

WID

Escuela Superior de Ingeniería

Emilio Figueras Martín

Grado en Ingeniería Informática
Universidad de Cádiz



6 de marzo de 2018

ESCUELA SUPERIOR DE INGENIERÍA
GRADO EN INGENIERÍA INFORMÁTICA



WID (Web Info Download): Web scraping, análisis de información y estadísticas para páginas webs con comentarios.

- Departamento: Ingeniería Informática
- Director del proyecto: Alberto Salguero
- Autor del proyecto: Emilio Figueras Martín

El Puerto de Santa María, 6 de marzo de 2018

Fdo: Emilio Figueras Martín

Resumen

En la actualidad hay muchas páginas webs (tiendas online, webs de noticias...) con un enfoque de interacción con los visitantes a través de sistemas de opiniones, valoraciones o comentarios. Con este conjunto de aplicaciones damos la opción de controlar el sentimiento general de los visitantes sobre cada uno de sus productos ofrecidos.

El funcionamiento de esta herramienta lo podemos dividir en tres aplicaciones:

- **Web scraping:** Una primera herramienta en el que el técnico informático de la empresa cliente o los administradores de la aplicación tendrán que llenar un archivo XML con información sobre la estructura de la web. Esta configuración es leída por la aplicación para obtener y almacenar toda la información pertinente sobre la web objetivo.
- **Análisis:** En la aplicación web se ofrece la posibilidad de realizar un análisis de los datos a través de un conjunto de consultas SQL que es mantenido y controlado por el cliente.
- **Resultado:** En esa misma aplicación web se muestran, a través de un sistema de gráficos, los datos estadísticos resultantes del análisis realizado. También se puede obtener en una hoja de cálculo la información almacenada.

Palabras claves: Web scraping, Beautiful Soup, análisis, estadísticas, feedback, sentimientos.

Índice

1. Introducción	18
1.1. Motivación	18
1.2. Objetivos	18
1.3. Alcance	20
1.3.1. Organización personal	20
1.3.2. Riesgos	21
1.3.3. Costes	23
1.3.4. Precio	24
1.3.5. Recuperación de la inversión	24
1.3.6. Recursos	25
1.4. Glosario	25
1.4.1. Definiciones	25
1.5. Descripción del sistema actual	26
2. Estado del arte	28
3. Desarrollo del calendario	31
3.1. Fases	31
3.2. Diagrama de Gantt	32
4. Descripción general del proyecto	35
4.1. Características del usuario	35
4.2. Restricciones generales	35
4.2.1. Lenguajes de programación y tecnologías	35
4.2.2. Herramientas	35
4.2.3. Sistemas operativos y hardware	36
5. Desarrollo del proyecto	38

5.1.	Metodología	38
5.2.	Requisitos	38
5.2.1.	Funcionales	38
5.2.2.	De información	42
5.2.3.	No funcionales	44
5.2.4.	Reglas de negocio	45
5.3.	Catálogo de actores	47
5.4.	Análisis del sistema	48
5.4.1.	Casos de uso	48
5.4.2.	Modelo conceptual de datos del dominio	59
5.5.	Diseño del sistema	61
5.5.1.	Diseño de la arquitectura	61
5.6.	Diseño de la interfaz de usuario	62
5.6.1.	Pantalla del programa Python de web scraping	62
5.6.2.	Pantalla de Login, registro y recuperación de la contraseña	62
5.6.3.	Pantalla general de la aplicación web y pantalla de configuración	62
5.6.4.	Pantalla para el rol 'Administrador'	62
5.6.5.	Pantalla para el rol 'Cliente Administrador'	63
5.6.6.	Pantalla para el rol 'Cliente Usuario'	64
5.7.	Diseño de componentes	65
5.8.	Implementación	65
5.8.1.	Entorno tecnológico	65
5.8.2.	Código fuente	66
5.8.3.	Calidad del Código	69
5.9.	Pruebas y validación	72
5.9.1.	Estrategia	72
5.9.2.	Descripción del entorno de pruebas	72
5.9.3.	Pruebas unitarias	73

5.9.4. Pruebas de integración	83
5.9.5. Pruebas de sistema	85
5.9.6. Pruebas de aceptación	85
6. Epílogo	91
6.1. Manual de instalación	91
6.1.1. Despliegue y configuración de la aplicación web	91
6.1.2. Despliegue y configuración de la aplicación de web scraping	91
6.2. Manual de uso	94
6.2.1. Archivo de configuración	95
6.2.2. Obtener la información de la página web objetivo	98
6.2.3. Aplicación web	107
6.3. Conclusiones y trabajo futuro	122
6.3.1. Objetivos alcanzados	122
6.3.2. Lecciones aprendidas	122
6.3.3. Trabajo futuro	123
Referencias	127

Índice de figuras

1.	Diagrama de Gantt I	32
2.	Diagrama de Gantt II	33
3.	Subsistema de gestión de usuarios.	48
4.	Subsistema del rol de Administrador.	51
5.	Subsistema del rol de Cliente Administrador.	53
6.	Subsistema del rol de Cliente Usuario.	57
7.	Modelo conceptual de datos de la aplicación.	59
8.	Modelo conceptual de datos de los usuarios.	60
9.	Pantalla de reportar informe	67
10.	Pantalla de reportar informe	67
11.	Pantalla de reportar informe	68
12.	Informe Pylint 1	70
13.	Informe Pylint 2	71
14.	Pantalla del menú de ayuda del programa de web scraping	94
15.	Codificación	95
16.	Base de datos	95
17.	URL web objetivo	96
18.	Productos	96
19.	Comentarios	98
20.	Codificación	99
21.	Codificación	99
22.	Enlace a productos	99
23.	Enlace a productos	100
24.	Enlace a productos	100
25.	Enlace a productos	101
26.	Enlace a comentarios	101
27.	Enlace a comentarios	102

28.	Enlace a comentarios	103
29.	Enlace a producto	104
30.	Enlace a producto	104
31.	Atributos de productos	105
32.	Atributos de productos	105
33.	Identificación del producto	106
34.	Identificación del producto	106
35.	Atributo tagID	106
36.	Atributo tagID	107
37.	Pantalla del Login	107
38.	Pantalla de recuperación de contraseña	108
39.	Email de recuperación de contraseña	109
40.	Pantalla de registro inicial	110
41.	Email de registro	110
42.	Pantalla de Login en dispositivo móvil	111
43.	Pantalla de configuración de usuario	112
44.	Pantalla de inicio del Administrador	112
45.	Pantalla de registro de un nuevo administrador	113
46.	Pantalla de visualización de usuarios activos	113
47.	Pantalla de edición de usuarios desde la administración	114
48.	Pantalla de inicio para el rol 'Cliente Administrador'	114
49.	Pantalla de edición de usuario para 'Cliente Administrador'	115
50.	Pantalla de registro de usuario para 'Cliente Administrador'	116
51.	Pantalla de comentarios para 'Cliente Administrador'	117
52.	Pantalla de valoración de comentarios para 'Cliente Administrador'	117
53.	Pantalla de lista de informes para 'Cliente Administrador'	118
54.	Pantalla de visualización de informe	118
55.	Pantalla de la creación/edición de informes	119

56.	Pantalla de inicio para el rol 'Cliente Usuario'	120
57.	Pantalla de reportar informe	121
58.	Pantalla de descargar comentarios	121

Índice de tablas

1.	OBJ-001 – Archivo XML como entrada de datos	18
2.	OBJ-002 – Recolección y almacenamiento de datos	18
3.	OBJ-003 – Herramienta de análisis de datos	19
4.	OBJ-004 – Administración del análisis	19
5.	OBJ-005 – Panel de control de usuarios	19
6.	OBJ-006 – Estadísticas gráficas	19
7.	OBJ-007 – Comunicación a través de la web	19
8.	OBJ-008 – Correos electrónicos automáticos	19
9.	OBJ-009 – Valoración externa de comentarios obtenidos	20
10.	OBJ-010 – Documento para minería de datos	20
11.	OBJ-011 – Realización de consultas SQL a través de la aplicación web	20
12.	R-001 – El hardware deja de funcionar	21
13.	R-002 – Subestimar la complejidad del problema	21
14.	R-003 – Página web objetivo se actualiza	21
15.	R-004 – Probabilidad de que la base de datos se desconecte	21
16.	R-005 – Probabilidad de que el servicio de envío de correos se desconecte	22
17.	R-006 – Probabilidad de que la actividad profesional requiera mayor dedicación temporal	22
18.	FRQ-001 – Creación de la base de datos	38
19.	FRQ-002 – Reinicio de la base de datos	38
20.	FRQ-003 – Seleccionar modo detallado	39
21.	FRQ-004 – Indicar el nombre del archivo XML	39
22.	FRQ-005 – Registro en la página web	39
23.	FRQ-006 – Registrar usuarios del cliente	39
24.	FRQ-007 – Modificar datos de usuario	39
25.	FRQ-008 – Valorar comentarios	39
26.	FRQ-009 – Modificar valoraciones	39

27.	FRQ-010 – Descargar comentarios	40
28.	FRQ-011 – Filtrar comentarios	40
29.	FRQ-012 – Ignorar palabras en comentarios	40
30.	FRQ-013 – Crear informes	40
31.	FRQ-014 – Editar informes	40
32.	FRQ-015 – Desactivar informes	40
33.	FRQ-016 – Eliminar informes	40
34.	FRQ-017 – Ver gráficos de informes	41
35.	FRQ-018 – Reportar error en informe	41
36.	FRQ-019 – Elegir tipo de gráfico en informe	41
37.	IRQ-001 – Información de los <i>usuarios</i>	42
38.	IRQ-002 – Información de los <i>productos</i>	42
39.	IRQ-003 – Información de los <i>comentarios</i>	42
40.	IRQ-004 – Información de los <i>análisis</i>	42
41.	IRQ-005 – Información de los <i>informes</i>	43
42.	NRQ-001 – Usabilidad	44
43.	NRQ-002 – Escalabilidad	44
44.	NRQ-003 – Funcionalidad	44
45.	NRQ-004 – Seguridad	44
46.	ACT-001 – Catálogo de actores - Administrador	47
47.	ACT-002 – Catálogo de actores - Cliente Administrador	47
48.	ACT-003 – Catálogo de actores - Cliente Usuario	47
49.	CP-001 – Argumentos de entrada	73
50.	CP-002 – Creación de la base de datos	74
51.	CP-003 – Datos obligatorios del archivo de configuración	76
52.	CP-004 – Funcionamiento de la función <i>is_repit</i>	80
53.	CP-005 – Login	80
54.	CP-006 – Registro y validación	80

55.	CP-007 – Olvido de contraseña	81
56.	CP-008 – Cambio de contraseña	81
57.	CP-009 – Activar usuario	81
58.	CP-010 – Modificar datos de otro usuario	81
59.	CP-011 – Registrar administrador	81
60.	CP-012 – Registrar cliente usuario	82
61.	CP-013 – Modificar datos de otro usuario desde 'admin_customer'	82
62.	CP-014 – Puntuar comentarios	82
63.	CP-015 – Generar Excel	82
64.	CP-016 – Crear y editar informes	83
65.	CP-017 – Mostrar gráficos	83
66.	CP-018 – Reportar error en un informe	83
67.	CP-019 – Prueba completa de la aplicación de web scraping	84
68.	CP-020 – Prueba completa de la aplicación web para administrador	85
69.	CP-021 – Prueba completa de la aplicación web para cliente administrador	85
70.	CP-022 – Prueba completa de la aplicación web para cliente usuario	85

1. Introducción

En este apartado se explican los aspectos preliminares que se deben tener en cuenta para entender el desarrollo del proyecto.

1.1. Motivación

La elección de este Trabajo de Fin de Grado viene dada por el hecho de ser un proyecto que reúne aspectos muy diferentes de la informática y que a la vez me resultan interesantes de conocer o profundizar. Estos son:

- Web Scraping: Una técnica que me llamaba la atención descubrir pero que nunca había utilizado. Además, elegí Python como lenguaje a utilizar para la aplicación del web scraping porque es un lenguaje al que sólo me he acercado brevemente y me gustaba el hecho de poder hacer más uso de él y conocerlo mejor.
- Conocer información teórica sobre diferentes métodos de análisis de información, desde algunas más rudimentarias como consultas directas en SQL hasta la utilización de plataformas como Weka.
- Programación web: Algo que ya conocía de antemano y que me era de utilidad seguir profundizando, concretamente, en el uso del framework CodeIgniter. Este es un framework que había estudiado con anterioridad y que utilicé en mi actual empleo.

1.2. Objetivos

Código	OBJ-001
Descripción	Archivo XML como entrada de datos
Importancia	Muy importante
Estado	Realizado
Estabilidad	Alta

Tabla 1: OBJ-001 – Archivo XML como entrada de datos

Código	OBJ-002
Descripción	Recolección y almacenamiento de datos
Importancia	Muy importante
Estado	Realizado
Estabilidad	Alta

Tabla 2: OBJ-002 – Recolección y almacenamiento de datos

Código	OBJ-003
Descripción	Herramienta de análisis de datos
Importancia	Muy importante
Estado	Realizado
Estabilidad	Alta

Tabla 3: OBJ-003 – Herramienta de análisis de datos

Código	OBJ-004
Descripción	Administración del análisis
Importancia	Muy importante
Estado	Realizado
Estabilidad	Alta

Tabla 4: OBJ-004 – Administración del análisis

Código	OBJ-005
Descripción	Panel de control de usuarios
Importancia	Importante
Estado	Realizado
Estabilidad	Alta

Tabla 5: OBJ-005 – Panel de control de usuarios

Código	OBJ-006
Descripción	Estadísticas gráficas
Importancia	Importante
Estado	Realizado
Estabilidad	Alta

Tabla 6: OBJ-006 – Estadísticas gráficas

Código	OBJ-007
Descripción	Comunicación a través de la web
Importancia	Opcional
Estado	Realizado
Estabilidad	Media

Tabla 7: OBJ-007 – Comunicación a través de la web

Código	OBJ-008
Descripción	Correos electrónicos automáticos
Importancia	Importante
Estado	Realizado
Estabilidad	Alta

Tabla 8: OBJ-008 – Correos electrónicos automáticos

Código	OBJ-009
Descripción	Valoración externa de comentarios obtenidos
Importancia	Importante
Estado	Realizado
Estabilidad	Alta

Tabla 9: OBJ-009 – Valoración externa de comentarios obtenidos

Código	OBJ-010
Descripción	Documento para minería de datos
Importancia	Opcional
Estado	Realizado
Estabilidad	Alta

Tabla 10: OBJ-010 – Documento para minería de datos

Código	OBJ-011
Descripción	Realización de consultas SQL a través de la aplicación web
Importancia	Muy importante
Estado	Realizado
Estabilidad	Alta

Tabla 11: OBJ-011 – Realización de consultas SQL a través de la aplicación web

1.3. Alcance

1.3.1. Organización personal

El proyecto se ha realizado de forma individual. La única organización necesaria es con el director del proyecto. La comunicación es llevada a cabo a través de correos electrónicos y reuniones presenciales concertadas.

