciones	
Comentarios	Solo se modificarán los nombres de aquellas columnas para las que se haya introducido un valor (las demás quedan inalteradas). Permite modificar varias columnas a la vez. Si el usuario hace clic en “Cerrar” o pulsa fuera de la ventana se cerrará la ventana manteniendo los datos.

Tabla 4.13.- C.U. 11: Eliminar fila

C.U. 11	*Eliminar fila*
Actores	Usuario registrado (y administrador)
Descripción	Permite eliminar una fila de la tabla.
Dependencias	C.U.6
Precondición	Previamente debe haber una tabla cargada.
Secuencia normal	P1 - Hacer click en uno de los botones “Eliminar fila” que hay en la última columna.
Poscondición	La aplicación eliminará la fila correspondiente al botón sobre el que se ha clicado.
Excepciones	
Comentarios	Para agilizar la modificación de los datos, no habrá un mensaje de confirmación para eliminar la fila.

Tabla 4.14.- C.U. 12: Eliminar columna

C.U. 12	*Eliminar columna*
Actores	Usuario registrado (y administrador)
Descripción	Permite eliminar una columna de la tabla de datos.
Dependencias	C.U.6
Precondición	Previamente debe haber una tabla cargada.
Secuencia normal	P1 - Pulsar la opción “Editar head” de la última columna de la cabecera. P2 - Hacer click en el botón “Eliminar columna” de la columna que se desea eliminar.
Poscondición	La aplicación eliminará la columna que se ha hecho click.
Excepciones	
Comentarios	Para agilizar la modificación de los datos, no habrá un mensaje de confirmación para eliminar la fila. Si el usuario hace clic en “Cerrar” o pulsa fuera de la ventana se cerrará la ventana manteniendo los datos.

Tabla 4.15.- C.U. 13: Visualizar histogramas

C.U. 13	*Visualizar histogramas*
Actores	Usuario registrado (y administrador)
Descripción	Permite al usuario la generación y visualización de histogramas desde 2 barras hasta 50.
Dependencias	C.U.6
Precondición	Previamente debe haber una tabla cargada.
Secuencia normal	P1 - Pulsar la opción “Info” de la columna que se quiere visualizar el histograma. P2 - ¿Es la columna seleccionada de tipo numérico? P3 - Introducir el número de columnas que se desea hacer el histograma (limitado entre 2 y 50). P4 - Hacer click en “Mostrar histograma”
Poscondición	La aplicación mostrará un histograma en función al tipo de variable y al número de barras introducido. Si la columna es de tipo texto el histograma refleja las veces que se repite cada valor.
Excepciones	
Comentarios	Si el usuario hace clic en “Cerrar” o pulsa fuera de la ventana se cerrará la ventana manteniendo los datos.

Tabla 4.16.- C.U. 14: Visualizar datos columna

C.U. 14	*Visualizar datos columna*
Actores	Usuario registrado (y administrador)
Descripción	Permite visualizar información relevante de la columna seleccionada. En caso de ser una columna de tipo cadena mostrará aquellos elementos más repetidos. En el caso de las columnas de tipo numérico, mostrará el máximo, mínimo y promedio de los datos.
Dependencias	C.U.6
Precondición	Previamente debe haber una tabla cargada.
Secuencia normal	P1 - Pulsar la opción “Info” de la columna que se quiere visualizar la información.
Poscondición	La aplicación mostrará los valores máximos, mínimos y promedio en caso de ser tipo entero o el valor/es más repetidos en caso de ser de tipo carácter.
Excepciones	
Comentarios	Si el usuario hace clic en “Cerrar” o pulsa fuera de la ventana se cerrará la ventana manteniendo los datos.

Tabla 4.18.- C.U. 15: Segmentar columna

C.U. 15	*Segmentar columna*
Actores	Usuario registrado (y administrador)
Descripción	Permite al usuario segmentar una columna (discretizar). Se pueden segmentar tanto columnas de tipo carácter, como de tipo numéricas.
Dependencias	C.U.6
Precondición	Previamente debe haber una tabla cargada.
Secuencia normal	P1 - Pulsar la opción “Info” de la columna que se quiere visualizar la segmentar P2 - ¿Se quiere segmentar de forma personalizada? P3 - Seleccionar el número de segmentos y sus rangos. P4 - Indicar la etiqueta de cada segmentos P5 - Pulsar el botón “segmentar manual” P6 - Si no // segmentación automática P7 - Pulsar el botón de “segmentar automático”.
Poscondición	La aplicación creará una nueva columna con los datos segmentados (discretizados) acorde al modo de segmentación seleccionado.
Excepciones	
Comentarios	Si el usuario hace clic en “Cerrar” o pulsa fuera de la ventana se cerrará la ventana manteniendo los datos.

Tabla 4.18.- C.U. 16: Descargar tabla

C.U. 16	*Descargar tabla*
Actores	Usuario registrado (y administrador)
Descripción	Descargar la tabla que hay cargada en la web en un fichero CSV.
Dependencias	C.U.6
Precondición	Previamente debe haber una tabla cargada.
Secuencia normal	P1 - Introducir el nombre de fichero. P2 - Seleccionar el tipo de separador (coma, espacio, tabulador, etc.). P3 - Seleccionar el tipo de decimal (punto o coma). P4 - Pulsar el botón “Descargar”.
Poscondición	Descarga un fichero CSV con el formato indicado o redirige a una nueva página para poder descargar de nuevo en caso de que ocurra algún error.
Excepciones	En caso de no introducir el nombre del fichero, mostrará un mensaje de error indicando que se debe rellenar. En caso de que el nombre no tenga la extensión adecuada, modificara el nombre del fichero para que contenga la extensión de forma correcta.
Comentarios	

4.2.3.- Actor: administrador

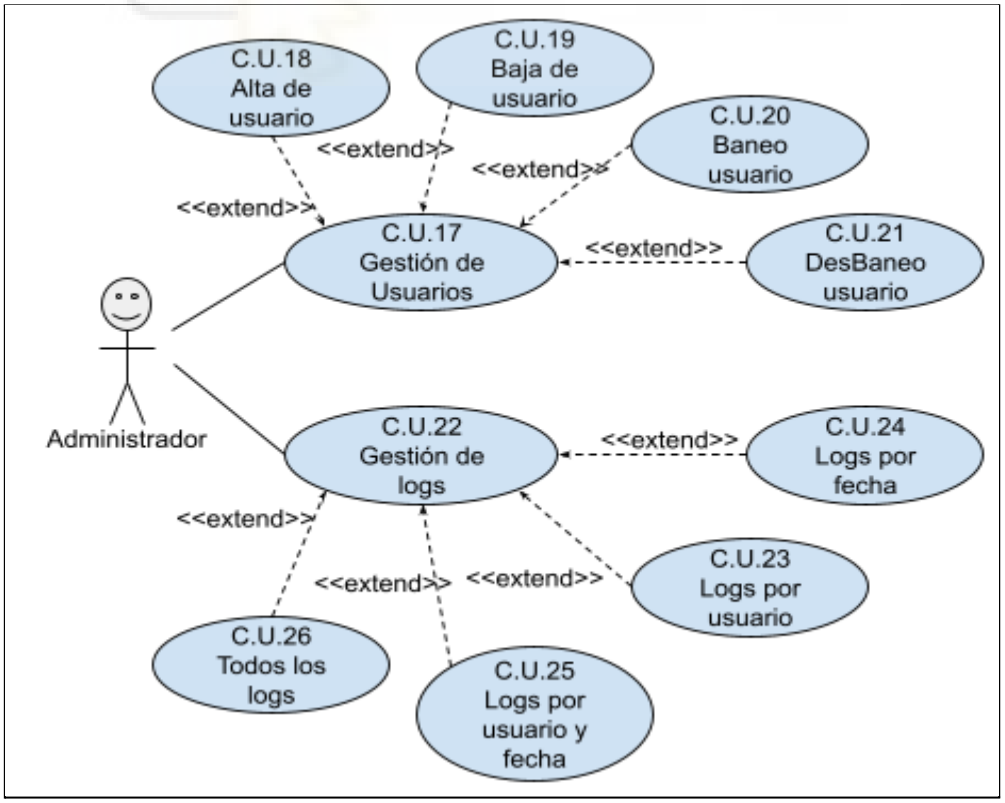


Figura 4.5.- Casos de uso del administrador

El usuario administrador, además de realizar todas las tareas que ya se han descrito en el apartado anterior, también desempeñará funciones de gestión de la aplicación. La figura 4.5 representa en diagrama de casos de uso particular de este actor. En las tablas 4.19 a 4.28 se describen detalladamente dichos casos de uso.

Tabla 4.19.- C.U. 17: Gestión de Usuario

C.U. 17	*Gestión de Usuarios*
Actores	Administrador
Descripción	Acceder a las acciones de gestión de los usuarios de la aplicación.
Dependencias	
Precondición	
Secuencia normal	P1 - Pulsar la opción “Gestión Usuario” del menú principal.
Poscondición	La aplicación muestra un listado de usuarios y las opciones de Alta, Baja, Baneo y Desbaneo de los mismos.
Excepciones	
Comentarios	Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones de gestión de usuarios.

Tabla 4.20.- C.U. 18: Alta de usuario

C.U. 18	*Alta de usuario*
Actores	Administrador
Descripción	Registrar un nuevo usuario en la aplicación.
Dependencias	C.U.17
Precondición	Haber ejecutado el C.U.17
Secuencia normal	P1 - Pulsar la opción “Alta Usuario”. P2 - Rellenar el formulario con los datos del nuevo usuario. P3 - Pulsar “Aceptar”.
Poscondición	La aplicación registra el nuevo usuario y vuelve a mostrar las opciones del C.U.17.
Excepciones	En P3, se valida que todos los datos se han introducido así como que el identificador no se ha repetido.
Comentarios	El administrador puede en cualquier momento cancelar el proceso y se volverá al C.U.17 así como los datos se quedarán grabados en la web por si desea introducirlos de nuevo. Estos datos se quedarán hasta que se recargue otra página. Si el usuario pulsa cualquier otra opción del menú principal se sale de las opciones de gestión de usuarios.

Tabla 4.21.- C.U. 19: Baja de usuario

C.U. 19	*Baja de usuario*
Actores	Administrador
Descripción	Elimina un usuario de la aplicación.
Dependencias	C.U.17, C.U.18.
Precondición	El usuario a eliminar debe existir en la base de datos.
Secuencia normal	P1 - Pulsar la opción “Baja Usuario”. P2 - Seleccionar un usuario de la lista. P3 - Confirmar la baja del usuario.
Poscondición	La aplicación elimina el usuario de la base de datos y se vuelven a mostrar las opciones del C.U.17.
Excepciones	
Comentarios	El administrador puede en cualquier momento cancelar el proceso de “Baja de Usuario” y volverá al C.U.17. Si pincha fuera del recuadro, se saldrá de la opción de baja de usuario y volverá al C.U.17.