Además, se ha hecho uso de la plataforma web Trello para organizar, detallar y controlar la lista de tareas a realizar a través de las tarjetas virtuales que ofrece este software de administración de proyectos.

Como repositorio online se ha utilizado la plataforma GitHub en donde se ha ido almacenando y actualizando el desarrollo del proyecto.

1.3.2. Riesgos

ID Riesgo: R-001	Probabilidad: Baja.
Impacto	Alto.
Descripción	El hardware deja de funcionar.
Contexto	No podría continuar la realización del proyecto.
Reducción / Supervisión	Realizar copias de seguridad para poder continuar trabajando en otro equipo provisionalmente.
Gestión / Plan de contingencia	Almacenar el código creado en repositorios online como GitHub.

Tabla 12: R-001 – El hardware deja de funcionar

ID Riesgo: R-002	Probabilidad: Media
Impacto	Medio
Descripción	Subestimar la complejidad del problema.
Contexto	Conocimientos acerca del problema.
Reducción / Supervisión	Realizar un análisis previo del problema y de las herramientas que voy a utilizar para abarcarlo de forma más adecuada.
Gestión / Plan de contingencia	Definir unos límites iniciales alcanzables para que en caso de que la complejidad se expanda, se pueda retroceder hasta un punto funcional y utilizable.

Tabla 13: R-002 – Subestimar la complejidad del problema

ID Riesgo: R-003	Probabilidad: Alta
Impacto	Alto
Descripción	La página web objetivo se actualiza y modifica su código HTML.
Contexto	Si la web objetivo modifica su código HTML será necesario actualizar el archivo de configuración
Reducción / Supervisión	Control de errores en la ejecución del web scraping para paralizar su ejecución y pasar a corregir dicho código.
Gestión / Plan de contingencia	Correo electrónico predefinido para avisar al cliente de estos hechos.

Tabla 14: R-003 – Página web objetivo se actualiza

ID Riesgo: R-004	Probabilidad: Baja
Impacto	Alto
Descripción	Probabilidad de que la base de datos se desconecte.
Contexto	El servidor de base de datos actual es un portátil personal.
Reducción / Supervisión	Copia de seguridad periódica de la base de datos.
Gestión / Plan de contingencia	Migrar a un host de pago que apporte mayor seguridad.

Tabla 15: R-004 – Probabilidad de que la base de datos se desconecte

ID Riesgo: R-005	Probabilidad: Baja
Impacto	Bajo
Descripción	Probabilidad de que el servicio de envío de correos falle
Contexto	Se utiliza el servicio SMTP gratuito de Gmail.
Reducción / Supervisión	La cuenta de correo que envía los mensajes es la misma que recibe los avisos de nuevos registros y que se utiliza para establecer contacto con los clientes, por lo que su revisión debe de ser periódica.
Gestión / Plan de contingencia	Se puede utilizar las cuentas de correo de los administradores, en caso de ser estrictamente necesario, para enviar emails en el momento en el que el servicio de Gmail esté inactivo.

Tabla 16: R-005 – Probabilidad de que el servicio de envío de correos se desconecte

ID Riesgo: R-006	Probabilidad: Alta
Impacto	Medio
Descripción	Probabilidad de que la actividad profesional requiera mayor dedicación temporal
Contexto	A la vez que se inicia el proyecto también comienzo un nuevo empleo.
Reducción / Supervisión	Aprovechar momentos de menor carga profesional para dedicar más tiempo al proyecto e ir adaptando los horarios.
Gestión / Plan de contingencia	Tener un tiempo mínimo de dedicación al proyecto en los fines de semana para los momentos con mayor carga profesional.

Tabla 17: R-006 – Probabilidad de que la actividad profesional requiera mayor dedicación temporal

1.3.3. Costes

Distinguiremos entre costes monetarios y costes temporales.

- **Costes monetarios.** El coste de transporte para las reuniones y el coste del hardware necesario para el alojamiento tanto de la base de datos como de la aplicación web. En la realización del proyecto el coste monetario del alojamiento tanto para el servidor Apache como de las bases de datos MySQL es gratuito, puesto que se ha utilizado el propio portátil como servidor. A día de hoy no hay ningún hosting que ofrezca, de forma gratuita, el servicio de bases de datos necesario para la completa ejecución de la aplicación, por lo tanto, para el paso a la ejecución profesional del proyecto sería necesario la contratación de un hosting que ofrezca bases de datos ilimitadas. Lo ideal sería un 'Cloud Hosting' por la escalabilidad de los recursos que estos ofrecen y que se necesita para este proyecto. El coste monetario inicial sería de unos 20 euros mensuales de inicio, que aumentarían a la par que aumenten el número de clientes de la aplicación.
- **Costes temporales.** A continuación se detallan en distintas fases los costes temporales.
 - **Asignación del proyecto.** Periodo de tiempo en el que tanteo las diferentes posibilidades en cuanto a proyectos a realizar hasta que finalmente me decido por el proyecto actual.
 - **Planteamiento del proyecto.** Reuniones iniciales con el director del proyecto para plantear la base y objetivos principales del proyecto en cuestión.
 - **Análisis del problema.** Investigación y documentación sobre las herramientas necesarias para la realización del proyecto, así como el uso de estas.
 - **Diseño de la base de datos del web scraping.** Se diseña la estructura de la base de datos que almacenar los datos obtenidos del web scraping en la web objetivo, esta base de datos tiene una estructura que es fija y necesaria para todos los clientes, y otra parte que es variable y dependerá del archivo de configuración del cliente en cuestión.
 - **Diseño de la aplicación Python.** Paralelamente al diseño de la base de datos se diseña la aplicación Python encargada del web scraping en la página web objetivo.
 - **Implementación del web scraping.** Una vez diseñada la aplicación en Python se pasa a implementar la base de datos definitiva que irá asociada a la aplicación puesto que será la que cree la estructura de la base de datos para cada cliente.
 - **Diseño de la base de datos de la aplicación web.** Se diseña la estructura de la base de datos exclusiva de la aplicación web.
 - **Diseño de la aplicación web.** Paralelamente al diseño de la base de datos se va diseñando el funcionamiento y las características de la aplicación web, la cual utilizará una base de datos para el control de usuarios y otra base de datos, la de los datos del web scraping, para el análisis y la muestra de estadísticas.
 - **Implementación de la aplicación web.** Se pasa a implementar lo diseñado con respecto a la aplicación web teniendo en cuenta que también interactuará con la base de datos donde se almacenan los datos del web scraping.
 - **Pruebas.** Cada versión completada de la aplicación del web scraping se probaba a través de su ejecución por la terminal sobre dos webs diferentes, además también se ha realizado pruebas unitarias a través del framework Unit Testing de Python.

Posteriormente, para las pruebas de la aplicación web se utilizan los datos obtenidos en las pruebas de la aplicación del web scraping. Las pruebas realizadas están descritas en la sección 5.9.

- **Documentación** Esta fase empieza prácticamente a la vez de comenzar el proyecto y se prolonga hasta el final de este, actualizando los contenidos conforme el proyecto avanza.

1.3.4. Precio

Es difícil realizar un calculo del coste del proyecto al estar separado en dos partes, la aplicación web y la aplicación ejecutable que realiza el web scraping. Utilizando la web <http://howmuchdoesawebsitecost.com/> para calcular el coste estimado del proyecto nos indica que este está entre 18500 y 22000 dólares.

Si buscamos en exclusiva el precio de la aplicación que realiza el web scraping, sin tener en cuenta la aplicación web, podemos observar el precio de aplicaciones similares que se distribuyen por Internet, como es el caso de <http://www.webextractor.com/purchase.htm>, una aplicación de escritorio que extrae información de una página web objetivo y cuyo precio es de 200 dólares.

1.3.5. Recuperación de la inversión

Para recuperar la inversión realizada la idea de negocio del proyecto consiste en ofrecer el conjunto de aplicaciones como una venta de pago mensual dando al cliente dos opciones de compra:

- **Versión completa:** Esta opción sería la más cara y consistiría en que el cliente solo debe indicar al administrador la web objetivo y los datos que desea obtener. Los administradores se encargarían de la realización del archivo de configuración y del mantenimiento de este. Igualmente sería el propio cliente a través del usuario con el rol de "cliente administrador" el encargado de realizar las consultas que luego se reflejen en estadísticas gráficas para el usuario con el rol de "cliente usuario".
- **Versión reducida:** Esta sería una opción más económica en el que se ofrece al cliente el manual de instrucciones del archivo de configuración, disponible en la sección 6.2, así como los datos de la base de datos que se le ha asignado. De esta forma será el propio cliente el que deba llenar y mantener el archivo de configuración. Al igual que en la versión anterior se le asignaría de inicio un usuario con el rol de "cliente administrador" para la realización de las consultas así como la creación de usuarios dentro de su organización.

1.3.6. Recursos

▪ Recursos humanos:

- **Trabajo y desarrollo.** Proyecto individual con un único realizador.
- **Cliente.** Proyecto destinado a empresas interesadas en la opinión y las sensaciones de sus clientes o grupo de personas con respecto a los productos que manejan o la marca que ofrecen.

▪ Recursos materiales:

- **Espacio de trabajo.** Al ser un proyecto individual el principal espacio de trabajo para el desarrollo del proyecto ha sido la habitación del autor del proyecto.
- **Equipamiento.** Se ha utilizado el portátil personal así como la infraestructura necesaria para el desarrollo ofrecida en la casa del autor del proyecto, esto es, conexión a internet y electricidad.
- **Material básico.** El único sistema operativo utilizado ha sido Debian 8, y como software para el desarrollo se ha utilizado el editor de texto *Sublime Text* para toda la programación.

Para la aplicación web se ha utilizado el framework *CodeIgniter 3* que utiliza el lenguaje de programación *PHP*, mientras que para la aplicación de web scraping se ha utilizado el lenguaje de programación *Python 3* con la biblioteca de web scraping *Beautiful Soup*. Además, para las pruebas en Python se ha utilizado el framework *Unit Testing*.

Como hardware de pruebas para la realización del proyecto se ha utilizado el mismo equipo de desarrollo donde se ha instalado un servidor de base de datos *MySQL* así como un servidor *HTTP Apache*.

Para la documentación se ha utilizado el sistema de composición de textos *LATEX*.

1.4. Glosario

1.4.1. Definiciones

- Web scraping: Técnica software utilizada para la extracción de información de páginas web. Es un proceso de recopilación de información de la Web de forma automática.
- Análisis de sentimiento: Proceso para determinar si una frase contiene una opinión, negativa o positiva, sobre un concepto o entidad concreta.
- Python: Lenguaje de programación interpretado, multiparadigma y multiplataforma.
- PHP: Lenguaje de programación situado en el servidor y diseñado originalmente para el desarrollo web de contenido dinámico.
- Framework: Estructura conceptual y tecnológica de asistencia definida que puede servir de base para la organización y desarrollo de software. Sirve de ayuda para desarrollar y unir los diferentes componentes de un proyecto.
- CodeIgniter: Es un framework de código abierto para el desarrollo web con PHP.

- Beautiful Soup: Biblioteca de Python utilizada para realizar web scraping. Sirve para el análisis de cualquier documento etiquetado.
- Trello: Software de administración de proyectos con interfaz web. Registra las actividades a través de tarjetas virtuales utilizándolas para organizar y distribuir las diferentes tareas a realizar dentro de un proyecto.
- CRM: Las siglas vienen de Customer Relation Management (Gestión de Clientes) y es un software muy común en las empresas que se utiliza para administrar, principalmente, la relación comercial con los clientes y el proceso posventa.

1.5. Descripción del sistema actual

Actualmente muchas páginas webs cuenta con la opción de comentar el contenido que ofrecen. Muchas de ellas cuentan con un sistema de valoración del contenido que se agrega junto al comentario. La aplicación que aquí se ofrece da la posibilidad de generar análisis automatizados de los comentarios y los productos ofrecidos por la web objetivo, para de esta forma obtener estadísticas basadas en estos datos. Además, también es capaz de realizar un análisis de sentimientos de los comentarios independientemente de que estos tengan o no un sistema de valoración asociado. El usuario de la aplicación podrá almacenar y analizar cualquier información disponible en la página web objetivo, sin la necesidad de tener ningún acceso especial sobre el contenido.

2. Estado del arte

El desarrollo del conjunto del proyecto es algo que no se puede ver ofrecido en el mercado online.

En este proyecto tenemos por un lado el sistema de estadísticas de datos de la web objetivo, estadísticas de los comentarios y la posibilidad de obtener un archivo preparado para el análisis de sentimientos a través de herramientas externas. La ventaja que da este proyecto es que toda esta información se obtiene sin necesidad de que exista relación alguna entre el cliente y la web objetivo, puesto que todos los datos obtenidos para su análisis están públicos en el código de la web objetivo.

Por otro lado tenemos el programa que realiza el web scraping cuya funcionalidad sí que está recogida en diferentes proyectos en el mercado, como por ejemplo el ya mencionado anteriormente <http://www.webextractor.com/purchase.htm>, que ofrece al cliente una herramienta gráfica para la obtención de datos de una página web.

Por lo tanto, mientras que en el mercado existen varias herramientas que pueden proporcionar características similares a algunas que realiza este proyecto, se puede decir que no hay una herramienta que proporcione todo lo que este proyecto ofrece en su conjunto.

El proyecto en su conjunto se compone de la utilización de un administrador regular de procesos en segundo plano como 'cron', la aplicación de web scraping en Python, y la aplicación web. Con todo esto, tenemos un programa que se ejecutará cada cierto tiempo y que recorrerá una página web objetivo para obtener los datos deseados así como los comentarios que los usuarios de la web objetivo hagan sobre cada uno de los productos. Estos datos se recogerán sin la necesidad de tener ningún permiso ni acceso especial a la página web. Esta información almacenada en una base de datos se utilizará para la aplicación web. A través de la aplicación web el cliente podrá dar su valoración personal, entre positiva o negativa, de cada comentario almacenado, para así obtener una hoja de cálculo con los comentarios y una valoración fiable de estos que pueda ser utilizada para realizarle un análisis de sentimientos a todos los comentarios a través de herramientas como Weka. La realización de análisis de sentimientos es el objetivo principal de la generación de la hoja de cálculo, pero no el único, puesto que en dicha hoja de cálculo aparecerá toda la información almacenada de la página web objetivo y para su generación se podrá filtrar por atributos que contengan un dato deseado, si así se prefiere. La aplicación web también proporciona tres informes predefinidos los cuales son:

- Número de comentarios por producto: Un gráfico de barras que muestra el número de comentarios por producto de los productos más comentados.
- Comentarios positivos vs negativos: Gráfico de pastel que muestra los comentarios que contienen la palabra 'buen' contra los comentarios que contiene la palabra 'mal'.
- Comentarios favorables vs desfavorables: Gráfico de paste que muestra, de los comentarios que han sido valorados personalmente por el cliente, el número de comentarios votados positivamente contra el número de comentarios valorados negativamente.

Además de estos tres informes iniciales, los usuarios podrán crear sus propios informes a través de introducir consultas SQL siguiendo las instrucciones de creación de informes que se detallan en la vista correspondiente de la aplicación web.

Las aplicaciones de extracción de datos que ofrecen características similares a la de este proyecto, como la ya mencionada 'Web Data Extractor', proporcionan una interfaz gráfica más sencilla para el usuario común puesto que es una aplicación ejecutable donde se puede obtener los datos de la web objetivo en un archivo CSV a golpe de click, es decir, en el propio programa nos muestra la página web objetivo y vamos seleccionando la información que nos interesa. La ventaja de WID con respecto a otras herramientas disponibles en el mercado es que cuenta con una mayor flexibilidad a la hora de manejar los datos que se obtienen, así como que WID es una opción multiplataforma para el cliente, puesto que el cliente puede contratar la opción en la que la obtención de datos la realizamos en nuestro propio servidor, y lo único que se necesita para poder disfrutar de las ventajas que ofrece WID es un navegador web con el que acceder a la aplicación web desde la que manejar los resultados. Otras herramientas, como 'Web Data Extractor', son exclusivas para sistemas operativos Windows.

3. Desarrollo del calendario

3.1. Fases

El proyecto se dividirá en diferentes fases cuya realización puede ocurrir de forma concurrente puesto que algunas de estas están relacionadas entre sí, afectando los cambios en una a otras fases. Las fases planteadas son las siguientes:

- **Análisis del problema:** Planteamiento de la idea inicial, discusión con el director del proyecto sobre los diferentes caminos y herramientas para realizarla y estudio de estas herramientas para proceder a la elección de las más conveniente.
- **Diseño de la base de datos del web scraping:** Se comprueba que se necesita un servidor de base de datos y una base de datos para almacenar la información obtenida a través del web scraping de cada cliente. Se diseña la estructura de la base de datos que será común a todos los clientes teniendo en cuenta la información necesaria y obligatoria que tendrán que almacenar todos los clientes. Luego, dentro de cada una de las tablas fijas, se tiene en cuenta una parte variable que consiste en atributos de estas tablas que cada uno de los clientes decidan almacenar dependiendo de lo que ofrezca la web objetivo.
- **Diseño del programa que realiza el web scraping:** Conforme se va diseñando la base de datos se va, a su vez, diseñando parte de la estructura y características que tendrán el software encargado de la obtención de los datos de la web objetivo a través de la realización de web scraping.
- **Implementación de la base de datos del web scraping:** Una vez diseñada la base de datos se procede a su implementación para darle posterior uso en la implementación del programa encargado de obtener los datos.
- **Implementación del programa que realiza el web scraping:** Después de su diseño se pasa a la implementación del programa que realiza el web scraping así como se implementa la opción de generar la base de datos directamente desde este programa. Durante esta fase se irá implementando diferentes funcionalidades del programa en cuestión.
- **Pruebas del programa que realiza el web scraping:** Concurrentemente a la implementación se realizan pruebas de cada una de las funcionalidades que son implementadas. Estas pruebas servirán tanto para corregir errores como para observar modificaciones necesarias para mejorar el funcionamiento de las funcionalidades.
- **Diseño de la base de datos de la aplicación web:** La aplicación web, además de conectarse con la base de datos de cada cliente, también necesita una base de datos propia que almacene los usuarios y los roles. Se diseña la estructura de esta base de datos conforme al estudio y análisis planteado inicialmente.
- **Diseño de la aplicación web:** Teniendo claro los diferentes roles de usuarios con los que vamos a trabajar así como los datos que nos podrá proporcionar el programa encargado del web scraping se pasará a diseñar las funcionalidades e interfaces que se ofrecerá al cliente en la aplicación web para cada uno de los diferentes roles disponibles.

- **Implementación de la base de datos de la aplicación web:** Una vez implementada la base de datos relacionada con el programa de web scraping se implementa la base de datos de la aplicación web que servirá para el control de usuarios de esta.
- **Implementación de la aplicación web:** A la vez que se va diseñando las funcionalidades e interfaces de la aplicación web se pasa a la implementación de estas.
- **Pruebas de la aplicación web:** Conforme se va implementando las diferentes funcionalidades de la aplicación web se irá elaborando las pruebas que corroboren su correcto funcionamiento.
- **Documentación:** Desde el inicio hasta el final del proyecto se ha ido escribiendo y actualizando la documentación donde se relata los detalles del desarrollo.

3.2. Diagrama de Gantt

En primer lugar se puede observar la división de tareas y fases situadas en el diagrama de Gantt y sus fechas:

Nombre	Fecha de inicio	Fecha de fin
● Preparación e investigación	19/09/16	13/10/16
● Análisis del problema	13/10/16	3/11/16
▣ ● Aplicación web scraping	3/11/16	30/07/17
● Diseño de la base de datos	3/11/16	21/11/16
● Diseño del algoritmo	14/11/16	29/01/17
● Implementación de la BD	27/01/17	5/02/17
● Implementación del algoritmo	6/02/17	30/06/17
● Pruebas	1/03/17	30/07/17
▣ ● Aplicación web	28/07/17	8/01/18
● Diseño de la base de datos	28/07/17	6/08/17
● Diseño de la web	21/08/17	25/09/17
● Implementación de la BD	22/09/17	1/10/17
● Implementación de la web	1/10/17	25/12/17
● Pruebas	8/12/17	8/01/18
● Pruebas conjuntas y cambios	8/01/18	11/02/18
● Documentación	1/11/16	25/02/18

Figura 1: Diagrama de Gantt I

Y aquí el diagrama de Gantt:

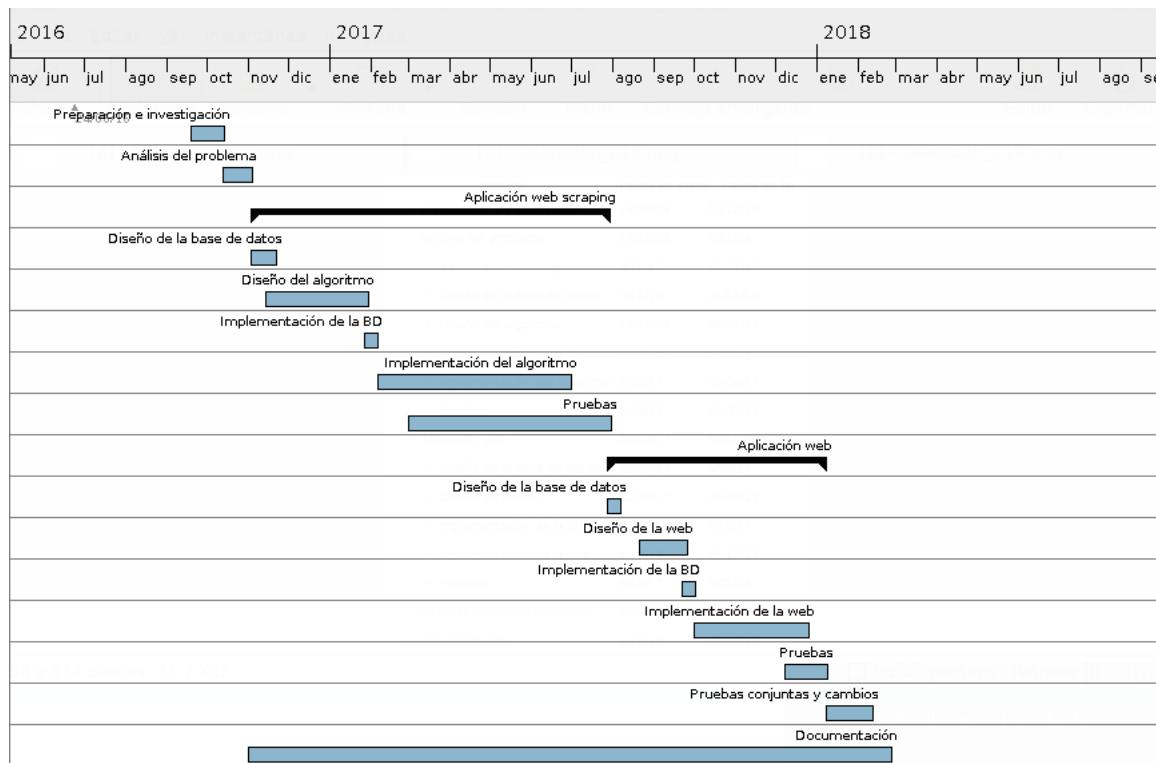


Figura 2: Diagrama de Gantt II

4. Descripción general del proyecto

4.1. Características del usuario

La aplicación va dirigida a empresas que cuente con diferente personal dentro de ella, representado esto en el proyecto a través de los diferentes roles de usuarios disponibles en la aplicación web. Deberá existir dos tipos de usuario dentro de cada cliente, uno es la persona encargada de realizar las consultas sobre los datos obtenidos que darán lugar a las estadísticas deseadas y que, además, en caso de haber seleccionado la versión reducida de la aplicación deberá encargarse de completar y mantener el archivo de configuración de la herramienta de web scraping. Esta persona debe tener conocimientos informáticos. El otro tipo de usuario que se espera que use el proyecto será el encargado de visualizar las estadísticas realizada. Para esta tarea no se requiere conocimientos ni habilidades informáticas avanzadas.

4.2. Restricciones generales

4.2.1. Lenguajes de programación y tecnologías

El lenguaje de programación elegido para la aplicación de web scraping fue Python. Las razones por la que se ha elegido Python ha sido por disponer de una herramienta como BeautifulSoup que está hecha específicamente para tareas de web scraping, además de ser un lenguaje limpio y sencillo. También me resultaba interesante entrar con mayor profundidad en un lenguaje el cual solo había hecho uso práctico de forma muy básica.

Por otro lado, en la aplicación web se ha utilizado el lenguaje de programación PHP con el framework CodeIgniter 3. La utilización de PHP junto con CodeIgniter viene de la necesidad de tener una aplicación con una interfaz gráfica similar a la de un CRM y con unas características concreta que requerían la integración con diferentes componentes. Por esto, además de estas dos herramientas, también se ha utilizado Bootstrap para el desarrollo de la interfaz gráfica, Chart.js para la visualización de las estadísticas gráficas, JQuery para las funciones en javascript de la aplicación web y WOW.js para la animación del scroll. PHP es un lenguaje que encaja perfectamente en el objetivo deseado y, además, a la hora de desarrollar disponía de unos conocimientos previos que facilitaban la programación de la aplicación web utilizando este lenguaje de programación y las diferentes tecnologías mencionadas.

4.2.2. Herramientas

Tanto para el desarrollo de la aplicación de web scraping como para el desarrollo de la aplicación web se ha utilizado el editor de texto Sublime Text, el cual permite codificar en numerosos lenguajes, utilizando posteriormente para ver el resultado de lo programado la terminal y el navegador web respectivamente, siendo Firefox el navegador web habitual.

Para redactar este documento se ha utilizado el sistema de composición de textos L^AT_EX.

Para la organización de tareas se ha hecho uso de la plataforma web Trello.

4.2.3. Sistemas operativos y hardware

El sistema operativo utilizado para el desarrollo de este proyecto es Debian 8 ("jessie"), en un ordenador portátil ASUS A55V que cuenta con una memoria RAM de 6 GB y un procesador Intel® Core™ i7-3630QM .

5. Desarrollo del proyecto

5.1. Metodología

Este proyecto ha sido desarrollado siguiendo una metodología ágil, la metodología Extreme Programming o Programación Extrema. Se ha optado por esta metodología dada las circunstancias de disponibilidad de tiempo del desarrollador, por esto se ha decidido seguir una metodología que dé prioridad al desarrollo y a la adaptabilidad antes que a la previsibilidad. Así pues, el objetivo siempre se ha ido marcando en unas determinadas funcionalidades que conforme se han ido completando correctamente se han ido ampliando.