Tabla 4.22.- C.U. 20: Baneo de usuario

C.U. 20	*Baneo de usuario*
Actores	Administrador
Descripción	Banear a un usuario para no permitirle el uso de la aplicación.
Dependencias	C.U.17 y C.U.18
Precondición	El usuario a banear debe estar registrado y no baneado.
Secuencia normal	P1 - Pulsar la opción “Ban/DesBan Usuario”. P2 - Seleccionar un usuario de la lista. P3 - Confirmar el ban del usuario.
Poscondición	La aplicación modifica los permisos del usuario de la base de datos y se vuelven a mostrar las opciones del C.U.17.
Excepciones	P2/P3 - El administrador antes de confirmar la acción puede salir de la ventana con la opción de “Cerrar”. P1/P2/P3 - Si se hace clic en cualquier otra parte de la ventana esta se cerrará automáticamente.
Comentarios	El administrador puede en cualquier momento cancelar el proceso de “Baneo Usuario” y volverá al C.U.17. Si pincha fuera del recuadro, se saldrá de la opción de “Baneo Usuario”.

Tabla 4.23.- C.U. 21: Desbaneo de usuario

C.U.21	*Desbaneo de usuario*
Actores	Administrador
Descripción	Permite al administrador desbanear a un usuario para permitir el uso de la aplicación.
Dependencias	C.U.17, C.U. 18 y C.U. 20
Precondición	El usuario a desbanear debe estar registrado y previamente baneado.
Secuencia normal	P1 - Pulsar la opción “Ban/Desbaneo Usuario”. P2 - Seleccionar un usuario de la lista. P3 - Confirmar el desbaneo del usuario.
Poscondición	La aplicación modifica los permisos del usuario de la base de datos y se vuelven a mostrar las opciones del C.U.17.
Excepciones	P2/P3 - El administrador antes de confirmar la acción puede salir de la ventana con la opción de “Cerrar”. P1/P2/P3 - Si se hace clic en cualquier otra parte de la ventana esta se cerrará automáticamente.
Comentarios	El administrador puede en cualquier momento cancelar el proceso de “Desbaneo Usuario” y volverá al C.U.17. Si pincha fuera del recuadro, se saldrá de la opción de “Desbaneo Usuario”.

Tabla 4.24.- C.U. 22: Gestión de logs

C.U. 22	*Gestión de logs*
Actores	Administrador
Descripción	Opción del menú principal del acceso privado del administrador que da acceso a las acciones de consulta de las acciones de los usuarios.
Dependencias	
Precondición	
Secuencia normal	P1 - Pulsar la opción “Gestión logs” del menú principal
Poscondición	La aplicación muestra un listado con todos los logs además de diferentes apartados para descargar ficheros en base a unos filtros
Excepciones	
Comentarios	El usuario en el listado puede visualizar todos los logs además de buscar en base a una barra de búsqueda

Tabla 4.25.- C.U. 23: Logs por usuario

C.U. 23	*Logs por usuario*
Actores	Administrador
Descripción	Descargar un fichero de datos en formato “CSV” con los logs de uso de la herramienta de un determinado usuario.
Dependencias	C.U.22
Precondición	Haber ejecutado el C.U.22
Secuencia normal	P1 - Pulsar la opción “Usuario” del menú principal. P2 - Seleccionar al usuario cuyos logs se desea descargar. P3 - Pulsar el botón “Descargar” para iniciar la descarga.
Poscondición	La aplicación descarga un fichero con aquellos logs del usuario seleccionado.
Excepciones	
Comentarios	Si el usuario pulsa cualquier cualquier parte fuera del cuadrado de selección de usuario o pulsa en cerrar, se cerrará la ventana.

Tabla 4.26.- C.U. 24: Logs por fecha

C.U. 24	*Logs por fecha*
Actores	Administrador
Descripción	Descargar un fichero de datos en formato “CSV” con los logs de uso de la herramienta comprendidos entre dos fechas.
Dependencias	C.U.22
Precondición	Haber ejecutado el C.U.22.
Secuencia normal	P1 - Pulsar la opción “Fecha” del menú principal. P2 - Indicar las fechas por las que se desea filtrar para descargar los logs. P3 - Pulsar el botón “Descargar” para iniciar la descarga.
Poscondición	La aplicación descarga un fichero con los logs correspondiente a las fechas introducidas.
Excepciones	Se validará que los campos no son nulos. Si alguno no se ha introducido se mostrará una alerta indicando que el campo debe ser completado.
Comentarios	Si el usuario pulsa cualquier cualquier parte fuera del cuadrado de selección de usuario o pulsa en cerrar, se cerrará la ventana.

Tabla 4.27.- C.U. 25: Logs por usuario y fecha

C.U. 25	*Logs por usuario y fecha*
Actores	Administrador
Descripción	Descargar un fichero de datos en formato “CSV” con los logs de uso de la herramienta de un usuario determinado comprendidos entre dos fechas.
Dependencias	C.U.22
Precondición	Haber ejecutado el C.U.22.
Secuencia normal	P1 - Pulsar la opción “Usuario y fecha” del menú principal. P2 - Seleccionar al usuario y las fechas a filtrar que se desean para descargar los logs. P3 - Pulsar el botón “Descargar” para iniciar la descarga.
Poscondición	La aplicación descarga un fichero con aquellos logs correspondiente al usuario y las fechas introducidas.
Excepciones	Se validará que los campos no son nulos. Si alguno no se ha introducido se mostrará una alerta indicando que el campo debe ser completado.
Comentarios	Si el usuario pulsa cualquier cualquier parte fuera del cuadrado de selección de usuario o pulsa en cerrar, se cerrará la ventana.

Tabla 4.28.- C.U. 26: Todos los logs

C.U. 26	*Todos los logs*
Actores	Administrador
Descripción	Descargar un fichero de datos en formato “CSV” con todos los logs de uso de la herramienta.
Dependencias	C.U.22
Precondición	Haber ejecutado el C.U.22
Secuencia normal	P1 - Pulsar la opción “Todos” del menú principal. P2 - Pulsar el botón “Descargar” para iniciar la descarga.
Poscondición	La aplicación descarga un fichero con todos los logs de la base de datos.
Excepciones	
Comentarios	Si el usuario pulsa cualquier cualquier parte fuera del cuadrado de selección de usuario o pulsa en cerrar, se cerrará la ventana.

4.3.- DISEÑO E IMPLEMENTACIÓN

4.3.1.- Diagrama Entidad/Relación

Para esta aplicación, no se requiere una base de datos muy compleja, con tan solo dos tablas se consigue toda la funcionalidad que propuesta en los objetivos del proyecto. De estas dos tablas, una es para almacenar a los usuarios de la aplicación, y la otra para tener un registro de logs, es decir, de las acciones realizadas por los propios usuarios. Como se puede ver en la Figura 4.6 ambas tablas están relacionadas mediante el el campo "correo", que se utiliza como identificados de usuario.

En la tabla de usuarios, está toda la información relativa a los usuarios de la aplicación, desde el identificador de acceso (PK "correo" en figura 4.6), la contraseña, el tipo de usuario (registrado o administrador), y el estado de dicho usuario (baneado: sí o no).

En la tabla de logs se guardará registro de todas las acciones de los usuarios junto con el identificador de dicho usuario (FK1 "user" en figura 4.5). También se guarda la fecha en la que la acción fue realizada.

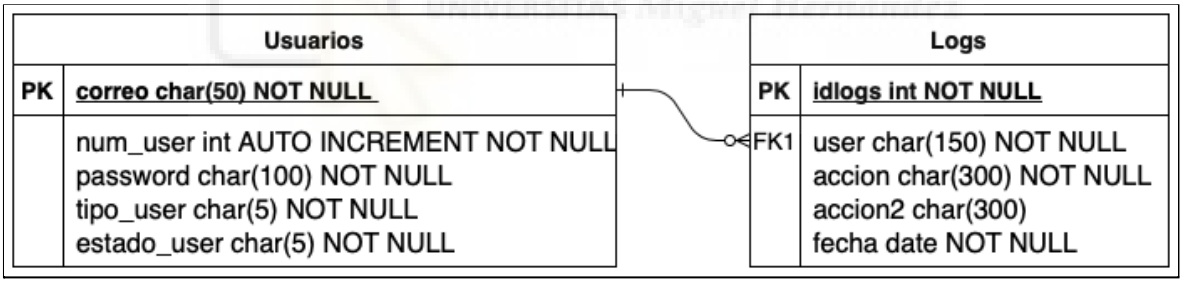


Figura 4.6.- Diagrama entidad/relación

4.3.2.- Interfaz gráfica

En este apartado se van a mostrar algunos de los diseños de pantalla realizados así como su posterior implementación real en la aplicación. El primer esquema que se puede observar (figuras 4.7 y 4.8) es el de la página principal de la aplicación, con la cabecera y pie de la interfaz, así como de la zona principal de trabajo. Como se puede ver hay poca diferencia entre el esquema y la implementación real. Se ha tratado de ser lo más fiel posible a los esquemas realizados con el objetivo de simular que detrás de los mockups hay un grupo de especialistas en UI/UX y de esta manera se cumplen uno de los objetivos propuestos de una interfaz sencilla e intuitiva.