Algunas de las características de esta metodología llevada a cabo han sido:

- **Simplicidad:** Diseño simplificado para agilizar el desarrollo. Mejor hacer algo simple que requiera un poco más de esfuerzo para ser modificado, que hacer algo complicado sin estar seguro de si se va a utilizar.
- **Desarrollo interactivo e incremental:** Pequeñas mejoras y una tras otra.
- **Retroalimentación o Feedback:** El propio desarrollador puede tomar el rol de cliente a la hora de realizar las pruebas, puesto que hay que ejercerlas sobre una página web que sería el cliente.
- **Refactorización del código:** Reescritura del código para aumentar legibilidad y mantenibilidad sin cambiar su comportamiento.
- **Corrección de errores** antes de añadir cualquier funcionalidad nueva.

5.2. Requisitos

5.2.1. Funcionales

Código	FRQ-001
Nombre	Creación de la base de datos
Descripción	El usuario podrá crear la base de datos a través del archivo XML y el programa en Python.

Tabla 18: FRQ-001 – Creación de la base de datos

Código	FRQ-002
Nombre	Reinicio de la base de datos
Descripción	El usuario podrá resetear la base de datos a través del archivo XML y el programa en Python.

Tabla 19: FRQ-002 – Reinicio de la base de datos

Código	FRQ-003
Nombre	Seleccionar modo detallado
Descripción	El usuario podrá seleccionar la opción de ver la ejecución del programa en Python con información detallada del proceso.

Tabla 20: FRQ-003 – Seleccionar modo detallado

Código	FRQ-004
Nombre	Indicar el nombre del archivo XML
Descripción	El usuario podrá nombrar como desee al archivo XML creado y luego indicar el nombre del mismo en la ejecución del programa.

Tabla 21: FRQ-004 – Indicar el nombre del archivo XML

Código	FRQ-005
Nombre	Registro en la página web
Descripción	El usuario podrá registrarse en la página web y, con el registro, se enviará un correo electrónico al administrador para informar de dicho registro y con ello proceder a informar al cliente registrado de las opciones disponibles y los pasos a seguir.

Tabla 22: FRQ-005 – Registro en la página web

Código	FRQ-006
Nombre	Registrar usuarios del cliente
Descripción	El usuario con rol de cliente administrador podrá, a través de la página web, registrar nuevos usuarios propios.

Tabla 23: FRQ-006 – Registrar usuarios del cliente

Código	FRQ-007
Nombre	Modificar datos de usuario
Descripción	Cada usuario podrá editar su información personal

Tabla 24: FRQ-007 – Modificar datos de usuario

Código	FRQ-008
Nombre	Valorar comentarios
Descripción	El usuario con rol de cliente administrador podrá valorar, como positivo o negativo, cada comentario obtenido del proceso de almacenamiento del programa en Python.

Tabla 25: FRQ-008 – Valorar comentarios

Código	FRQ-009
Nombre	Modificar valoraciones
Descripción	El usuario con rol cliente administrador podrá modificar valoraciones de comentarios ya valorados anteriormente.

Tabla 26: FRQ-009 – Modificar valoraciones

Código	FRQ-010
Nombre	Descargar comentarios
Descripción	El usuario podrá descargarse todos los comentarios almacenados.

Tabla 27: FRQ-010 – Descargar comentarios

Código	FRQ-011
Nombre	Filtrar comentarios
Descripción	El usuario podrá agregar filtros personalizados al descargar los comentarios.

Tabla 28: FRQ-011 – Filtrar comentarios

Código	FRQ-012
Nombre	Ignorar palabras en comentarios
Descripción	El usuario podrá agregar una lista de palabras prohibidas que serán ignoradas al descargar los comentarios.

Tabla 29: FRQ-012 – Ignorar palabras en comentarios

Código	FRQ-013
Nombre	Crear informes
Descripción	El usuario con rol de cliente administrador podrá crear informes a través de un formulario en la página web.

Tabla 30: FRQ-013 – Crear informes

Código	FRQ-014
Nombre	Editar informes
Descripción	El usuario con rol de cliente administrador podrá editar informes a través de un formulario en la página web.

Tabla 31: FRQ-014 – Editar informes

Código	FRQ-015
Nombre	Desactivar informes
Descripción	El usuario con rol de cliente administrador podrá desactivar informes a través de la página web, quedando oculto para el cliente usuario pero no eliminándose.

Tabla 32: FRQ-015 – Desactivar informes

Código	FRQ-016
Nombre	Eliminar informes
Descripción	El usuario con rol de cliente administrador podrá eliminar informes de forma definitiva a través la página web.

Tabla 33: FRQ-016 – Eliminar informes

Código	FRQ-017
Nombre	Ver gráficos de informes
Descripción	El usuario podrá ver los gráficos de las consultas almacenadas en su base de datos correspondiente.

Tabla 34: FRQ-017 – Ver gráficos de informes

Código	FRQ-018
Nombre	Reportar error en informe
Descripción	El usuario con rol de cliente usuario podrá reportar un error en un informe concreto a través la página web.

Tabla 35: FRQ-018 – Reportar error en informe

Código	FRQ-019
Nombre	Elegir tipo de gráfico en informe
Descripción	El usuario con rol de cliente administrador podrá seleccionar el tipo de gráfica que mostrará su informe entre gráfica de barras, gráfica de líneas o gráfica de pastel.

Tabla 36: FRQ-019 – Elegir tipo de gráfico en informe

5.2.2. De información

Código	IRQ-001
Nombre	Información de los <i>usuarios</i>
Descripción	El sistema deberá almacenar la información correspondiente a los usuarios y su rol.
Datos específicos	Id Nombre Contraseña Correo NombreEmpresa BaseDeDatosAsignada Activo Rol: Admin, ClienteAdmin o ClienteUsuario

Tabla 37: IRQ-001 – Información de los *usuarios*

Código	IRQ-002
Nombre	Información de los <i>productos</i>
Descripción	El sistema deberá almacenar la información correspondiente a los productos.
Datos específicos	IdAutoincremental Otros datos definidos por el cliente

Tabla 38: IRQ-002 – Información de los *productos*

Código	IRQ-003
Nombre	Información de los <i>comentarios</i>
Descripción	El sistema deberá almacenar la información correspondiente a los comentarios.
Datos específicos	IdAutoincremental IdProducto Comentario Otros datos definidos por el cliente

Tabla 39: IRQ-003 – Información de los *comentarios*

Código	IRQ-004
Nombre	Información de los <i>análisis</i>
Descripción	El sistema deberá almacenar la información correspondiente a las valoraciones personales de los usuarios sobre los comentarios.
Datos específicos	IdAutoincremental IdComentario Valoración

Tabla 40: IRQ-004 – Información de los *análisis*

Código	IRQ-005
Nombre	Información de los <i>informes</i>
Descripción	El sistema deberá almacenar la información correspondiente a los informes.
Datos específicos	<p>IdAutoincremental Título Descripción Tipo SelectDeLaConsulta WhereDeLaConsulta GroupByDeLaConsulta OrderByDeLaConsulta Activo</p>

Tabla 41: IRQ-005 – Información de los *informes*

5.2.3. No funcionales

Código	NRQ-001
Nombre	Usabilidad
Descripción	División de las tareas de la aplicación en roles, agrupando los roles según el conocimiento informático requerido para realizar ciertas actividades. Además, el sistema cuenta con una interfaz simple y con mensajes de ayuda en los puntos de más complejidad.

Tabla 42: NRQ-001 – Usabilidad

Código	NRQ-002
Nombre	Escalabilidad
Descripción	Facilidad para ampliar el software agregando nuevas funcionalidades. Se ha utilizado el paradigma Modelo-Vista-Controlador, el cual proporciona una clara separación según el tipo de lógico, haciendo fácil el mantenimiento y la escalabilidad de la aplicación.

Tabla 43: NRQ-002 – Escalabilidad

Código	NRQ-003
Nombre	Funcionalidad
Descripción	Tiene las funciones necesarias para llevar a cabo las propuestas que ofrece y realiza dichas funciones correctamente. El menú de opción detalla de forma concreta y simple cada una de las funcionalidades de la aplicación.

Tabla 44: NRQ-003 – Funcionalidad

Código	NRQ-004
Nombre	Seguridad
Descripción	Los datos de cada cliente están en bases de datos independientes y se realizarán copias de seguridad de las bases de datos. Se utiliza las características del framework CodeIgniter para garantizar la seguridad de la web, además se separa en dos las bases de datos, una que contiene la información de todos los usuarios y otra donde está la información de cada cliente, de esta última cada cliente tendrá su base de datos con su usuario de acceso correspondiente. Esto proporciona que desde las vistas de creación de informe, los cuales se realizan introduciendo consultas SQL, no se pueda acceder de ninguna forma a la base de datos común a todos los clientes, sino que esa consulta solo podrá realizarla sobre la propia base de datos del cliente.

Tabla 45: NRQ-004 – Seguridad

5.2.4. Reglas de negocio

La aplicación presentará aspectos diferentes dependiendo del rol asignado al usuario que la use. Para los clientes existen dos roles, que son:

- **Cliente administrador:** Será el encargado de administrar los usuarios de su organización así como de crear nuevos usuarios. También se encargará de crear las consultas a la base de datos que generarán los informes gráficos que podrán ver y modificar. Pueden obtener archivos Excel con la información almacenada, filtrar dichos datos y valorar comentarios de forma positiva o negativa.
- **Cliente usuario:** Dispone de una interfaz más sencilla donde solo podrá ver los informes generados por el cliente administrador, así como descargar los datos almacenados en un archivo Excel y, si lo desea, aplicar diferentes filtros sobre dichos datos.

Por otro lado, la idea de negocio de la aplicación consta de dos formas de comercialización, que son:

- **Versión completa:** Esta opción sería la más cara y consistiría en que el cliente solo debe indicar al administrador la web objetivo y los datos que desea obtener. Los administradores se encargarían de la realización del archivo de configuración y del mantenimiento de este. Igualmente sería el propio cliente a través del usuario con el rol de "cliente administrador" el encargado de realizar las consultas que luego se reflejen en estadísticas gráficas para el usuario con el rol de "cliente usuario".
- **Versión reducida:** Esta sería una opción más económica en el que se ofrece al cliente el manual de instrucciones del archivo de configuración así como los datos de la base de datos que se le ha asignado. De esta forma será el propio cliente el que deba llenar y mantener el archivo de configuración. Al igual que en la versión anterior se le asignaría de inicio un usuario con el rol de "cliente administrador" para la realización de las consultas así como la creación de usuarios dentro de su organización.

También podemos destacar las siguientes reglas:

- El formato para crear el archivo de configuración XML necesario para la ejecución del programa en Python debe seguir una nomenclatura y formato marcados en el manual de uso de la aplicación disponible en la sección 6.2.
- Los informes se generan a través de consultas SQL que se configuran desde la aplicación web y deben estar bien formuladas siguiendo las reglas indicadas en la vista de creación de informes, que son:
 - Para acceder a la información de las tablas se deben utilizar los alias asignados, que son 'p' para la tabla Productos, 'c' para la tabla Comentarios y 'a' para la tabla Análisis.
 - El select puede ser de dos formas, una forma donde se muestren dos columnas, una llamada 'label' que será el nombre de las columnas del informe final y otra llamada 'data' donde irán los valores numéricos relacionados a cada dato de la columna

'label', o la otra forma es que todos los resultados de la consulta sean numéricos, en este caso los nombres de cada columna de la consulta serán los nombres de las columnas del informe que se mostrará.

5.3. Catálogo de actores

Código	ACT-001
Nombre	Administrador
Descripción	Encargado de dar de alta a los nuevos clientes registrados y de la comunicación con ellos, así como de preparar los archivos XML de los clientes que hayan optado por la versión completa. Puede editar la información de cualquier usuario y es el único con permisos para crear nuevos administradores.

Tabla 46: ACT-001 – Catálogo de actores - Administrador

Código	ACT-002
Nombre	Cliente Administrador
Descripción	Encargado de crear nuevos usuarios para su organización y del control de estos usuarios. Podrá realizar valoraciones personales (positiva o negativa) a los comentarios almacenados, así como descargar la información almacenada aplicándole, si lo desea, diferentes filtros. También es el que debe administrar y crear los informes.

Tabla 47: ACT-002 – Catálogo de actores - Cliente Administrador

Código	ACT-003
Nombre	Cliente Usuario
Descripción	Su menú va directo a la visualización de la información almacenada. Puede descargar la información almacenada y aplicar diferentes filtros sobre ella. También puede visualizar los informes, así como reportar errores en ellos, dichos reportes irán directamente a los administradores de la organización vía correo electrónico.

Tabla 48: ACT-003 – Catálogo de actores - Cliente Usuario

5.4. Análisis del sistema

5.4.1. Casos de uso

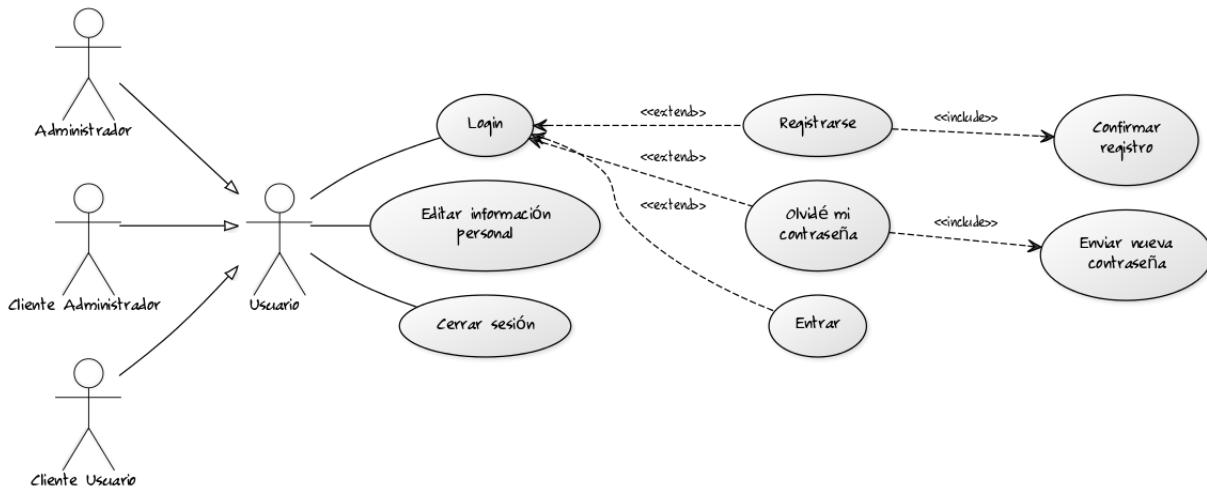


Figura 3: Subsistema de gestión de usuarios.

Descripción de los casos de uso del subsistema de gestión de usuario

- Caso de uso **Login**

- Escenario principal

1. El sistema muestra la pantalla de inicio de sesión.
2. El usuario inserta su nombre de usuario y contraseña en los campos correspondiente.
3. El usuario pulsa el botón "Entrar".
4. El sistema comprueba que los datos son correctos.
5. El sistema inicia la pantalla principal correspondiente al usuario logueado.

- Escenarios alternativos

- 3.1 El usuario pulsa el botón "¡Regístrate aquí!"
 - 3.1.1 Se lleva a cabo el caso de uso **Registrarse**.
 - 3.1.2 Muestra el escenario principal del caso de uso.
- 3.2 El usuario pulsa el botón "Olvidé mi contraseña"
 - 3.2.1 Se lleva a cabo el caso de uso **Olvidé mi contraseña**.
 - 3.2.2 Muestra el escenario principal del caso de uso.
- 4.1 Datos de acceso erróneos
 - 4.1.1 El sistema detecta que los datos son erróneos, advierte al usuario del error y muestra el escenario principal del caso de uso.

- Caso de uso **Registrarse**

- Escenario principal

1. El sistema muestra la pantalla de registro de usuario.
2. El usuario introduce el nombre de usuario, correo electrónico y contraseña.
3. El usuario introduce el nombre de la página web objetivo.
4. El usuario pulsa el botón Completar.
5. El sistema comprueba que los datos del registro son correctos.
6. El sistema envía un correo electrónico a la dirección de correo confirmando el registro.
7. Se muestra la pantalla de registro con un mensaje informando de que el registro ha sido satisfactorio.

- Escenarios alternativos

- 4.1 El usuario pulsa el botón "Volver al Inicio".
 - 4.1.1 El sistema carga el caso de uso **Login**
- 5.1 Datos erróneos.
 - 5.1.1 El nombre de usuario está utilizado.
 - 5.1.2 El sistema muestra el error en la pantalla inicial del caso de uso.

- Caso de uso **Olvidé mi contraseña**

- Escenario principal

1. El sistema muestra la pantalla del formulario de recuperación de la contraseña.
2. El usuario introduce su nombre de usuario y su correo electrónico.
3. El usuario pulsa el botón Completar.
4. El sistema comprueba que los datos introducidos son correctos.
5. El sistema envía un correo electrónico con una nueva contraseña.
6. Se muestra la pantalla inicial con un mensaje indicando que el cambio de contraseña ha sido satisfactorio.

- Escenarios alternativos

- 3.1 El usuario pulsa el botón "Volver al Inicio".
 - 3.1.1 El sistema carga el caso de uso **Login**
- 4.1 Datos erróneos
 - 4.1.1 El sistema comprueba que el correo electrónico y/o el nombre de usuario introducido no coincide con lo almacenado en la base de datos.
 - 4.1.2 Se muestra la pantalla inicial indicando que los datos introducidos son erróneos.

- Caso de uso **Editar información personal**

- Escenario principal

1. Se lleva a cabo el caso de uso **Login**
2. El usuario pulsa sobre el botón "Configuración" situado en el desplegable que aparece al pulsar sobre el nombre de usuario situado en la esquina superior derecha de la pantalla.
3. El sistema muestra la pantalla de "Configuración" con la información del usuario y los datos que puede cambiar.

4. El usuario introduce nueva información.
5. El sistema comprueba que los datos son correctos y modifica la información.
6. El sistema muestra la pantalla de "Configuración" con un mensaje indicando que la modificación ha sido satisfactorio.

- **Escenarios alternativos**

- 5.1 Datos erróneos

- 5.1.1 El sistema comprueba que la nueva contraseña ha modificar, que hay que introducir dos veces, no coinciden entre ellas.
 - 5.1.2 Se muestra la pantalla de "Configuración" con un mensaje indicando el error.

- Caso de uso **Cerrar sesión**

- **Escenario principal**

1. Se lleva a cabo el caso de uso **Login**.
2. El usuario pulsa sobre el botón "Cerrar sesión" situado en el desplegable que aparece al pulsar sobre el nombre de usuario situado en la esquina superior derecha de la pantalla.
3. El sistema muestra la pantalla de "Login" con un mensaje de despedida.

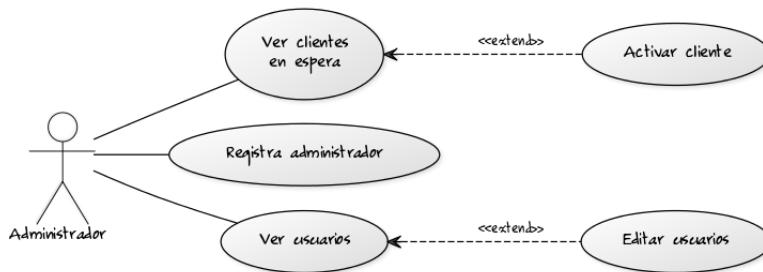


Figura 4: Subsistema del rol de Administrador.

Descripción de los casos de uso del subsistema del rol de Administrador

■ Caso de uso **Ver clientes en espera**

- Escenario principal

1. Se lleva a cabo el caso de uso **Login**.
2. Se muestra directamente la pantalla de clientes a la espera de confirmación.
3. El sistema carga y muestra los clientes registrados que están inactivos y su información correspondiente.

- Escenarios alternativos

- 2.1 Se pulsa el botón de la casa situado en la esquina superior izquierda, en la barra lateral.
- 2.2 Se pulsa sobre el botón "Cuadro de mando" situado en la barra lateral.

■ Caso de uso **Activar cliente**

- Escenario principal

1. Se lleva a cabo el caso de uso **Ver clientes en espera**.
2. Se pulsa sobre el botón "Activar" del cliente inactivo deseado.
3. Se muestra la información del cliente seleccionado y un botón de activación.
4. Se introduce el nombre de la base de datos que se le ha creado a dicho cliente y se pulsa sobre "Activar".
5. El sistema activa la base de datos y se muestra la pantalla de activación con un mensaje de que la activación se ha llevado a cabo satisfactoriamente.

- Escenarios alternativos

- 5.1 El sistema comprueba que el nombre de la base de datos ya existe y muestra la pantalla de activación con un mensaje de error.

■ Caso de uso **Registrar administrador**

- Escenario principal

1. Se lleva a cabo el caso de uso **Login**.
2. El usuario pulsa sobre el botón "Registrar administrador" de la barra lateral.
3. El sistema carga y muestra el formulario de registro.
4. El usuario introduce los datos y pulsa el botón "Completar".

5. El sistema crea un nuevo usuario administrador y muestra un mensaje de que se ha realizado el registro satisfactoriamente.

- **Escenarios alternativos**

- 5.1 El sistema comprueba que el nombre de usuario ya existe y muestra un mensaje de error por pantalla.

- Caso de uso **Ver usuarios**

- **Escenario principal**

1. Se lleva a cabo el caso de uso **Login**.
2. El usuario pulsa sobre el botón "Ver usuarios" de la barra lateral.
3. El sistema carga y muestra la información de los usuarios registrados y activos.

- Caso de uso **Editar usuarios**

- Escenario principal

1. Se lleva a cabo el caso de uso **Ver usuarios**.
2. El usuario pulsa sobre el botón "Editar" del usuario a editar.
3. El sistema carga y muestra la pantalla de edición de usuario.
4. El usuario edita el dato deseado.
5. El sistema modifica los datos del usuario y se muestra un mensaje informando de la edición correcta del dato modificado.

- Escenario alternativo

- 5.1 El sistema falla al modificar los datos del usuario y se muestra un mensaje informando del error.

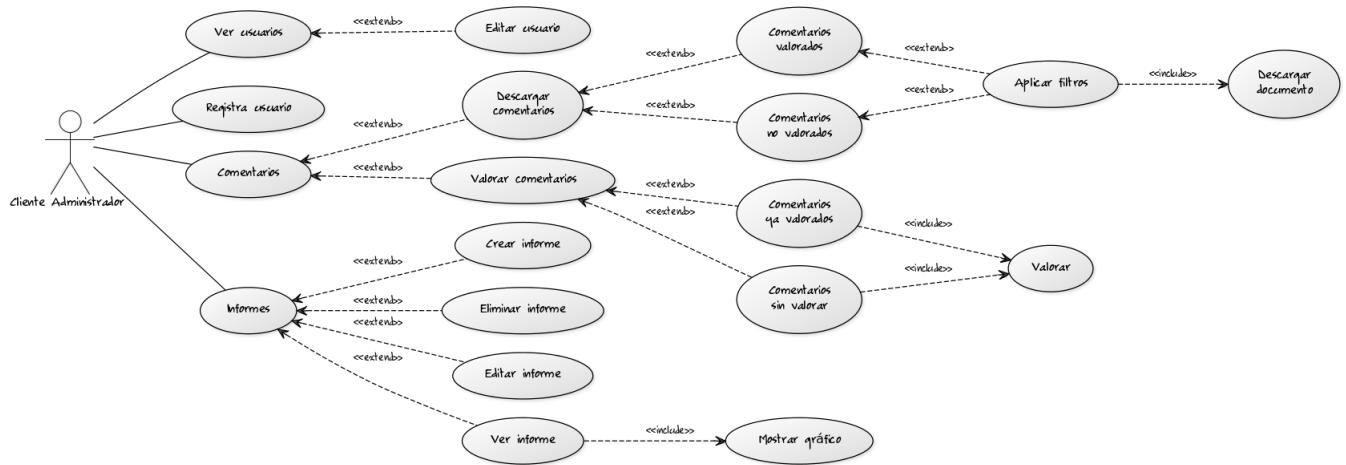


Figura 5: Subsistema del rol de Cliente Administrador.

Descripción de los casos de uso del subsistema del rol de Cliente Administrador

■ Caso de uso **Ver usuarios**

- Escenario principal

1. Se lleva a cabo el caso de uso **Login**.
2. Se muestra directamente la pantalla de usuarios activos.
3. El sistema carga y muestra la información correspondiente a los clientes registrados que están activos y están asociado al cliente logueado.

- Escenarios alternativos

- 2.1 Se pulsa el botón de la casa situado en la esquina superior izquierda, en la barra lateral.
- 2.2 Se pulsa sobre el botón "Cuadro de mando" situado en la barra lateral.

■ Caso de uso **Editar usuario**

- Escenario principal

1. Se lleva a cabo el caso de uso **Ver usuarios**.
2. El usuario pulsa sobre el botón "Editar" del usuario a editar.
3. El sistema carga y muestra la pantalla de edición de usuario.
4. El usuario edita el dato deseado.
5. El sistema modifica los datos del usuario y se muestra un mensaje informando de la edición correcta del dato modificado.

- Escenario alternativo

- 5.1 El sistema falla al modificar los datos del usuario y se muestra un mensaje informando del error.

■ Caso de uso **Registrar usuario**

- Escenario principal

1. Se lleva a cabo el caso de uso **Login**.
2. El usuario pulsa sobre el botón "Registrar usuario" de la barra lateral.
3. El sistema carga y muestra el formulario de registro.
4. El usuario introduce los datos y pulsa el botón "Completar".
5. El sistema crea un nuevo usuario de rol Cliente Usuario relacionado con el usuario logueado y muestra un mensaje de que se ha realizado el registro satisfactoriamente.

- **Escenarios alternativos**

- 5.1 El sistema comprueba que el nombre de usuario ya existe y muestra un mensaje de error por pantalla.

- Caso de uso **Comentarios**

- **Escenario principal**

1. Se lleva a cabo el caso de uso **Login**.
2. El usuario pulsa sobre el botón "Comentarios" de la barra lateral.
3. El sistema muestra las opciones de valoración de comentarios así como las opciones de descarga de comentarios.

- Caso de uso **Valorar comentarios**

- **Escenario principal**

1. Se lleva a cabo el caso de uso **Comentarios**.
2. El usuario selecciona "Comentarios no valorados".
3. El usuario pulsa sobre el botón "Valorar".
4. El sistema carga y muestra un máximo de 10 de los comentarios correspondientes.
5. El usuario valora los comentarios y envía el formulario.
6. El sistema comprueba que el formulario está lleno, almacena la información y, si hay más comentarios, los carga.

- **Escenarios alternativos**

- 2.1 El usuario selecciona "Comentarios ya valorados".
- 5.1 El sistema comprueba que se ha quedado algún comentario sin valorar y devuelve un error por pantalla informando de que faltan comentarios por valorar.

- Caso de uso **Descargar comentarios**

- **Escenario principal**

1. Se lleva a cabo el caso de uso **Comentarios**.
2. El usuario selecciona "Comentarios no valorados".
3. El usuario introduce las palabras prohibidas.
4. El usuario introduce las restricciones asociadas a los productos.
5. El usuario introduce las restricciones asociadas a los comentarios.
6. El usuario pulsa sobre el botón "Descargar".
7. El sistema crea un fichero Excel de los comentarios obtenidos teniendo en cuenta todas las indicaciones introducidas por el usuario.