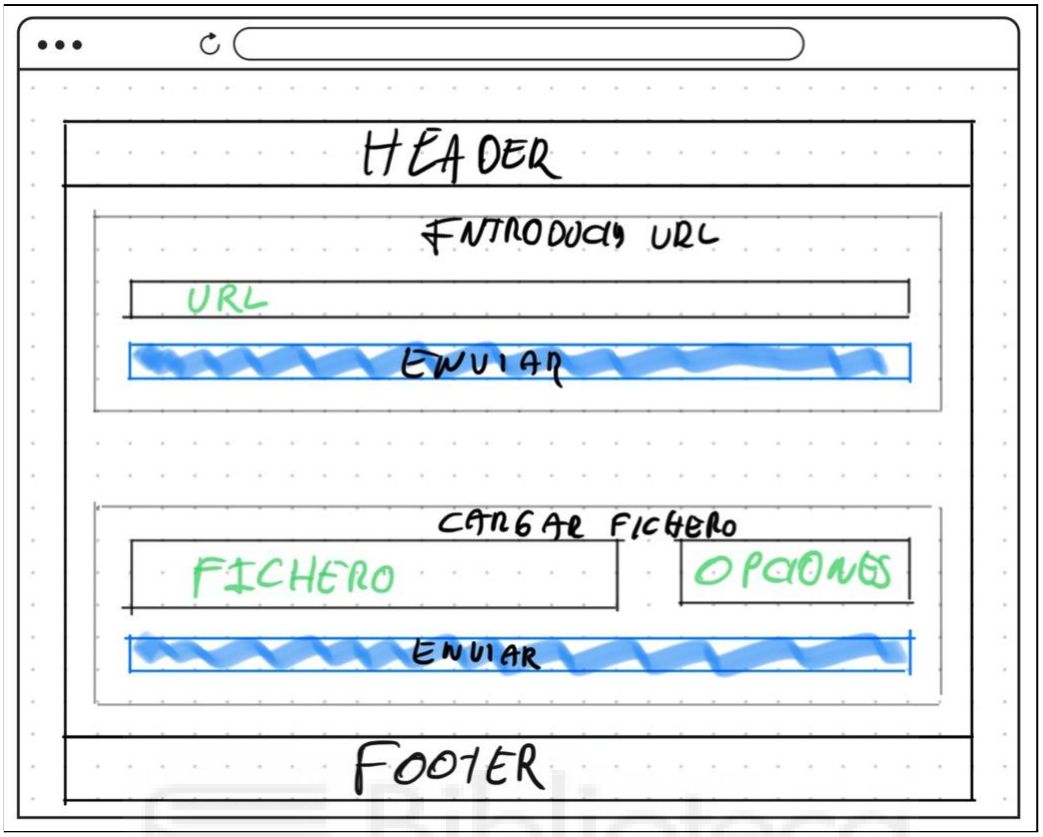


Figura 4.7.- Mockup de la página principal

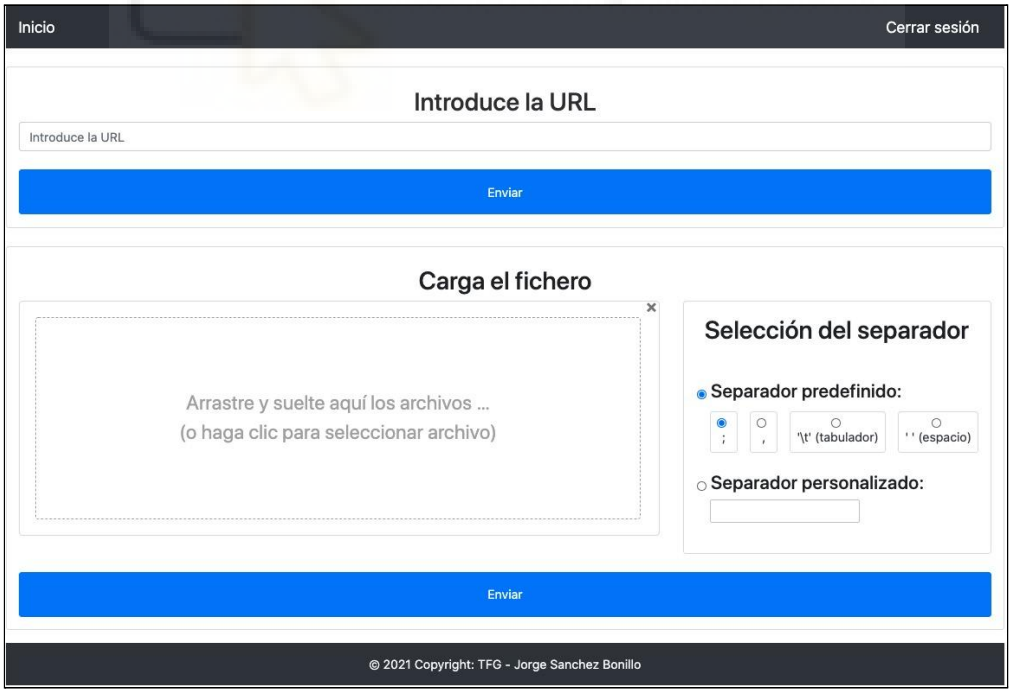


Figura 4.8.- Implementación de la página principal

Otro de los esquemas realizados es el perteneciente a la gestión de usuarios, como en el caso anterior, se ha tratado de ser lo más fiel posible al esquema de diseño.

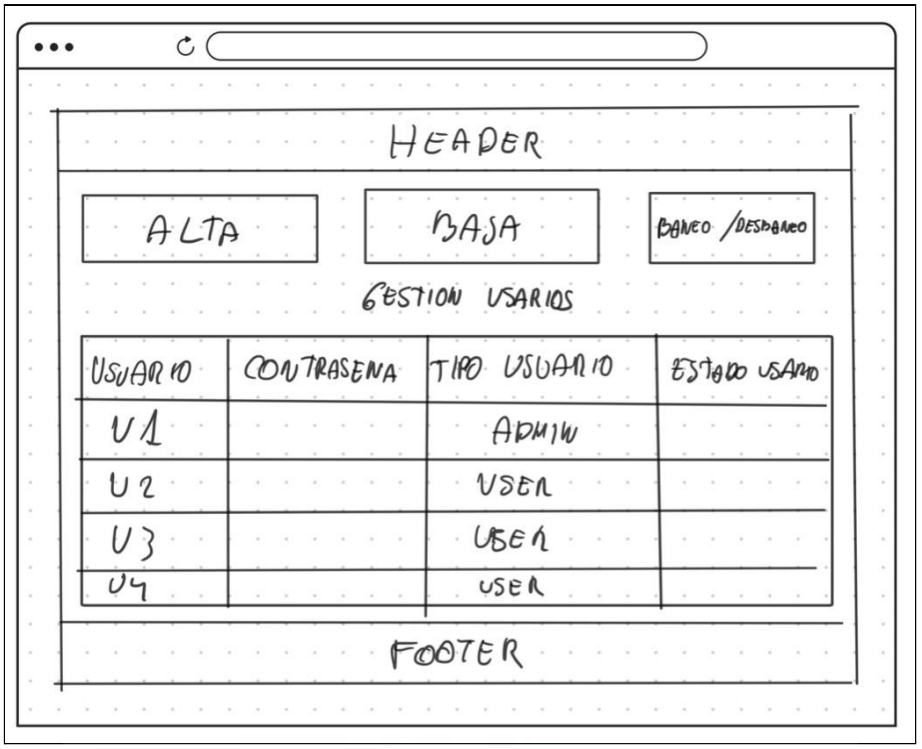


Figura 4.9.- Mockup de la página de gestión de usuarios

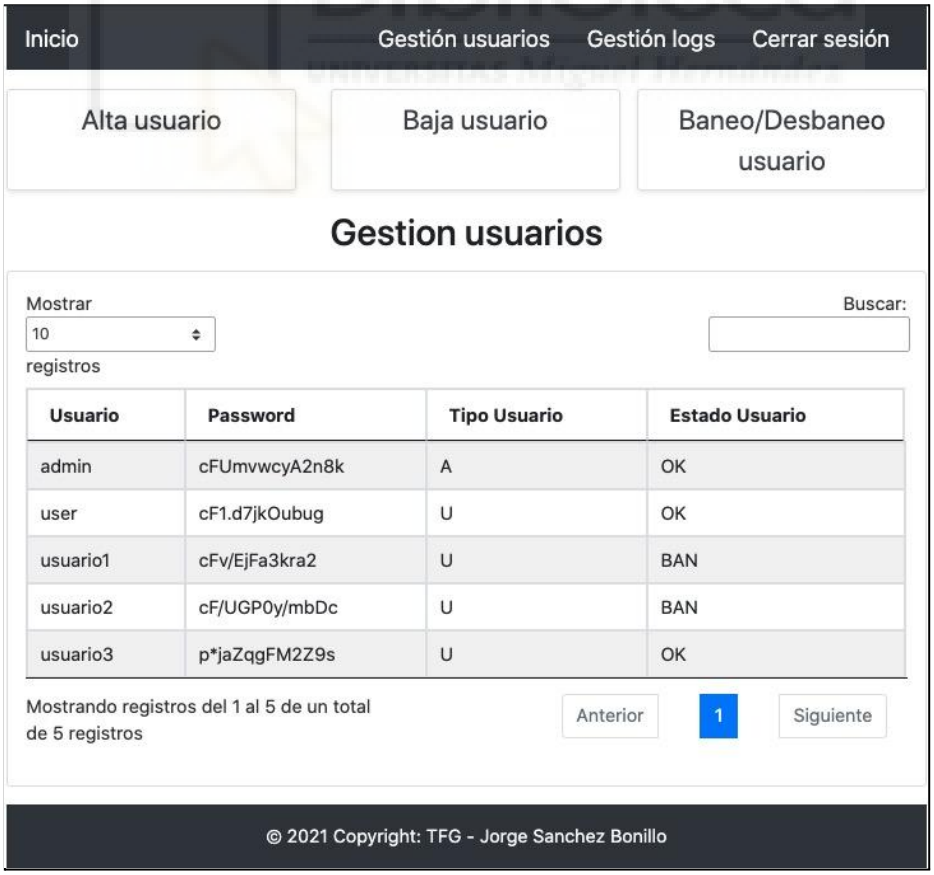


Figura 4.10.- Implementación de gestión de usuarios

4.4.- IMPLANTACIÓN

Finalmente, para implantar la aplicación, es necesario disponer de un servidor web, en este caso FLASK, que puede ser programado en Python. También se necesita el sistema gestor de bases de datos MySQL (o MariaDB) con la base de datos del sistema (apartado 4.3.1), y el usuario administrador dado de alta de antemano en la tabla de usuarios.

En el anexo 2 (Manual de instalación) se explica con más detalle como realizar todo el proceso de instalación.

Capítulo 5
Conclusiones y trabajo futuro

5.1.- CONCLUSIONES

En cuanto a las conclusiones del proyecto, se puede afirmar que finalmente se ha conseguido el principal objetivo de raspar los datos contenidos en tablas de páginas web dada una dirección URL, así como la posibilidad de seleccionar la tabla que se desea descargar o se desea modificar. También se ha cumplido el objetivo de poder cargar un fichero exclusivamente del formato CSV y manipularlo.

También se ha cumplido el objetivo de realizar un control de errores estricto, minimizando lo máximo posible los errores por parte del usuario, obligando a introducir valores antes de enviar los datos o comprobando en la parte del servidor que los datos han sido introducidos correctamente. En caso de haber un error se han implementado mensajes de error adaptados a cada situación para ayudar al usuario a resolverlo.

En la parte de diseño, se ha procurado ser lo más minimalista y simple posible para no recargar la interfaz y permitir que una persona sin conocimientos técnicos pueda hacer uso de la aplicación cumpliendo uno de los objetivos secundarios del proyecto.

El objetivo de mantener un control de las acciones de los usuarios mediante un sistema gestor de logs se ha cumplido satisfactoriamente de la misma manera que el sistema de gestión de usuarios de forma que se protege al máximo el acceso a la aplicación a extraños no registrados.

En cuanto a los objetivos que me marqué a la hora de desarrollar la aplicación de aprender un nuevo lenguaje y nuevas tecnologías que no había estudiado previamente, puedo afirmar que el conocimiento que tengo del lenguaje principal Python en relación a cuando empecé el proyecto es muy superior con un mayor control de la tecnología. También el uso de un framework no visto antes por mi parte, “Flask”, me ha hecho aprender mucho de la tecnología así como todas las posibilidades de desarrollo que ofrece. Así como poder asentar todo el conocimiento en cuanto a lenguajes obtenido durante el tiempo de estudio de la carrera y la posterior aplicación en este proyecto.

Por último, la creación de una memoria técnica, me ha permitido aplicar los conocimientos teóricos estudiados a lo largo de la carrera así como la realización de los casos de uso o la realización del manual de instalación o el manual de uso.

5.2.- POSIBLES DESARROLLOS FUTUROS

En este apartado, se van a incluir algunos de los posibles desarrollos futuros que se pueden seguir implementando en la aplicación.

- Implantación del protocolo HTTPs: la aplicación actualmente funciona solamente bajo el protocolo HTTP, pero en la actualidad, todas las aplicaciones que trabajen con datos importantes, en el caso de la aplicación, los datos del usuario y la contraseña, deben ir cifrados bajo el protocolo HTTPs para evitar que un usuario externo se haga con los datos.
- Agregación de tablas: guardar las tablas de manera interna bajo un identificador y permitir unir varias tablas en una sola.
- Planes de pago (premium): realizar un sistema de planes de pago bajo demanda en los cuales, dependiendo del plan elegido, se van a poder raspar una cantidad limitada de páginas al mes o un número limitado de filas.
- Permitir a los usuarios registrarse: basándonos en la anterior mejora, permitir al usuario registrarse en la aplicación de manera gratuita, hacer uso durante un tiempo de prueba y limitando la cantidad de datos que puede raspar.

- Raspar datos que se cargan dinámicamente: actualmente la aplicación solo permite raspar datos de páginas que tienen en el código HTML un elemento <TABLE>, son la mayoría de las páginas webs y la mayoría de las tablas en Internet, pero hay ciertas webs que cargan sus datos de manera dinámica al hacer scroll mediante un script, permitir cargar los datos de estas páginas.
- Raspar datos en DIVs: al hilo del punto anterior, decir que también existen webs en las que los datos se presentan, no en un objeto <TABLE>, sino en una serie de elementos <DIV> anidados y maquetados con CSS. La versión actual de la aplicación tampoco puede raspar estos datos.

Bibliografía

[1] The WorldWideWeb browser
Tim Berners-Lee
https://www.w3.org/People/Berners-Lee/WorldWideWeb.html
Consultado: diciembre 2020

[2] Conoce la historia de Internet desde su primera conexión hasta hoy
Elena Bello
https://www.iebschool.com/blog/historia-de-internet-innovacion/
Consultado: diciembre 2020

[3] Historia de Internet: cómo nació y cuál fue su evolución
Luis Bahillo
https://marketing4ecommerce.net/historia-de-internet
Consultado: diciembre 2020

[4] Total number of Websites
https://www.internetlivestats.com/total-number-of-websites/
Consultado: diciembre 2020

[5] Open Data Handbook
Open Knowledge Foundation
https://opendatahandbook.org/guide/es
Consultado: diciembre 2020

[6] Open Data: datos, transparencia y conocimiento abierto.
Ignasi Alcalde
https://ignasialcalde.es/open-data-datos-transparencia-y-conocimiento-abierto/
Consultado: diciembre 2020

[7] Datos abiertos más allá del sector público
Gobierno de España (datos.gov.es)
https://datos.gob.es/sites/default/files/doc/file/toc_informe_1_-_datos_abiertos_mas_alla_de_los_gobiernos_final_0.pdf
Consultado: enero 2021

[8] Los 8 principios básicos de los Datos Abiertos.
Azahara Benito
https://www.ogoov.com/es/blog/los-8-principios-basicos-de-los-datos-abiertos/
Consultado: diciembre 2020

[9] Comunidades en España que fomentan el Open Data
https://oshl.umh.es/2015/07/09/comunidades-en-espana-que-fomentan-el-open-data
Consultado: enero 2021

[10] Trabajo con archivos CSV y JSON en soluciones de datos.
https://docs.microsoft.com/es-es/azure/architecture/data-guide/scenarios/csv-and-json
Consultado: enero 2020

[11] XML JSON YAML: formatos para intercambiar información.
Andrea Rodriguez
https://hipertextual.com/archivo/2014/05/xml-json-yaml/
Consultado: enero 2020

[12] Qué es el Web scraping? Introducción y herramientas.
Marq Martí
https://sitelabs.es/web-scraping-introduccion-y-herramientas/
Consultado: enero 2021

[13] Qué es el web scraping
Ainhoa Lafuente
https://aukera.es/blog/web-scraping/
Consultado: enero 2020

[14] Con import.io transforme información de Internet en datos utilizables para su historia
Jorge Luis Alonso G.
https://jl-alonso.medium.com/con-import-io-transforme-informacion-de-internet-en-datos-utilizables-para-su-historia-29982da54199
Consultado: enero 2021

[15] Import.io
https://www.capterra.es/software/135185/import-io
Consultado: enero 2021

[16] La magia de Import.io
https://es.schoolofdata.org/2014/12/04/la-magia-de-import-io/
Consultado: enero 2021

[17] Table Capture
George Mike
https://chrome.google.com/webstore/detail/table-capture/iebpjdmgckacbodjpijphcplhebcmeop?hl=es
Consultado: diciembre 2020

[18] Sobre Nosotros
Octoparse
https://www.octoparse.es/about
Consultado: diciembre 2020

[19] Las 20 mejores herramientas de web scraping para extracción de datos
Melisa
https://melisa-40349.medium.com/las-20-mejores-herramientas-de-web-scraping-para-extracción-de-datos-cb9b63b42e4a
Consultado: diciembre 2020

[20] 80legs Datafiniti
https://80legs.com/products/datafiniti/
Consultado: enero 2021

[21] 80legs Pricing
https://80legs.com/pricing/
Consultado: enero 2021

[22] Solicitudes HTTP en Python con Request
Moises
https://unipython.com/solicitudes-http-en-python-con-requests/
Consultado: diciembre 2020

[23] Requests: HTTP para Humanos
https://requests.readthedocs.io/es/latest/
Consultado: diciembre 2020

[24] Python's Request Library (Guide)
Alex Ronquillo
https://realpython.com/python-requests/#other-http-methods
Consultado: enero 2021

[25] Beautiful Soup
Leonard Richardson
https://www.crummy.com/software/BeautifulSoup/
Consultado: diciembre 2020

[26] Web scraping con Python. Extraer datos de una web. Guía de inicio de Beautiful Soup
https://j2logo.com/python/web-scraping-con-python-guia-inicio-beautifulsoup/
Consultado: diciembre 2020

[27] Cómo hacer Web Scraping con Selenium
Rafael Zambrano
https://openwebinars.net/blog/como-hacer-web-scraping-con-selenium/
Consultado: diciembre 2020

[28] Instalando Selenium en Python
https://unipython.com/instalando-selenium-python/
Consultado: diciembre 2020

[29] Python y Selenium: Cómo construir un bot simple de inicio de sesión automático.
PythonDiario
http://pythondiario.com/2017/05/python-y-selenium-como-construir-un-bot.html
Consultado: diciembre 2020

[30] Web Scraping con Scrapy Framework y Jupyter
José Fernando González Montero
https://josefgonzalez.me/es/post/scrapy-en-jupyter/
Consultado: diciembre 2020

[31] Scrapy
https://scrapy.org
Consultado: enero 2021

[32] Scrapy Cloud
https://www.scrapinghub.com/scrapy-cloud/
Consultado: enero 2021

[33] Lenguajes del lado servidor o cliente: diferencias
https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/lenguajes-del-lado-servidor-o-del-cliente-diferencias/
Consultado: enero 2021

[34] HTML
https://desarrolloweb.com/home/html#track4
Consultado: enero 2021

[35] HTML Tags
https://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html
Consultado: enero 2021

[36] Lección 1: ¿Qué es CSS?
http://es.html.net/tutorials/css/lesson1.php
Consultado: enero 2021

[37] CSS
https://developer.mozilla.org/es/docs/Web/CSS
Consultado: enero 2021

[38] Lección 2: ¿Cómo funciona CSS?
http://es.html.net/tutorials/css/lesson2.php
Consultado: enero 2021

[39] JavaScript
https://developer.mozilla.org/es/docs/Web/JavaScript
Consultado: enero 2021

[40] ¿Qué es JavaScript?
https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Qué_es_JavaScript
Consultado: enero 2021

[41] About Bootstrap
https://getbootstrap.com/docs/5.0/about/overview/
Consultado: enero 2021

[42] ¿Qué es AJAX y cómo funciona?
Gustavo B.
https://www.hostinger.es/tutoriales/que-es-ajax/
Consultado: enero 2021

[43] Introducción a JSON
https://www.json.org/json-es.html
Consultado: enero 2021

[44] Trabajando con JSON
https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON
Consultado: enero 2021

[45] JQuery
https://jquery.com
Consultado: enero 2021

[46] Los mejores 26 plugins JQuery que tienes que conocer.
Andrea Cumbo
https://cumboandrea.me/26-jquery-plugins-muy-utiles/
Consultado: enero 2021

[47] DataTables jQuery plugin
Jose Aguilar
https://www.jose-aguilar.com/blog/datatables-jquery-plugin/
Consultado: enero 2021

[48] DataTables
https://datatables.net
Consultado: enero 2021

[49] Bootstrap File Input
https://plugins.krajee.com/file-input
Consultado: enero 2021

[50] Github fileInput
https://github.com/kartik-v/bootstrap-fileinput
Consultado: enero 2021

[51] Preguntas frecuentes generales sobre Python
Autor / URL https://docs.python.org/es/3/faq/general.html
Consultado: enero 2021

[52] The Making of Python
Bill Venners
https://www.artima.com/intv/python.html
Consultado: enero 2021

[53] Programación en Python - Nivel básico
"@Covantec"
https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion1/caracteristicas.html
Consultado: enero 2021

[54] About pandas
https://pandas.pydata.org/about/index.html
Consultado: enero 2021

[55] Package overview
https://pandas.pydata.org/pandas-docs/stable/getting_started/overview.html
Consultado: enero 2021

[56] Pandas en Python, con ejemplos -Parte I- Introducción
https://jarroba.com/pandas-python-ejemplos-parte-i-introduccion/
Consultado: enero 2021

[57] Chapter 1 Introduction to MySQL Connector/Python
https://dev.mysql.com/doc/connector-python/en/connector-python-introduction.html
Consultado: enero 2021

[58] Chapter 4 Connector/Python Installation
https://dev.mysql.com/doc/connector-python/en/connector-python-installation.html
Consultado: enero 2021

[59] 7.1 Connector/Python Connection Arguments
https://dev.mysql.com/doc/connector-python/en/connector-python-connectargs.html
Consultado: enero 2021

[60] MatPlotLib
https://matplotlib.org
Consultado: enero 2021

[61] NumPy
https://numpy.org
Consultado: enero 2021

[62] Introducción a la Librería Matplotlib de Python - Parte 1
https://aprendeia.com/libreria-pandas-de-matplotlib-tutorial/
Consultado: enero 2021

[63] Operaciones con fechas y horas. Calendarios
"Pherkad"
https://python-para-impacientes.blogspot.com/2014/02/operaciones-con-fechas-y-horas.html
Consultado: enero 2021

[64] Encoding and Decoding Base64 Strings in Python
"@RajuKumar19"
https://www.geeksforgeeks.org/encoding-and-decoding-base64-strings-in-python/
Consultado: enero 2021

[65] Base64
https://codebeautify.net/base64
Consultado: enero 2021

[66] ¿Qué es MySQL? Explicación detallada para principiantes.
Gustavo B.
https://www.hostinger.es/tutoriales/que-es-mysql/
Consultado: enero 2021

[67] ¿Por qué aprender Flask?
Eduardo Ismael García Pérez
https://codigofacilito.com/articulos/por-que-flask
Consultado: enero 2021

[68] MVC(Model, View, Controller) Explicado.
Uriel Hernandez
https://codigofacilito.com/articulos/mvc-model-view-controller-explicado
Consultado: enero 2021

[69] Qué es MVC
Miguel Angel Alvarez
https://desarrolloweb.com/articulos/que-es-mvc.html
Consultado: enero 2021