- **Escenarios alternativos**

- 2.1 El usuario selecciona "Comentarios valorados".
- 3.1 El usuario no introduce ninguna palabra prohibida.
- 4.1 El usuario no introduce ninguna restricción asociada a los productos.
- 5.1 El usuario no introduce ninguna restricción asociada a los comentarios.

- Caso de uso **Informes**

- Escenario principal

1. Se lleva a cabo el caso de uso **Login**.
2. El usuario pulsa sobre el botón "Informes" de la barra lateral.
3. El sistema muestra los informes disponibles asociados al usuario logueado, sus opciones y el botón de creación de un nuevo informe.

- Caso de uso **Crear informe**

- Escenario principal

1. Se lleva a cabo el caso de uso **Informes**.
2. El usuario pulsa sobre el botón "Crear nuevo informe".
3. El sistema carga el formulario de creación de informe.
4. El usuario introduce los datos del formulario.
5. El sistema valida el formulario y crea un nuevo informe, carga la pantalla de informes e indica que se ha creado correctamente a través de un mensaje.

- Escenarios alternativos

- 5.1 El usuario deja en blanco el título, descripción, tipo o consulta del informe introducido, el sistema cargará la pantalla de informes indicando el error a través de un mensaje.

- Caso de uso **Eliminar informe**

- Escenario principal

1. Se lleva a cabo el caso de uso **Informes**.
2. El usuario pulsa sobre el botón "Eliminar" del informe que deseé borrar.
3. El sistema muestra un mensaje advirtiendo que el informe seleccionado se eliminará.
4. El usuario pulsa sobre el botón "Aceptar" y el sistema elimina el informe y muestra un mensaje indicando que se ha eliminado satisfactoriamente.

- Escenarios alternativos

- 4.1 El usuario pulsa sobre el botón "Cancelar", se cierra el mensaje y el sistema no hace nada.

- Caso de uso **Editar informe**

- Escenario principal

1. Se lleva a cabo el caso de uso **Informes**.
2. El usuario pulsa sobre el botón "Editar" del informe que desea modificar.
3. El usuario carga un formulario igual que el de creación de informes pero con los datos llenos del informe seleccionado.
4. El usuario modifica los datos deseados.

5. El sistema valida el formulario y edita el informe, carga la pantalla de informes e indica que se ha modificado correctamente a través de un mensaje.

- **Escenarios alternativos**

- 5.1 El usuario deja en blanco el título, descripción, tipo o consulta del informe introducido, el sistema cargará la pantalla de informes indicando el error a través de un mensaje.

- Caso de uso **Ver informe**

- **Escenario principal**

1. Se lleva a cabo el caso de uso **Informes**.
2. El usuario pulsa sobre el botón "Ver" del informe deseado.
3. El sistema carga una nueva pestaña donde se muestra el gráfico relacionado al informe seleccionado.

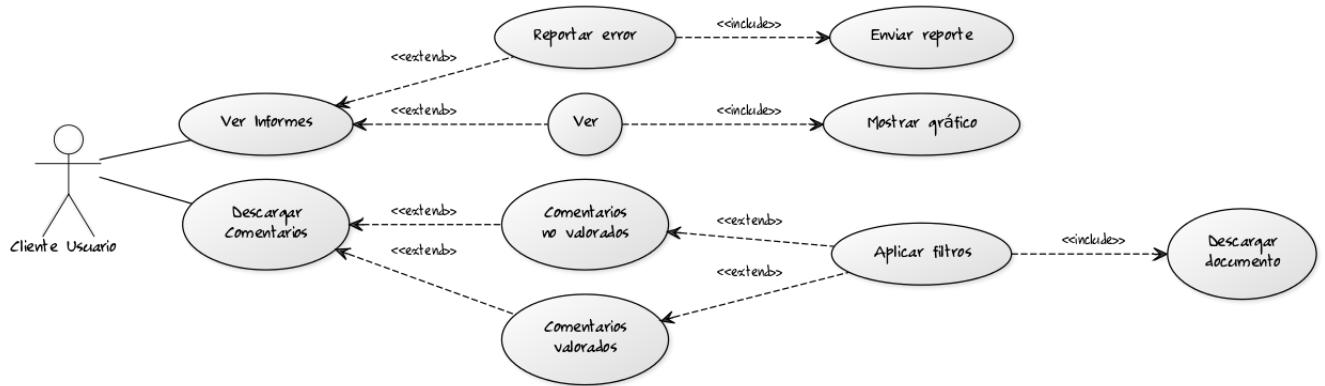


Figura 6: Subsistema del rol de Cliente Usuario.

Descripción de los casos de uso del subsistema del rol de Cliente Usuario

■ Caso de uso **Ver informes**

- Escenario principal
 1. Se lleva a cabo el caso de uso **Login**
 2. Se muestra directamente la pantalla de los informes activos.
 3. El sistema carga la lista de informes activos relacionados con el usuario logueado.
- Escenarios alternativos
 - 2.1 El usuario pulsa sobre el botón de la casa situado en la esquina superior izquierda, en la barra lateral.
 - 2.2 El usuario pulsa sobre el botón "Ver informes" situado en la barra lateral.

■ Caso de uso **Ver**

- Escenario principal
 1. Se lleva a cabo el caso de uso **Ver informes**.
 2. El usuario pulsa sobre el botón "Ver" del informe deseado.
 3. El sistema carga una nueva pestaña donde se muestra el gráfico relacionado al informe seleccionado.

■ Caso de uso **Reportar error**

- Escenario principal
 1. Se lleva a cabo el caso de uso **Ver informes**.
 2. El usuario pulsa sobre el botón "Reportar error" del informe deseado.
 3. El sistema muestra la pantalla de reportar errores.
 4. El usuario describe el problema en la casilla correspondiente y pulsa sobre el botón "Enviar".
 5. El sistema envía un correo electrónico a los administradores relacionados con el cliente indicándole el informe reportado, la descripción del problema y el correo electrónico del usuario que lo ha reportado.

6. El sistema muestra la pantalla de los informes y un mensaje indicando que se ha enviado correctamente.

- Caso de uso **Descargar comentarios**

- **Escenario principal**

1. Se lleva a cabo el caso de uso **Login**.
2. El usuario pulsa el botón "Descargar comentarios" situado en la barra lateral.
3. El sistema muestra la vista y el formulario de la descarga de comentarios.
4. El usuario selecciona "Comentarios no valorados".
5. El usuario introduce las palabras prohibidas.
6. El usuario introduce las restricciones asociadas a los productos.
7. El usuario introduce las restricciones asociadas a los comentarios.
8. El usuario pulsa sobre el botón "Descargar".
9. El sistema crea un fichero Excel de los comentarios obtenidos teniendo en cuenta todas las indicaciones introducidas por el usuario.

- **Escenarios alternativos**

- 4.1 El usuario selecciona "Comentarios valorados".
- 5.1 El usuario no introduce ninguna palabra prohibida.
- 6.1 El usuario no introduce ninguna restricción asociada a los productos.
- 7.1 El usuario no introduce ninguna restricción asociada a los comentarios.

5.4.2. Modelo conceptual de datos del dominio

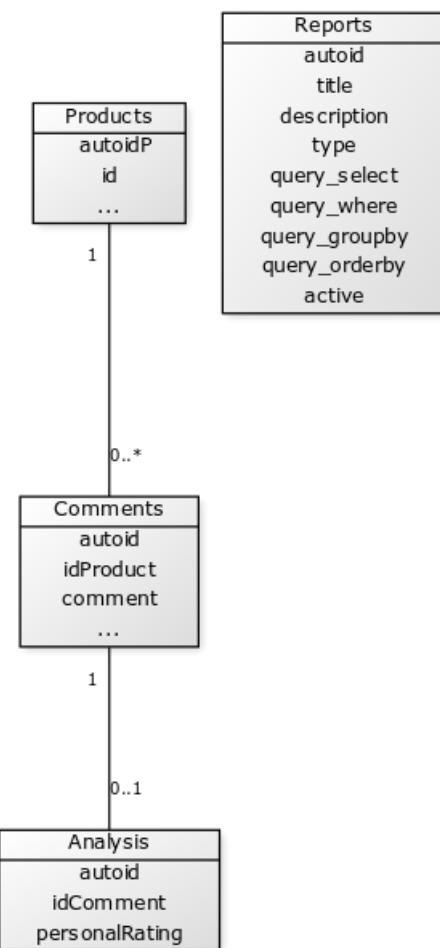


Figura 7: Modelo conceptual de datos de la aplicación.

La base de datos de la aplicación que se muestra en la figura 7 cuenta con cuatro tablas:

- 'Products': Se almacena la información que el cliente desea de los productos de la página web objetivo. Como datos fijos se tiene un id autoincremental y otro id que será el elemento identificativo del producto para acceder a sus comentarios. Este último ID no tiene por qué ser numérico, en ocasiones, por ejemplo, puede ser un extracto de un enlace, y será lo que relacione en la base de datos a cada producto con sus comentarios.
- 'Comments': Se almacena la información que el cliente desea de los comentarios de la página web objetivo. Como datos fijos se tiene un id autoincremental, un id del producto que relacionará cada comentario con un único producto y el atributo donde se almacenará el comentario en sí. El resto de atributos son opcionales y depende de lo que el cliente desee almacenar de la web objetivo.
- 'Analysis': Se almacena la información de las opiniones, positiva o negativa, que el cliente tenga de cada uno de los comentarios. En este caso se tiene únicamente tres atributos fijos, que son un id autoincremental, el id del comentario que relaciona cada una de las puntuaciones con un único comentario, y la opinión introducida por el cliente a través de la aplicación web.

- 'Reports': Se almacena la información de los informes generados por el cliente. Por defecto, cuando se crea las tablas a través de la aplicación de Python, se incluyen también tres informes tras generar la estructura de las tablas. Todos sus atributos son fijos, y está compuesto por un id autoincremental, el título y descripción del informe, el tipo de gráfica en la que se mostrará el informe, en principio puede ser gráfica de barras, de pastel o de líneas, y toda la información de la consulta SQL que generará el informe, separada dicha consulta en el 'select', el 'where', el 'group by' y el 'order by', siendo solo el 'select' obligatorio de llenar, y el resto opcionales. También, por último, tenemos un atributo 'active' que señala si el informe está activo, y por lo tanto lo pueden ver los usuarios con el rol 'Cliente usuario' o está desactivado.

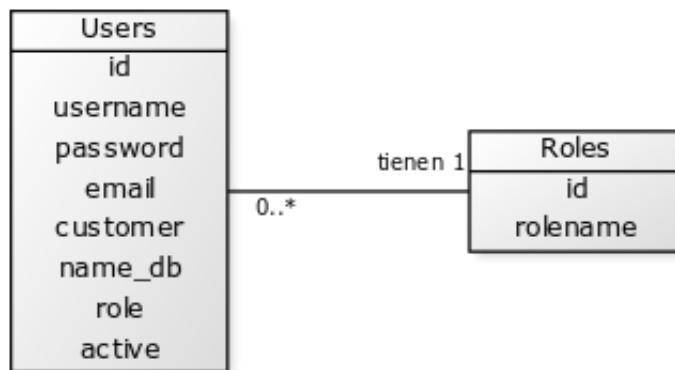


Figura 8: Modelo conceptual de datos de los usuarios.

Por otro lado tenemos la base de datos que se muestra en la figura 8 y que es donde se almacenan los usuarios de la aplicación web.

Esta base de datos está compuesta por dos tablas:

- 'Users': En esta tabla se almacena la información de los usuarios registrados. Concretamente se almacena un id autoincremental, el nombre de usuario, la contraseña codificada, el correo electrónico, la información de la web objetivo que haya introducido el usuario al registrarse, el nombre de la base de datos asignada, el id del rol asignado y si está o no está activo el usuario. Los usuarios al registrarse a través del formulario de registro de la aplicación web se crean desactivados y sin base de datos asignada. Es el administrador global de la aplicación el que, cuando esté preparada la base de datos, activa el usuario y le asigna la base de datos correspondiente a través de la aplicación web.
- 'Roles': En esta tabla se almacenan los roles disponibles en la aplicación web. Cuenta con dos atributos, el id del rol, el cual se relaciona con la tabla 'Users' a través del atributo 'role' de dicha tabla, y el nombre del rol. En principio la aplicación web cuenta con tres roles diferentes y, dependiendo del rol asignado al usuario, tendrá disponible unas funciones u otras en la aplicación web al iniciar sesión.

5.5. Diseño del sistema

En este apartado se especificará el diseño de la aplicación.

5.5.1. Diseño de la arquitectura

El sistema cuenta con los siguientes elementos principales:

- Python como intérprete y lenguaje de programación para la aplicación que realiza el web scraping.
- MySQL como gestor de las bases de datos.
- PHP y Javascript como lenguajes de programación para la aplicación web.

Podemos dividir la arquitectura en dos capas, una física que represente las relaciones de la aplicación con respecto al hardware y otra lógica que represente las relaciones de la aplicación a nivel de software:

1. **Arquitectura física:** La aplicación no está realizada para ningún hardware ni sistema operativo específico. La aplicación se compone de una aplicación que realiza el web scraping interpretada por Python y una aplicación web. La compatibilidad de la aplicación web con los principales navegadores ha sido probada con la herramienta gratuita que proporciona <http://browsershots.org/>.
2. **Arquitectura lógica:** Para el desarrollo de la aplicación se ha utilizado el siguiente software:
 - **Sublime Text:** Editor de texto utilizado para el desarrollo del código de la herramienta de web scraping y de la aplicación web.
 - **ShareLaTeX:** Editor de texto en L^AT_EX. Se ha utilizado para el desarrollo de la documentación.
 - **Intérprete de Python:** Intérprete para ejecutar el código desarrollado en dicho lenguaje de programación.
 - **Navegadores web:** Principalmente se ha utilizado Firefox 45.6.0 de Linux para visualizar la aplicación web en su vista para ordenador y Chrome 62.0.3202.84 para visualizarla en dispositivos móviles.