[70] Qué es Flask
José Domingo Muñoz
https://openwebinars.net/blog/que-es-flask/
Consultado: enero 2021

[71] Instalar y usar virtualenv con Python 3
https://help.dreamhost.com/hc/es/articles/115000695551-Instalar-y-usar-virtualenv-con-Python-3
Consultado: enero 2021

[72] flask.templating render_template Example Code
Matt Makai
https://www.fullstackpython.com/flask-templating-render-template-examples.html
Consultado: enero 2021

[73] Request Flask
http://www.manualweb.net/flask/request-flask/
Consultado: enero 2021

[74] Cómo servir ficheros y contenido estático con Flask
Juan José Lozano Gómez
https://j2logo.com/python/flask/servir-ficheros-y-contenido-estatico-con-flask/
Consultado: enero 2021

[75] Sesión en Flask
Victor Cuervo
http://lineadecodigo.com/python/sesion-en-flask/
Consultado: enero 2021

[76] PyCharm: uno de los mejores IDE para Python
https://www.escuelapython.com/pycharm-uno-de-los-mejores-ide-para-python/
Consultado: enero 2021

[77] JetBrains Company
https://www.jetbrains.com/company/
Consultado: enero 2021

[78] Funcionalidades de PyCharm
https://www.jetbrains.com/es-es/pycharm/features/
Consultado: enero 2021

[79] Funcionalidades de PyCharm
Encarna Abellan
https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html
Consultado: enero 2021

[80] Introducción a Scrum… en menos de 5 minutos
Exceltic
https://www.youtube.com/watch?v=P25JP0u6UKw
Consultado: enero 2021

[81] Enlace al fichero "TFG.zip" con la aplicación descrita en esta memoria
Jorge Sánchez Bonillo
https://drive.google.com/drive/folders/19CPN01J9DXJnj9u3M74kxCaoA8mEaBFM?usp=sharing

Anexo 1
La metodología SCRUM

SCRUM [79] es una metodología de trabajo ágil cuyo objetivo es la entrega de versiones en cortos periodos de tiempo de forma que permita al cliente trabajar de manera conjunta con el equipo desarrollador de la aplicación y haya una retroalimentación entre el cliente y el responsable del proyecto.

A1.1.- FUNDAMENTOS DE SCRUM

Los pilares fundamentales de la metodología son LOS SIGUIENTES:

- Transparencia: todos los actores del proyecto son conscientes del estado del proyecto y los problemas surgidos, de modo que hay una visión global común del proyecto.

- Inspección: el equipo comprueba frecuentemente el progreso del proyecto para detectar posibles problemas. Se trata de saber que el equipo funciona de manera organizada así como que el trabajo se realiza.

- Adaptación: los requisitos del proyecto son cambiantes para ello el equipo se organiza para conseguir el objetivo.

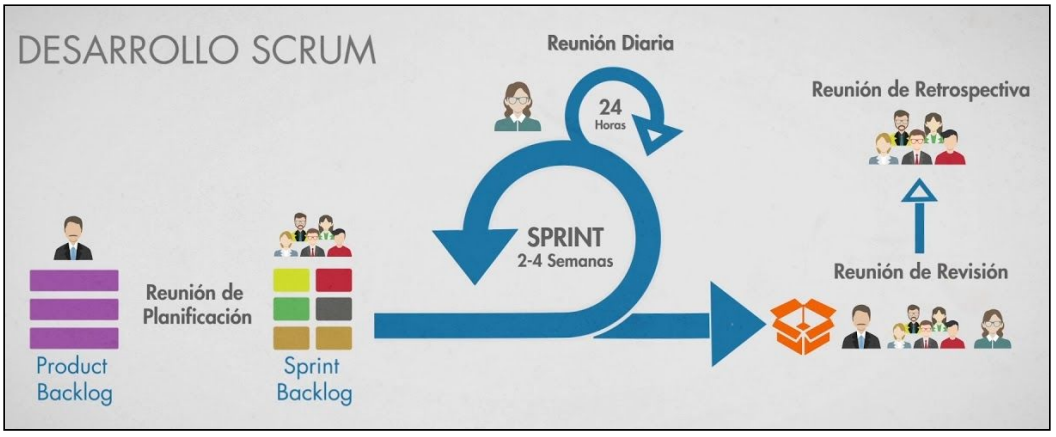


Figura A1.1 Esquema de la metodología SCRUM [80]

A1.2.- ACTORES EN SCRUM

En una metodología Scrum existe un equipo de trabajo en el que tienen que aparecer, al menos los siguientes tres roles principales:

- Product Owner: se trata del responsable del proyecto, es el encargado de hablar de manera constante con el cliente y debe de tener conocimientos sobre el negocio. Solo puede haber un rol de este tipo en el proyecto.

- Scrum Master: es el encargado de que las técnicas sean aplicadas de manera eficiente en el proyecto, también se encarga de resolver los posibles problemas ocurriendo en la fase *sprint.*

- Equipo de desarrollo: se trata de un equipo multifuncional y auto-organizado y se encargan de desarrollar el proyecto y las tareas propuestas por el *Product Owner*

A1.3.- HITOS DE SCRUM

De la misma manera, existen cinco hitos o momentos importantes durante el desarrollo que deben estar agendados, es importante que se cumplan rigurosamente para que el proyecto

no se estanque o, en otras palabras, para que la marcha del proyecto sea lo más ágil posible:

- Reunión de planificación: en esta reunión se organizan las tareas y se planifica el objetivo de *sprint.* La duración de esta reunión suele ser menor a 8 horas para los sprints de 1 mes. Esta reunión se realiza para que el equipo tenga un claro objetivo y se encuentre altamente comprometido con el proyecto. En la reunión se hacen las siguientes preguntas:
    - ¿Qué se va a hacer en el sprint?
    - ¿Cómo lo vamos a hacer?

- *Sprint*: es la parte más importante de la metodología, se trata de una fase en la cual se desarrollan todas las tareas planificadas en la fase anterior. La duración máxima de esta fase es de 1 mes aunque puede ser menor dependiendo de la comunicación que se tenga con el cliente.

- Reunión diaria: se trata de una reunión que se realiza todos los días y debe de tener una duración de menos de 15 minutos, en ella intervienen tanto el scrum Master como el equipo de desarrollo y cuyo objetivo es comprobar el estado de las tareas. Se deben de realizar las siguiente preguntas en la reunión:
    - ¿Qué hice ayer?
    - ¿Qué voy a hacer hoy?
    - ¿Tengo algún problema que me deben solucionar previamente?

- Reunión de revisión: se trata de la reunión que se realiza al final de cada sprint. Su duración es de hasta 4 horas y puede asistir el cliente, al cual se le muestra el funcionamiento así como valida los cambios y propone nuevas tareas para implementar en la aplicación.

- Reunión de retrospectiva: se trata del último evento de la metodología y se evalúa la manera en que se ha implementado la metodología Scrum así como se realizan una lista de mejoras para seguir mejorando en las siguientes iteraciones.

A1.4.- HERRAMIENTAS PARA SCRUM

Existen dos herramientas optimizadas que permiten maximizar la transparencia dentro del equipo:

- Product backlog: se trata del listado completo de tareas del proyecto, junto a la tarea deberá haber una estimación de tiempo provista por el equipo de desarrollo.

De esta lista se encarga el Product Owner y establece junto al cliente unas prioridades para el desarrollo de las tareas. En la reunión de planificación, el equipo de desarrollo elige las tareas de este listado para generar el siguiente listado.

- Sprint backlog: es la lista de tareas para desarrollar durante el periodo de sprint, es el propio equipo de desarrollo quien elige las tareas y compone esta nueva lista. Esta lista no puede cambiar durante la fase sprint.

A1.5.- VENTAJAS E INCONVENIENTES

Para concluir, a continuación se describen una serie de ventajas y desventajas de usar esta metodología ágil. Las ventajas más importantes serían:

- Facilidad de aprendizaje: los roles, hitos y herramientas son claros y están muy relacionados con la manera de trabajar diariamente.
- El cliente puede hacer uso del producto rápidamente
- El cliente puede ver el desarrollo de manera casi en directo
- Poca probabilidad de imprevisto debido a la alta comunicación que hay con el cliente

Entre las desventajas se encuentran:

- A pesar de ser fácil de aprender, es difícil de implementar, ya que debe haber un cambio de cultura en todos los roles del proyecto, desde el cliente hasta los directivos.
- Se requiere de equipos multidisciplinares y son difíciles de encontrar.
- El equipo puede verse tendido a conseguir el objetivo de una manera rápida y causar el no desarrollar un código de calidad.

Anexo 2
Manual de instalación

En el siguiente manual de instalación se va a explicar paso a paso como instalar la aplicación de web scraping desarrollada en este proyecto, aunque partiendo de ciertos prerrequisitos que debe cumplir el equipo donde se va a instalar la aplicación.

A2.1.- PRERREQUISITOS

Para la instalación de la aplicación es necesario que previamente se disponga de una base de datos MySQL o MariaBD instalada y en funcionamiento, una forma sencilla y rápida de disponer de este software es instalando el paquete XAMPP, disponible en apachefriends.org, que contiene, entre otros servicios, la base de datos MariaDB. También debe estar instalado Python con anterioridad, la presente guía se ha realizado con la versión 3.8. Este producto se puede obtener en la web python.org.

Por último, hay que descargar el fichero .zip "TFG.zip"[81] y descomprimirlo en la carpeta del sistema donde se desee instalar la aplicación, en este manual se va a considerar que la

aplicación se instalará dentro de la carpeta "**C:\web_python**", al descomprimir el fichero .zip en dicha carpeta se creará otra carpeta con el nombre "**TFG**", donde se realizarán los siguientes pasos.

A2.2.- INSTALACIÓN DE PAQUETES

Lo primero es abrir una consola o terminal de Windows en la carpeta del proyecto. A continuación se describen los pasos a seguir.

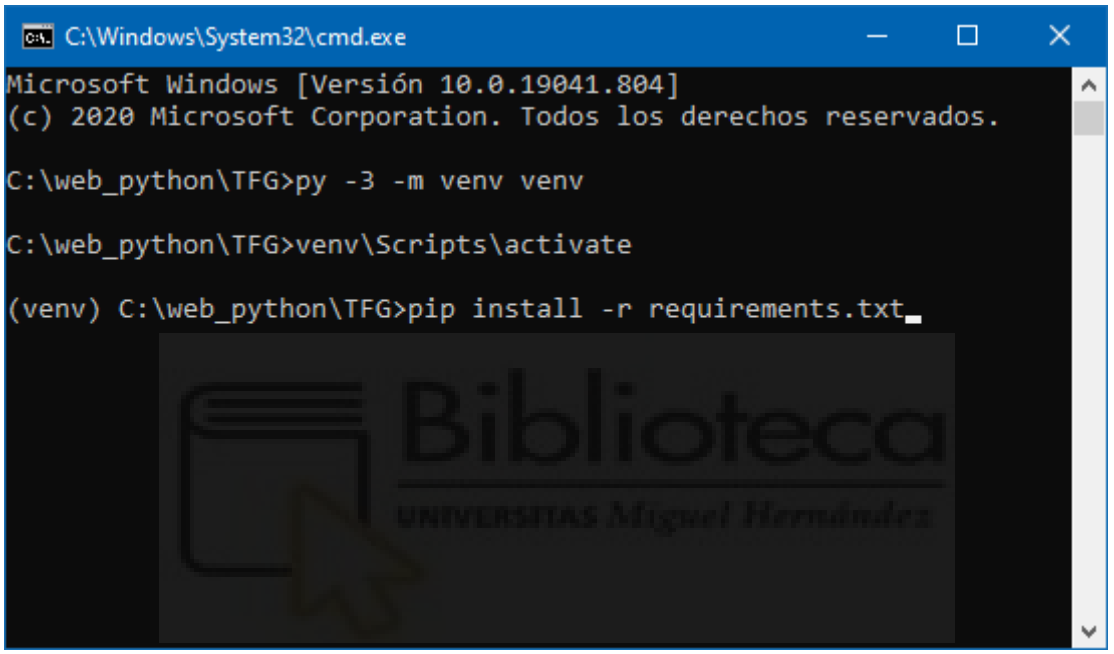


Figura A2.1.- Consola de Windows en la carpeta de instalación

Paso 1.- Crear el entorno virtual con el comando:

```
py -3 -m venv venv
```

Paso 2.- Activar el entorno con el comando:

```
venv\Scripts\activate
```

Tras el paso 2 se modifica el prompt de la consola, ahora es:

"**(venv) C:\web_python\scrap>**"

Paso 3.- Instalar las dependencias del fichero "requirements.txt" con el comando:

```
pip install -r requirements.txt
```

Paso 4.- Arrancar el servidor web ejecutando el script de Python "application.py":

```
python application.py
```

Tras ejecutar el paso 4, la consola queda como se muestra en la figura A2.2, como se puede ver en la última línea, el servicio web funciona en el puerto 5000, la URL de acceso será entonces una de las siguientes:

- http://localhost:5000/
- http://127.0.0.1:5000/

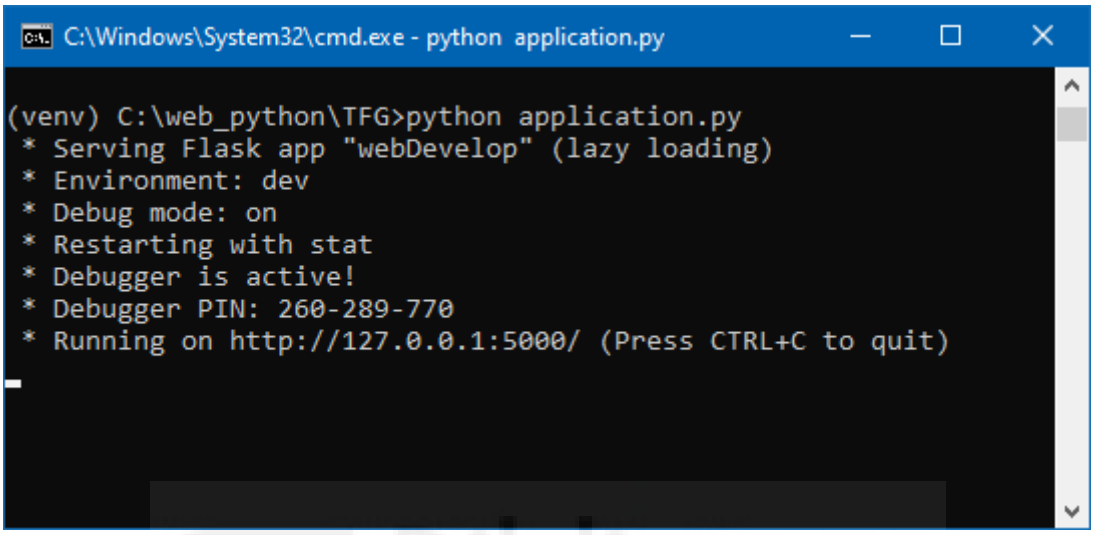


Figura A2.2.- Servidor web arrancado

La propia ventana de terminal sustenta el proceso que hace de servicio web, pulsando la combinación de teclas CTRL+C o cerrando dicho terminal se para el servidor.

Para volverlo a arrancar bastará con volver a abrir una ventana de consola en la misma carpeta y ejecutar en ella los pasos 2 (activar el entorno virtual) y 4 (arrancar el servidor web en el puerto 5000). Entre los ficheros que se proporcionan con la instalación están los denominados "**instalar.txt**" y "**arrancar.txt**" que contienen los comandos de los pasos 1, 2, 3 y 4 el primero, y solo los comandos 2 y 4 el segundo.

MUY IMPORTANTE: antes de arrancar el servidor web, el servidor de base de datos MySQL (o MariaDB) tiene que estar en funcionamiento

A2.3.- CONFIGURACIÓN

Una vez que está toda la aplicación instalada y funcionando hay que indicarle como conectar con el servidor de base de datos y cuales van a ser los datos de acceso del usuario administrador. La figura A2.3 muestra el formulario que aparece en el navegador la primera vez que nos conectamos a la aplicación, es para indicar la información de configuración

necesaria. Los tres primeros campos “Usuario BD”, “Contraseña BD” y “Nombre BD” son para indicar el nombre del usuario que tiene permisos para conectar con el servidor de base de datos, el siguiente campo es la contraseña de dicho usuario (se podrá dejar en blanco), y el tercer campo es para indicar el nombre de la nueva base de datos que se va a crear en sistema para dar soporte a la presente aplicación, obviamente, no puede haber otra base de datos ya existente con el mismo nombre.

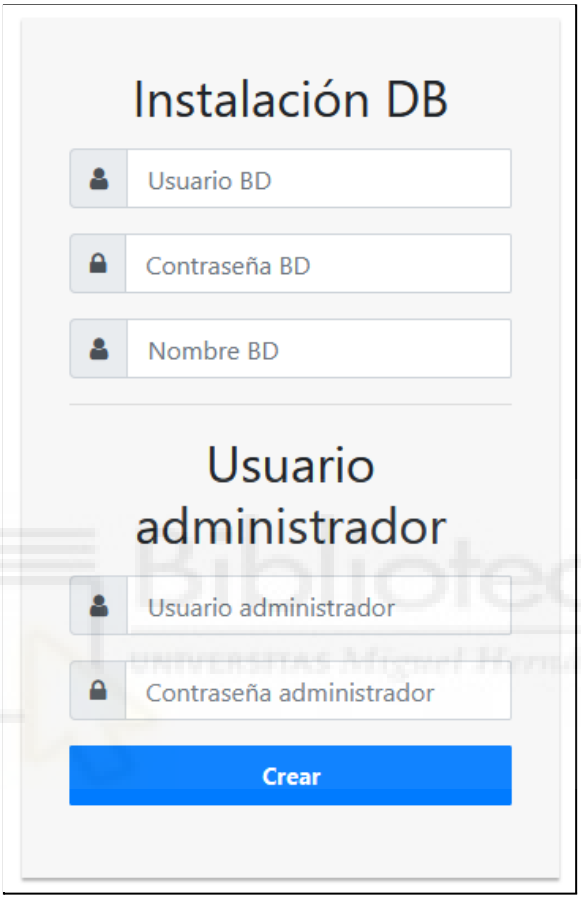


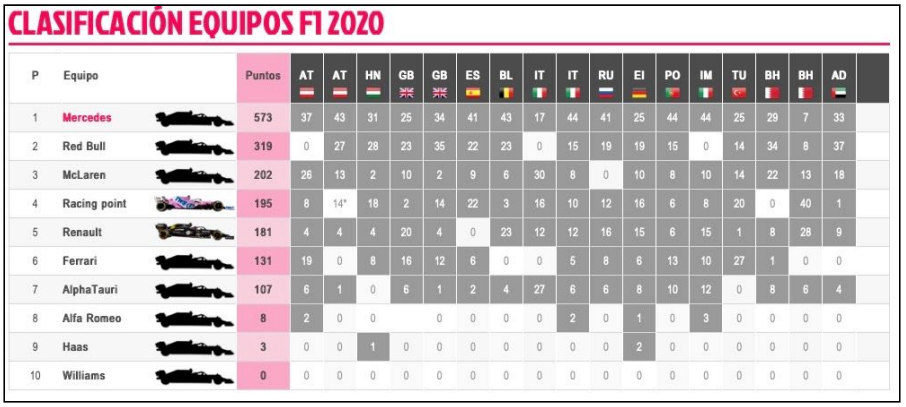
Figura A2.3.- Configuración Base de datos y usuario administrador

Los dos campos inferiores del formulario de la figura A2.3 son para indicar el nombre de usuario y la contraseña del usuario administrador de la aplicación. Toda esta información solamente habrá que proporcionarla una vez, no siendo necesaria en sucesivas ocasiones en las que se arranque el servidor y la aplicación.

Tras instalar y configurar la aplicación, el usuario administrador que se acaba de configurar ya podrá acceder a la misma y hacer uso de ella tal y como se muestra en el manual de usuario (anexo 3 de esta memoria).

Anexo 3
Manual de uso

En este apartado se va a mostrar el manual de uso de la aplicación para el usuario registrado y el administrador. Se va a ilustrar el funcionamiento usando como ejemplo la tabla que se muestra en la figura A3.1.



CLASIFICACIÓN EQUIPOS F1 2020

P	Equipo	Puntos	AT	AT	HN	GB	GB	ES	BL	IT	IT	RU	EI	PO	IM	TU	BH	BH	AD
1	Mercedes	573	37	43	31	25	34	41	43	17	44	41	25	44	44	25	29	7	33
2	Red Bull	319	0	27	28	23	35	22	23	0	15	19	19	15	0	14	34	8	37
3	McLaren	202	26	13	2	10	2	9	6	30	8	0	10	8	10	14	22	13	18
4	Racing point	195	8	14*	18	2	14	22	3	16	10	12	16	6	8	20	0	40	1
5	Renault	181	4	4	4	20	4	0	23	12	12	16	15	6	15	1	8	28	9
6	Ferrari	131	19	0	8	16	12	6	0	0	5	8	6	13	10	27	1	0	0
7	AlphaTauri	107	6	1	0	6	1	2	4	27	6	6	8	10	12	0	8	6	4
8	Alfa Romeo	8	2	0	0		0	0	0	0	2	0	1	0	3	0	0	0	0
9	Haas	3	0	0	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0
10	Williams	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura A3.1 Tabla de ejemplo de la clasificación de F1
(Fuente: https://soymotor.com/clasificacion-mundial-f1/2020)

A3.1.- INICIO

Lo primero que se ve al cargar la página inicial es un formulario de acceso a la aplicación (figura A3.2). Para entrar se deberá introducir las credenciales de acceso y, en caso que sean correctas, se permitirá el acceso a la aplicación. En caso de estar baneado el usuario, le mostrará una página indicandolo.

Figura A3.2.- Página de inicio: Formulario de acceso a la aplicación

A3.2.- USUARIO IDENTIFICADO

Una vez se ha identificado el usuario. Lo primero que se muestra es la página principal (figura A3.3), en la que se encuentran la opción de realizar el raspado de datos mediante una URL o introducir un fichero de texto en formato CSV (indicando el tipo de separador).