5.6. Diseño de la interfaz de usuario

5.6.1. Pantalla del programa Python de web scraping

El programa Python no tiene una interfaz gráfica, sino que es un código Python que se ejecuta desde la terminal. Lleva una guía a la que se puede acceder ejecutando el programa sin ninguna opción o usando la opción '-h'. En la guía se indican todas las opciones disponibles para la ejecución del código así como la forma de usarlo.

5.6.2. Pantalla de Login, registro y recuperación de la contraseña

Al entrar en la aplicación web la pantalla inicial es un Login básico, donde se puede introducir el nombre de usuario y la contraseña. Además de esto, desde esta pantalla se accede al registro y a la recuperación de la contraseña.

En la pantalla del registro se tiene un formulario en el que se pide insertar el nombre de usuario, un correo electrónico, una contraseña y el nombre o información de la página web objetivo. El registro genera un email automatizado con el que se confirma que todo ha funcionado correctamente.

En la pantalla de recuperación de la contraseña se muestra un formulario en el que se debe introducir el nombre de usuario y el correo electrónico asociado. Con esto, se enviará una nueva contraseña aleatoria al correo electrónico en caso de que los datos introducidos sean correctos.

El diseño de toda la web debe ser 'responsive', es decir, adaptarse a dispositivos de diferentes tamaños.

5.6.3. Pantalla general de la aplicación web y pantalla de configuración

Cada usuario, dependiendo de su rol, tiene unas opciones u otras en su pantalla. Pero todos los usuarios cuentan con un mismo diseño que consiste en una barra lateral donde se muestran las opciones a las que puede acceder a través de unos iconos, una opción fija representada con el ícono de una casa que dirige a la pantalla de inicio, y una barra superior en la que se cuenta con un botón para desplegar la barra lateral y poder ver fijo la descripción de cada ícono y otro botón con el que se podrá acceder a la pantalla de configuración y en el que se muestra el nombre de usuario y rol.

La pantalla de configuración es genérica a todos los usuarios. En ella se muestra toda la información relacionada con el usuario logueado y también se da la posibilidad de modificar la contraseña actual.

5.6.4. Pantalla para el rol 'Administrador'

Cuando se inicie sesión con el rol de administrador se muestra de inicio una lista con la información de los clientes que se han registrado pero están inactivas, es decir, clientes que están a la espera de confirmar su suscripción y de que se la asigne una base de datos.

En la barra lateral se cuenta con las siguientes opciones:

- Registrar administrador: Un formulario donde se debe introducir un nombre de usuario, correo electrónico y contraseña para crear un nuevo usuario activo que tendrá el rol de administrador global de la aplicación.
- Ver usuarios: Una lista en la que se muestra toda la información de los usuarios activos en la aplicación web. En la lista hay un botón de edición que nos lleva a otra pantalla en la que se muestra la información del usuario seleccionado y un formulario donde se puede editar algunos de sus datos, como la base de datos asignada, el rol o la contraseña. También se puede desactivar al usuario desde esta pantalla.

5.6.5. Pantalla para el rol 'Cliente Administrador'

Cuando se entra con un usuario con el rol de 'Cliente Administrador' se muestra de inicio una lista de los usuarios activos relacionados con el cliente logueado. Desde esta vista, a través de un botón de edición, se puede acceder a la pantalla de edición de usuario en el que se ve toda la información del usuario seleccionado y se puede editar el rol y cambiar la contraseña. También se puede eliminar el usuario.

En la barra lateral se cuenta con las siguientes opciones:

- Registrar usuario: Esa opción nos lleva a un formulario donde introduciendo nombre de usuario, correo electrónico y contraseña se creará un nuevo usuario asociado al cliente y con el rol predeterminado de 'Cliente Usuario'.
- Comentarios: El diseño de esta vista lo podemos separar en dos partes. En la parte superior se ve la valoración de comentarios, donde hay un formulario seleccionable que da la posibilidad de elegir entre 'Comentarios no valorados' y 'Comentarios ya valorados', y un botón 'Valorar'. A través de este formulario se accede a otra vista donde se muestra una lista de comentarios, dependiendo de la opción seleccionada serán comentarios valorados o no valorados, y podremos asignar una valoración positiva o negativa a cada comentario listado. Al final de la lista hay un botón con el que se almacena los datos introducidos y nos pasa a la siguiente página de comentarios. Por otro lado, en la parte inferior de la pantalla de los comentarios, tenemos la opción de 'Descargar comentarios', la cual también será un formulario con una opción seleccionable entre 'Comentarios no valorados' y 'Comentarios valorados', una casilla para introducir una lista de palabras prohibidas separadas por comas, y la opción de filtrar el resultado por algún atributo de la tabla de los productos o de la tabla de los comentarios de la base de datos. Al final de este formulario hay un botón con el que, al pulsarlo, se descargará un Excel cuyos datos dependerán de las opciones indicadas.
- Informes: En la vista de los informes tenemos un listado con la información de cada informe. Para cada informe tenemos tres opciones, editar, ver o eliminar. La opción editar nos lleva a una vista igual a la de creación de informes pero con los datos del formulario llenos con la información del informe seleccionado. La opción Ver nos abre una nueva ventana que carga el informe y muestra el gráfico, los datos y la leyenda. Por último, la opción eliminar nos abre una ventana emergente que preguntará si se desea eliminar y dará la opción a borrar el informe marcado. Al final de la lista hay un botón

que indica 'Crear nuevo informe'. Pulsando sobre este botón vamos a una nueva vista, un formulario donde se presentan los datos necesarios para la creación de informes, así como una explicación detallada de como se debe completar algunos de los datos del formulario.

5.6.6. Pantalla para el rol 'Cliente Usuario'

Al entrar con un usuario con rol 'Cliente Usuario' nos encontramos con dos opciones en la barra lateral, que son 'Ver informes' y 'Descargar comentarios'.

Al iniciar sesión con un usuario 'Cliente Usuario' lo primero que se nos carga es la opción 'Ver informes', la cual muestra una lista con la información de cada informe y los botones 'Ver' y 'Reportar error'. A través del botón Ver se abre una nueva ventana que carga el informe y muestra el gráfico, los datos y la leyenda. A través del botón Reportar error se abrirá una nueva ventana en la que se introduce una descripción del error y, pulsando en el botón de Enviar, se envía un correo electrónico al administrador del cliente con la información del informe reportado y la descripción del error introducida.

En la barra lateral también tenemos 'Descargar comentarios'. En esta opción se muestra un formulario donde se podrá seleccionar entre 'Comentarios no valorados' o 'Comentarios valorados', además de introducir varias opciones como palabras prohibidas o filtro por datos de las tablas de productos y/o comentarios de la base de datos del cliente. Con todas estas opciones, al pulsar sobre el botón 'Descargar' se generará una hoja de cálculo para descargar con la información de la base de datos.

5.7. Diseño de componentes

Podemos dividir la composición de la aplicación en los siguientes módulos:

- **Desarrollo del código en Python:** Para ello se ha utilizado Sublime Text para programar en Python, y MySQL como base de datos donde almacenar la información obtenida a través de la ejecución del código.
- **Desarrollo de la aplicación web:** Se ha utilizado Sublime Text para programar la página web con los lenguajes de programación PHP, Javascript y HTML, y MySQL para las bases de datos, tanto para la base de datos que almacena la información obtenida a través del programa en Python y que la aplicación web manipula, como para la información específica de la aplicación web relacionada con la autenticación y los roles.
- **Desarrollo de documentación y presentación:** Se ha utilizado el sistema de composición de textos L^AT_EXy la web ShareL^AT_EXpara la redacción de la documentación y la composición de la presentación.
- **Pruebas de la aplicación:** Se ha utilizado un ordenador portatil ASUS con el sistema operativo Debian Jessie donde se han realizado pruebas usando el navegador Firefox 45.6.0 y el interprete de Python, a través del framework Unittest de Python 3. Para la usabilidad y la visualización en dispositivos móviles se ha usado un smartphone ZTE Blade V6 con Android 5.0.2. Además, para probar la compatibilidad de la aplicación web se ha utilizado webs como <http://browsershots.org/>.

5.8. Implementación

5.8.1. Entorno tecnológico

El entorno que se ha utilizado para la construcción del sistema relacionado con la aplicación que realiza el web scraping ha sido Sublime Text.

Sublime Text es un editor de texto simple y elegante, además, de un editor muy cómodo cuando no requieres de capacidades extras a la hora de programar el código que deseas. Además de Python 3 se ha utilizado la biblioteca Beautiful Soup tanto para la lectura del archivo de configuración XML como para la realización del web scraping sobre la web objetivo. Se ha decidido seleccionar Python y Beautiful Soup porque esta es la más famosa herramienta para realizar web scraping desde Python y cuenta con una documentación muy clara y estructurada que facilita su aprendizaje y uso. Para las pruebas del código Python se ha utilizado el framework unittest, que ofrece la posibilidad de realizar código dentro de una clase del framework que sirva para probar lo ya programado, comparando los resultados salientes del código de la aplicación con resultados esperados. Para hacer funcionar la aplicación se requiere que el entorno de ejecución tenga Python 3 instalado, así como Python 3 MySQLdb y los modulos de Python Beautiful Soup y Requests. También se necesita conexión con el servidor MySQL donde se realizará la inserción de los datos obtenidos.

Para la aplicación web se ha utilizado un servidor Apache como entorno donde se ha programado en lenguaje PHP a través del framework CodeIgniter que sigue una estructura modelo-vista-controlador la cual se ha seguido para el desarrollo. También se ha utilizado Javascript

como lenguaje de programación. Como complementos para la página web se ha utilizado Bootstrap y Font Awesome para el diseño, Chart.js para la visualización de los gráficos, WOW.js para la visualización del scroll y jQuery para la programación de Javascript. El entorno necesario para arrancar la aplicación web solo requiere de un servidor Apache y la conexión con un servidor MySQL que aloje las bases de datos necesarias, es decir, la base de datos relacionada con los usuarios de la web y las bases de datos que almacene los datos de la web objetivo de cada cliente.

5.8.2. Código fuente

El código fuente de ambas aplicaciones está disponible en el repositorio web GitHub, el de la aplicación que realiza el web scraping () y el de la aplicación web ().

El funcionamiento base y la composición del código fuente de la aplicación que realiza el web scraping es el siguiente:

- **Pruebas al archivo XML:** Una primera función comprueba que los atributos obligatorios y necesarios para el funcionamiento del script están incluidos en el fichero de configuración XML. En caso de que no esté se parará la ejecución y mostrará un mensaje indicando el error.
- **Enlace de los productos:** Del archivo de configuración se obtiene el enlace donde estarán los productos de la web objetivo y, si es necesario, el número máximo de páginas de productos que se recorrerán. Se va recorriendo dicha web y entrando en cada uno de estos productos.
- **Almacenamiento de información de productos:** Dentro de la información de cada producto se almacena en la base de datos la información deseada de estos y se obtiene el ID del producto necesario para acceder a los comentarios relacionados.
- **Almacenamiento de comentarios:** A través del enlace indicado en el archivo de configuración y el ID del producto obtenido anteriormente se accede a los comentarios. Se almacenará en la base de datos todos los comentarios de cada producto, así como la información extra relacionada con los comentarios que haya sido indicada en el archivo de configuración.

Además de este funcionamiento esencial, esta aplicación también tiene las funcionalidades de crear y resetear la base de datos a partir de los datos de acceso y los atributos de las tablas que deben estar indicados en el archivo de configuración.

Respecto al código fuente de la aplicación web, al estar programado usando CodeIgniter mantiene una estructura de modelo-vista-controlador. Por ello, podemos dividir su estructura básica en:

- **Modelo:** En los modelos se encuentran las funciones encargadas de la manipulación y uso directo de la base de datos. Las funciones están divididas por roles de usuario, habiendo una genérica para todos los roles que es donde estarán, por ejemplo, las funciones relacionadas con el login.

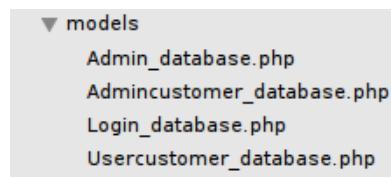


Figura 9: Pantalla de reportar informe

- Controlador: En los controladores están las funciones que se encargan de gestionar la información entre la vista y el modelo, transformando los datos a las necesidades de cada uno. Al igual que los modelos, en la aplicación web están divididas según al rol de usuario que afecte, habiendo un controlador para las funciones genéricas como el sistema de inicio de sesión o el cambio de contraseña.

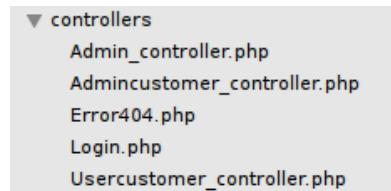


Figura 10: Pantalla de reportar informe

- Vista: En la vista está la información que ve el cliente así como los mecanismo para que el cliente interactúe con la aplicación web. Este modelo da la posibilidad de reutilizar la misma vista para diferentes pantallas, de esta forma, partes de la web que se repiten en muchas de las vistas, como la cabecera, el pie de página o la barra lateral, estarán presentes sin necesidad de reescribir el código cada vez que sea necesario mostrarla.