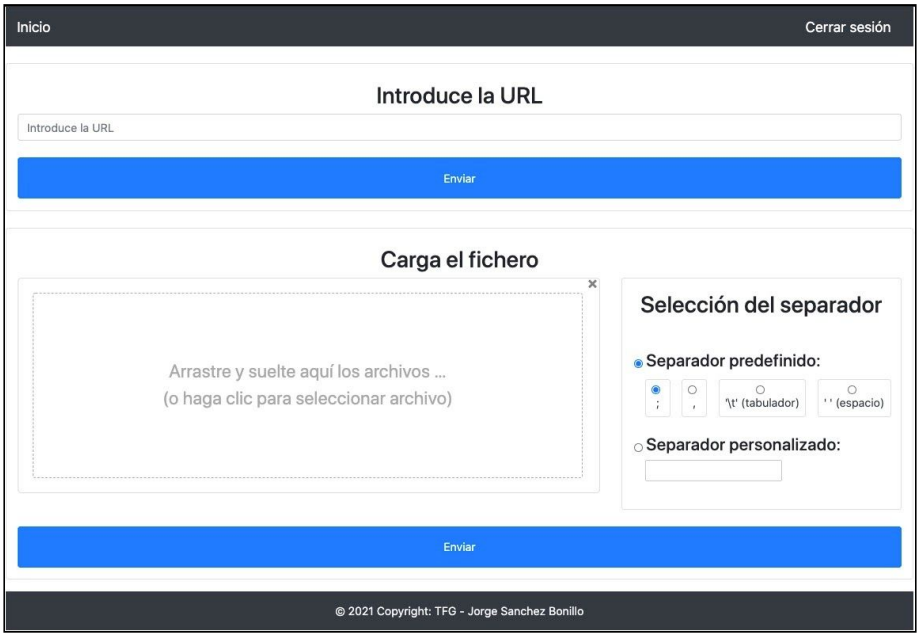


Figura A3.3 Página principal de la aplicación

La opción "*Introduce la URL*" (mostrará un mensaje de error si se pulsa el botón "Enviar" si no hay URL) comprueba que la dirección es correcta (se indicará con un mensaje si no lo es) y finalmente, se pasará a la opción de la selección de la tabla (ver figura A3.4).

La opción "*Carga el fichero*" de la página principal (figura A3.3), permite arrastrar el fichero a la caja "*Arrastre y suelte aquí...*", o seleccionar el fichero deseado del modo convencional haciendo click sobre dicha caja. Seguidamente el usuario indicará el tipo de separador del fichero CSV, predefinido o personalizado. Pulsando el botón "Enviar" se cargará dicho fichero en la aplicación (se indicará con un mensaje si el fichero no es CSV).

Continuando con la opción "*Introduce la URL*" de la página principal, la figura A3.4 muestra en la parte superior los botones "1", "2", "3", etc., que permiten cambiar entre las diferentes tablas que se hayan detectado en la URL proporcionada por el usuario, clicando en cada uno de ellos se selecciona y visualiza dicha tabla en la ventana actual.

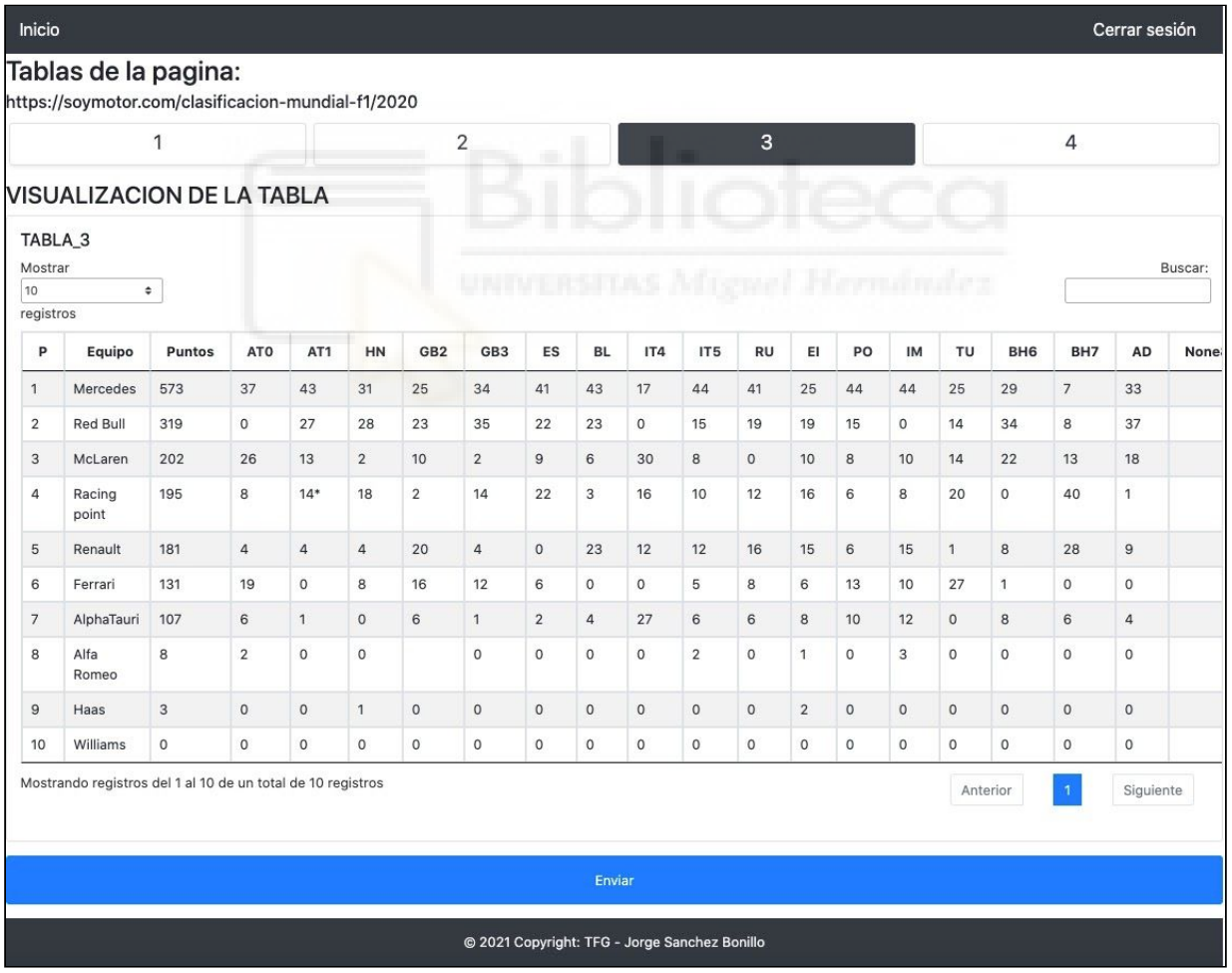


Figura A3.4 Página de selección de la tabla

Una vez seleccionada la tabla deseada, clicando el botón "Enviar" de la parte inferior se pasa a la ventana de la figura A3.5. También se llega directamente a dicha ventana desde la opción

"*Carga el fichero*" de la página principal. En este paso es posible visualizar y manipular los datos de la tabla elegida.

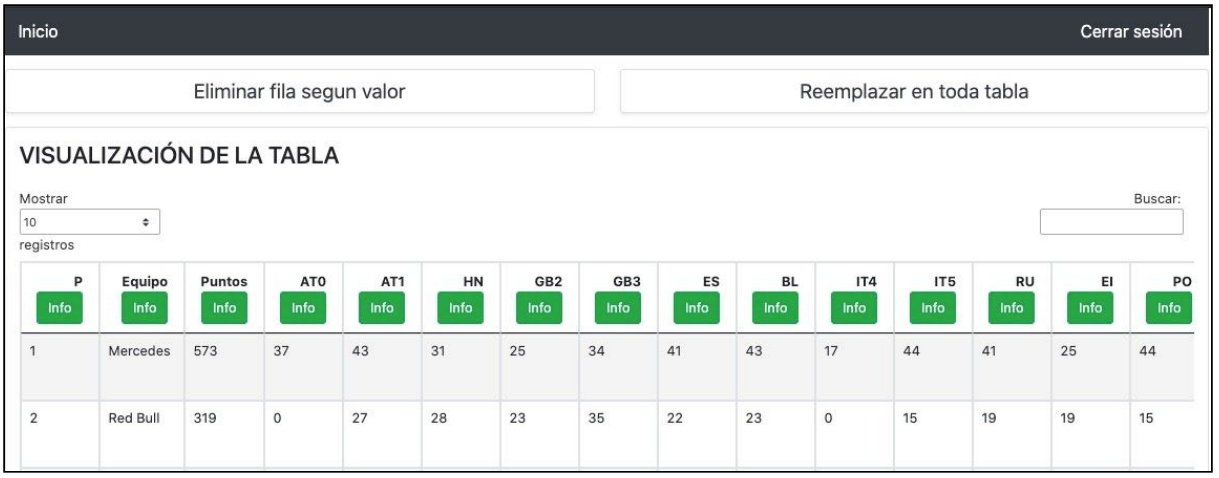


Figura A3.5 Página de la visualización de datos

Dentro de las opciones de visualización de la tabla, en la parte izquierda, se tiene un desplegable para poder mostrar el número de filas o registros que se desean. En la parte derecha hay un buscador para filtrar por aquellas filas que contienen el valor introducido.

En la parte superior (bajo el encabezado de fondo negro), están disponibles las dos opcioes siguienes:

- "*Eliminar fila según valor*": permite introducir un valor y eliminar todas aquellas filas que contengan el valor introducido.

- "*Reemplazar en toda tabla*": esta opción permite reemplazar un valor por otro en toda la tabla. Es interesante comentar que en caso de querer buscar o reemplazar un valor vacío (o nulo), basta con dejar el campo de búsqueda en blanco, sin introducir nada.

En la cabecera de la tabla (la primera fila), cada columna tiene un botón "*Info*" que abre una ventana como la que se muestra en la figura A3.6, con información sobre los datos de la columna, así como varias opciones de modificación de dichos datos. Dependiendo del tipo de datos de la columna (texto/etiquetas o números), se mostrará un tipo de información u otra. A la izquierda, el cuadro "*Datos*", en caso de que la columna sea numérica, muestra los valores máximo, mínimo y promedio. Si la columna es de tipo texto, se mostrarán la etiqueta más repetida (para ver todas las etiquetas, se debe pulsar el botón "*mostrar histograma*"). En la parte de abajo del cuadro "*Datos*" se dispone de opciones para dibujar un histograma, introduciendo un valor entre 2 y 50, se genera en el centro de la ventana el histograma con el número de segmentos (columnas) indicado. Por último, en la parte derecha, el cuadro "*Segmentar histograma*" permite crear una nueva columna en nuestra tabla de datos, resultante de hacer una segmentación de los datos de la columna actual. Se puede realizar la segmentación de dos maneras:

- Segmentar de forma automática: Los datos se segmentan o discretizan tal y como se están visualizando en el histograma, mismo número de segmentos y misma cantidad de valores por segmento.

- Segmentar de forma manual: En este caso, el usuario debe indicar, manualmente los valores límite o frontera entre los diferentes segmentos. El número de segmentos será el mismo que el del histograma (indicado en el campo "*Mostrar histograma*" a la izquierda), pero el número de elementos de cada segmento vendrá dado por los límites indicados por el usuario (no por los que se puedan ver en el histograma central).

En ambos casos el usuario debe rellenar los campos "*Etiquetas:*" del cuadro de la derecha. Al pulsar el botón "*Segmentar manual*" o "*Segmentar histograma*" se crea una nueva columna con el tipo de segmentación que corresponda.

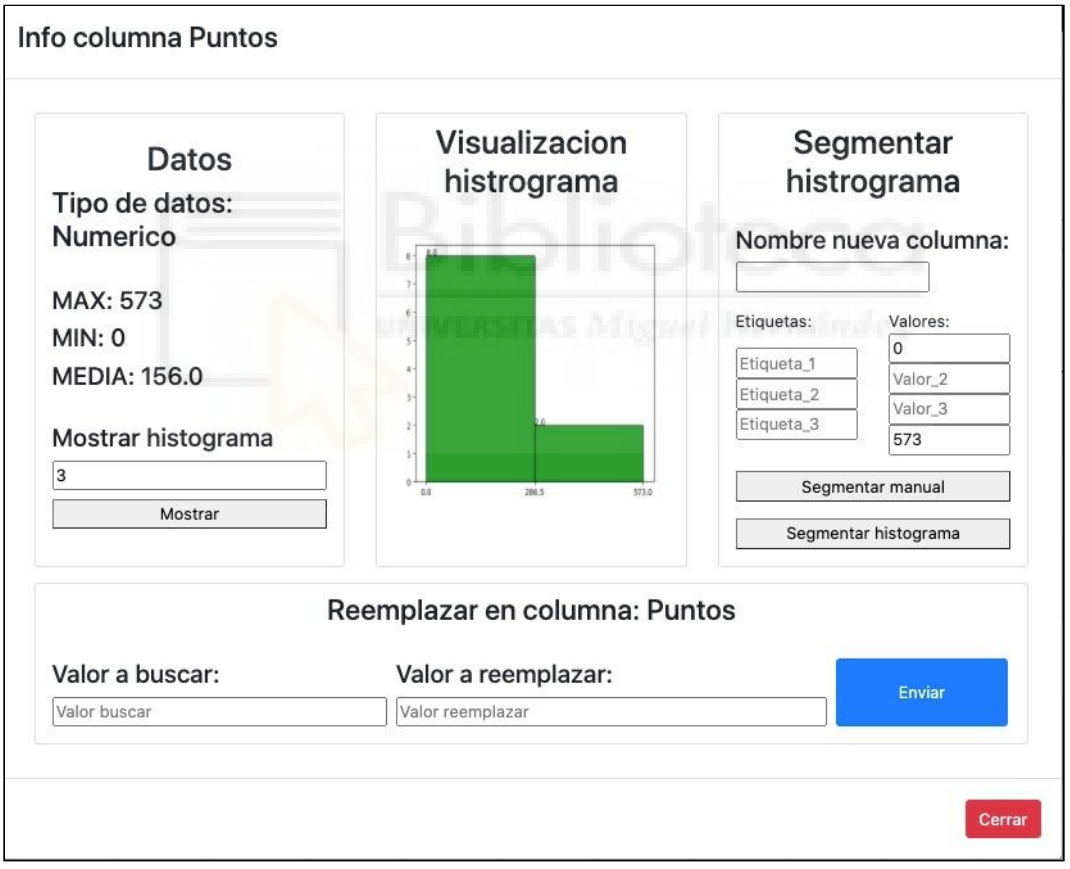


Figura A3.6.- Información de columna e histogramas

Además, en la parte inferior de la ventana se pueden buscar y reemplazar valores dentro de la columna seleccionada. Igual que en caso de "*Reemplazar en toda la tabla*", se debe introducir el valor a buscar, el valor a reemplazar y pulsar el botón "Enviar".

Volviendo a la pantalla de visualización de los datos de la tabla, en el lado derecho de la tabla (figura A3.7, habrá que hacer scroll horizontal si la tabla es demasiado ancha), se encuentra la

opción "*Editar Head*", que abre una ventana con todos los nombres de las columnas de la tabla y permite modificarlos; y también ofrece la opción de eliminar una columna de la tabla. También, en la figura A3.7 se muestra un botón "*Eliminar fila*" en cada registro de la tabla, dicha opción permite eliminar una fila concreta.

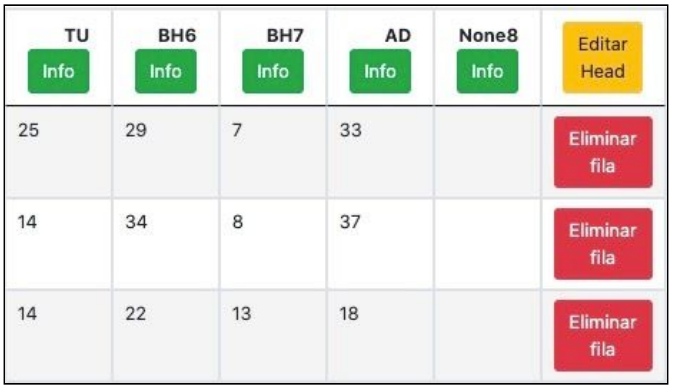


Figura A3.7 Visualización de los botones de modificar cabecera y eliminar fila

Por último, en la parte inferior de la página de visualización de datos está la opción de descarga del fichero. Para ello hay que introducir el nombre del fichero a descargar, el separador entre valores del fichero CSV y, para las columnas numéricas, el separador decimal (punto o coma). Una vez se han completado estos datos, pulsado el botón de "*Descargar CSV*", se descargan los datos de la tabla que se está manipulando.

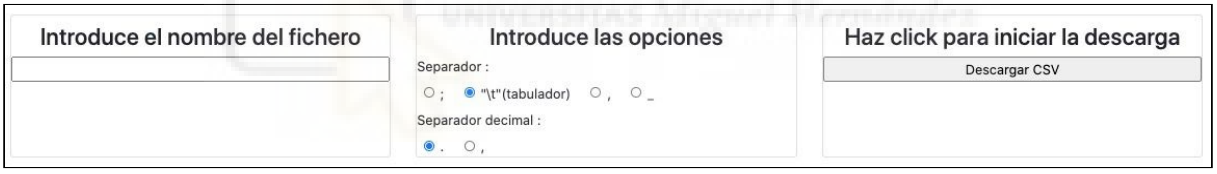


Figura A3.8.- Descarga de la tabla

Nótese que, en todo momento, en la ventana de trabajo, en el encabezado superior (fondo negro), están disponibles las opciones "*Inicio*" y "*Cerrar sesión*", la primera vuelve a la página principal de la aplicación tras la identificación de usuario (figura A3.3) donde el usuario puede poner una nueva URL donde "*scrapear*" datos o cargar un nuevo fichero CSV. La segunda opción ("*Cerrar sesión*") cierra la sesión de usuario actual y vuelve a la página de login (figura A3.2).

A3.3.- ADMINISTRADOR

El usuario administrador, una vez logueado correctamente tiene a su disposición todas las opciones que ya se han explicado en el apartado anterior, además, dispondrá de dos opciones adicionales, "*Gestión de usuarios*" y "*Gestión de logs*", ambas disponibles en la cabecera de la página (junto a "*Cerrar sesión*").

La pantalla de "*Gestión de usuarios*" (figura A3.9) dispone en la parte superior de las siguientes tres opciones:

- "*Alta usuario*": permite dar de alta a un usuario en la aplicación, muestra un formulario de entrada de datos para realizar esta tarea.
- "*Baja usuario*": permite dar de baja a un usuario en la aplicación, como el anterior, también muestra un formulario para realizar esta tarea.
- "*Baneo/Desbaneo usuario*": aparecerá un desplegable para cada estado de usuario y permitirá banear a aquellos usuarios que no lo están y viceversa.

También en el medio de la página aparece una tabla con información sobre los usuarios, información como el nombre, la contraseña cifrada, el tipo de usuario, si es administrador (A) o usuario registrado (U), y el estado del usuario, si está baneado (BAN) o no (OK). También permitirá filtrar con la barra de búsqueda de la parte superior derecha aquellas filas que contenga entre su fila el valor de búsqueda.

Figura A3.9 Gestión de usuarios

Por último, el administrador también tiene acceso al registro de logs de la aplicación (figura A3.10), que permitirá ver las acciones que ha realizado cada usuario así como poder descargar los logs según varios criterios. Para poder acceder a la descarga de los logs, se debe pulsar los botones que hay en la parte superior. Las opciones disponibles son:

- "*Usuario*": para descargar los logs pertenecientes a un usuario.
- "*Usuario y fecha*": para descargar los logs por usuario y en una fecha determinada o entre dos fechas.
- "*Fecha*": para descargar los logs de una fecha determinada o entre dos fechas.
- "*Todos*": para descargar todos los logs de la base de datos.

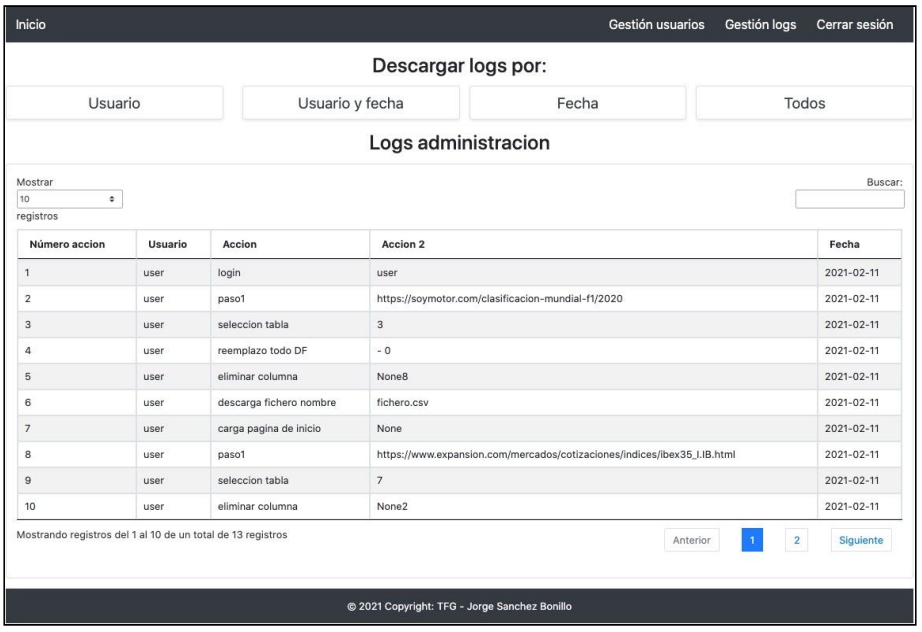


Figura A3.10 Gestión de logs.

De igual modo que en la gestión de usuarios, se podrán seleccionar cuántas filas se muestran por cada página con el selector de la parte superior izquierda de la tabla o buscar y mostrar solo aquellas filas que cumplan el requisito introducido en la barra de búsqueda.