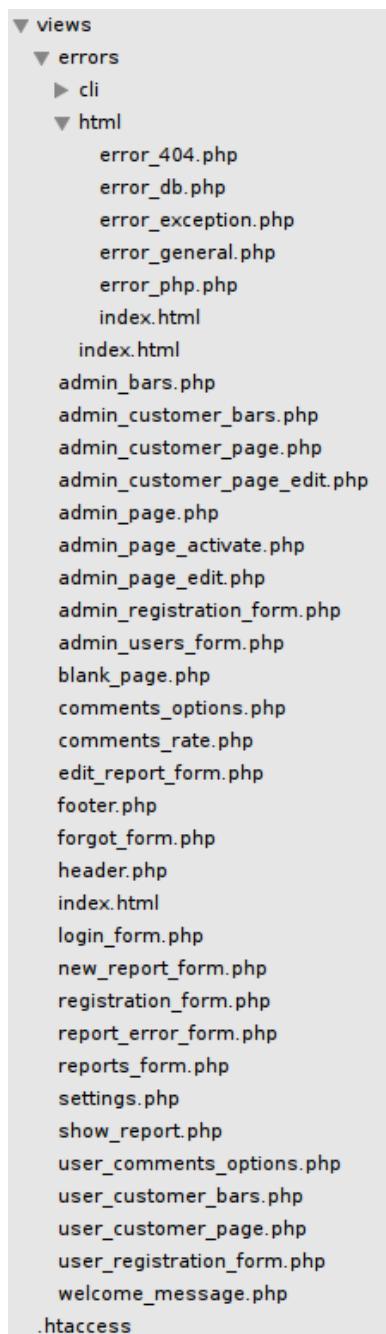


Figura 11: Pantalla de reportar informe

Además de estas carpetas, también cabe destacar que en la carpeta "third_party" está la librería PHPExcel que es la utilizada para la generación del documento Excel que los clientes tienen la posibilidad de descargar. Y en la carpeta "assets" están presentes los archivos CSS, JavaScript, fuentes e imágenes utilizados en la aplicación web.

5.8.3. Calidad del Código

Para la aplicación web se ha utilizado las herramientas que CodeIgniter proporciona para garantizar la calidad del código, como el uso de librerías y vistas para conseguir una abstracción que garantice no duplicar código, la estandarización del código y la separación de arquitectura y lógica al utilizar el patrón modelo-vista-controlador. Todas las funciones y variables de la aplicación web tienen nombres descriptivos así como comentarios explicativos de su funcionalidad con el fin de facilitar la lectura y comprensión.

Por parte de la aplicación en Python, para asegurar el correcto funcionamiento del código se ha utilizado el framework unittest para comprobar que las funciones tenían el comportamiento deseado y que, ante unos datos de entradas predefinidos, devolvía los datos de salida deseados. Mientras que para garantizar la calidad del código se ha utilizado la herramienta Pylint que realiza un análisis del código generando estadísticas y señalando errores con respecto a la calidad del código comprobando si está correctamente definidas las variables, definidas las funciones, importadas las bibliotecas, si el diseño es correcto, si hay conflicto de estilos, si está debidamente formateado, si hay duplicación de código... Las estadísticas generadas por Pylint dividen los datos generados en las siguientes categorías:

- Convención: Relacionados con el estándar de codificación.
- Refactorización: Relacionados con el uso de "buenas prácticas".
- Advertencia: Problemas menores relacionados con la programación o el estilo.
- Error: Problemas de programación importante que afectan al correcto funcionamiento de la aplicación.
- Fatal: Errores que impiden que Pylint finalice su análisis.

Utilizando esta herramienta se ha podido mejorar la calidad del código corrigiendo, aportando y/o asegurando los siguientes patrones:

- Eliminación de espacios innecesarios para mejorar la legibilidad del código.
- Añadido 'docstrings' en todos los módulos y funciones. Los 'docstrings' son cadenas de documentación que funcionan para explicar cada módulo o función del código y se puede consultar utilizando la función 'help'.
- Corregida las funciones que compartían nombres con funciones integradas en Python.
- Las variables mantienen un nombre correcto siguiendo las 'buenas prácticas' de la programación.
- Se controlan los errores según el tipo de error que pueda producir el código en cuestión.

- Se asegura de que todas las variables del código tienen utilidad.
- Se elimina el uso de la función 'len' para la comprobación de listas vacías, siguiendo así la guía de estilos oficial de Python, PEP 8.
- Se ha ordenado correctamente la importación de módulos.

Finalmente, los datos obtenidos con pylint sobre la calidad del código son los siguientes:

```
Report
=====
579 statements analysed.

Statistics by type
-----
+-----+-----+-----+-----+-----+
|type |number|old number|difference|%documented|%badname |
+=====+=====+=====+=====+=====+
|module|1|1|!=|100.00|0.00|
+-----+-----+-----+-----+-----+
|class |5|5|!=|100.00|20.00|
+-----+-----+-----+-----+-----+
|method|17|17|!=|100.00|23.53|
+-----+-----+-----+-----+-----+
|function|11|11|!=|100.00|0.00|
+-----+-----+-----+-----+-----+


External dependencies
-----
:::

MySQLdb (TFG)
bs4 (TFG)

Raw metrics
-----
+-----+-----+-----+-----+
|type |number|%|previous|difference|
+=====+=====+=====+=====+
|code |594|68.43|NC|NC|
+-----+-----+-----+-----+
|docstring|26|3.00|NC|NC|
+-----+-----+-----+-----+
|comment|130|14.98|NC|NC|
+-----+-----+-----+-----+
|empty|118|13.59|NC|NC|
+-----+-----+-----+-----+


Duplication
-----
+-----+-----+-----+-----+
| |now|previous|difference|
+=====+=====+=====+=====+
|nb duplicated lines|0|0|!=||

+-----+-----+-----+-----+
|percent duplicated lines|0.000|0.000|!=||

+-----+-----+-----+-----+
```

Figura 12: Informe Pylint 1

Messages by category			
type	number	previous	difference
convention	0	0	=
refactor	0	0	=
warning	0	0	=
error	0	0	=

Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)

Figura 13: Informe Pylint 2

5.9. Pruebas y validación

5.9.1. Estrategia

Las pruebas realizadas se han llevado a cabo de dos formas diferentes, por un lado se han realizado pruebas manuales donde se programa un comportamiento y se comprueba, de forma empírica su funcionamiento. Por otro lado se han programado pruebas automáticas donde se preparan unos datos de entrada y otros datos de salida esperado para comprobar que, tras pasar la función oportuna, el funcionamiento es correcto devolviendo esos datos de salida esperado. Se mostrará una "checklist" con objetivos que se esperaban cumplir y su resultado final.

5.9.2. Descripción del entorno de pruebas

El equipo y entorno utilizado para las pruebas llevadas a cabo han sido:

- Asus A55V

- Sistema Operativo: Debian 8 (Jessie).
- Tamaño pantalla: 15,6 pulgadas.
- CPU: Intel® Core™ i7-3630QM de 4 núcleos.
- RAM: 6 GB.

- ZTE Blade V6

- Sistema Operativo: Google Android 5.0.2 (Lollipop).
- Tamaño pantalla: 5 pulgadas.
- CPU: Procesador de cuatro núcleos.
- RAM: 2 GB.

5.9.3. Pruebas unitarias

El objetivo de estas pruebas unitarias es el de controlar el correcto funcionamiento de determinados modulos y/o funciones del programa por separado, es decir, de forma independiente al resto del sistema.

A continuación se expondrá las pruebas unitarias automatizadas realizadas sobre la aplicación que realiza el web scraping con ayuda del framework unittest, así como el código utilizado para la realización de dicha prueba. El código mostrado es solo el de la ejecución de la prueba en cuestión, no se ha reflejado en esta documentación el código de las funciones que preparan ni limpian la prueba, el código de la aplicación así como el de las pruebas está completo en el enlace reflejado en la sección 5.8.2.

Código	CP-001
Nombre	Argumentos de entrada
Descripción	Prueba de los comandos de la aplicación reflejados en el menú de ayuda, que son el modo verbose (se activa con '-v'), la lectura de ficheros (se activa con '-x [ruta del fichero]') y el uso de los valores por defecto (se activa con '-d').

Tabla 49: CP-001 – Argumentos de entrada

```

1 #Prueba del modo VERBOSE, de la lectura de ficheros y del modo por defecto
2 def test_params(self):
3     """Testing verbose mode, reading files and default mode."""
4     i = 0
5     for param in sys.argv:
6         ret = options(param, i)
7         if ret == 1:
8             break
9         i += 1
10    self.assertTrue(VERBOSE)
11    self.assertEqual(XPATH, 'prueba.xml')
12    sys.argv = ['TFG.py', '-d']
13    i = 0
14    for param in sys.argv:
15        ret = options(param, i)
16        if ret == 1:
17            break
18        i += 1
19    self.assertFalse(VERBOSE)
20    self.assertEqual(XPATH, 'info.xml')
```

Código	CP-002
Nombre	Creación de la base de datos
Descripción	Prueba del correcto funcionamiento de la creación de la base de datos a través de los datos de acceso introducido en un archivo de configuración XML.

Tabla 50: CP-002 – Creación de la base de datos

```

1  #Datos de creacion de base de datos incorrectos y archivo XML no encontrado
2  def test_db_fail(self):
3      """Creating and overwriting database with wrong data and XML file not found."""
4      sys.argv = ['TFG.py', '-x', 'test_exist.xml', '-c']
5
6      #Archivo no encontrado
7      with self.assertRaises(SystemExit) as cm:
8          main()
9          self.assertEqual(cm.exception.code, 1)
10
11     sys.argv = ['TFG.py', '-x', 'test_2.xml', '-c']
12
13     t_info = ET.Element("info")
14     t_charset = ET.SubElement(t_info, "charset")
15     t_charset.text = "utf-8"
16     ET.SubElement(t_info, "database", host="localhost", db="", user="admin",
17                 password="admin")
18
19     arbol = ET.ElementTree(t_info)
20     arbol.write("test_2.xml")
21
22     #Dato en blanco
23     with self.assertRaises(SystemExit) as cm:
24         main()
25         self.assertEqual(cm.exception.code, 1)
26         os.remove("test_2.xml")

```

```
1 #Creacion de base de datos con datos correctos y sobreescritura de base de datos
2 def test_db_ok(self):
3     """Creating and overwriting database with correct data."""
4     #Preparacion de prueba
5     sys.argv = ['TFG.py', '-x', 'test.xml', '-c']
6     t_info = ET.Element("info")
7     t_charset = ET.SubElement(t_info, "charset")
8     t_charset.text = "utf-8"
9     t_database = ET.SubElement(t_info, "database", host="localhost", db="tfsg_test",
10                               user="admin", password="admin")
11    ET.SubElement(t_database, "rowsProduct", rowName="name", type="VARCHAR",
12                  size="128")
13    ET.SubElement(t_database, "rowsProduct", rowName="price", type="FLOAT",
14                  size="10")
15    ET.SubElement(t_database, "rowsComment", rowName="nick", type="VARCHAR",
16                  size="32")
17    ET.SubElement(t_database, "rowsComment", rowName="comment", type="TEXT",
18                  size="", isText="true")
19    ET.SubElement(t_database, "rowsComment", rowName="date", type="VARCHAR",
20                  size="32")
21
22    arbol = ET.ElementTree(t_info)
23    arbol.write("test.xml")
24
25    #Creacion inicial de la base de datos
26    with self.assertRaises(SystemExit) as cm:
27        main()
28    self.assertEqual(cm.exception.code, 0)
29
30    #Sobreescritura de la base de datos
31    sys.argv = ['TFG.py', '-x', 'test.xml', '-e']
32    with self.assertRaises(SystemExit) as cm:
33        main()
34    self.assertEqual(cm.exception.code, 0)
35
36    os.remove("test.xml")
37    query = "DROP TABLE analysis, comments, products, reports"
38    run_query(query)
```

Código	CP-003
Nombre	Datos obligatorios del archivo de configuración
Descripción	Comprobación del comportamiento del programa al faltar alguna de las variables fundamentales del fichero de configuración XML. Las variables que se comprueban son 'isText', 'urlBase', 'urlProducts' y 'urlComments'.

Tabla 51: CP-003 – Datos obligatorios del archivo de configuración

```

1 #Prueba que XML contiene ISTEXT
2 def test_isText(self):
3     """Test that XML file contains isText variable."""
4     #Preparacion de prueba
5     sys.argv = ['TFG.py', '-x', 'test.xml']
6     t_info = ET.Element("info")
7     t_charset = ET.SubElement(t_info, "charset")
8     t_charset.text = "utf-8"
9     t_database = ET.SubElement(t_info, "database", host="localhost", db="tfg_test",
10                               user="admin", password="admin")
11    ET.SubElement(t_database, "rowsProduct", rowName="name", type="VARCHAR",
12                  size="128")
13    ET.SubElement(t_database, "rowsProduct", rowName="price", type="FLOAT",
14                  size="10")
15    ET.SubElement(t_database, "rowsComment", rowName="nick", type="VARCHAR",
16                  size="32")
17    ET.SubElement(t_database, "rowsComment", rowName="comment", type="TEXT",
18                  size="") #No ponemos a true isText
19    ET.SubElement(t_database, "rowsComment", rowName="date", type="VARCHAR",
20                  size="32")
21
22    arbol = ET.ElementTree(t_info)
23    arbol.write("test.xml")
24
25    with self.assertRaises(SystemExit) as cm:
26        main()
27        self.assertEqual(cm.exception.code, 1)

```

```
1 #Prueba que el XML contiene urlBase
2 def test_urlBase(self):
3     """Test that XML file contains urlBase variable."""
4     #Preparacion de prueba
5     sys.argv = ['TFG.py', '-x', 'test.xml']
6     t_info = ET.Element("info")
7     t_charset = ET.SubElement(t_info, "charset")
8     t_charset.text = "utf-8"
9     t_database = ET.SubElement(t_info, "database", host="localhost", db="tfgr_test",
10                               user="admin", password="admin")
11    ET.SubElement(t_database, "rowsProduct", rowName="name", type="VARCHAR",
12                  size="128")
13    ET.SubElement(t_database, "rowsProduct", rowName="price", type="FLOAT",
14                  size="10")
15    ET.SubElement(t_database, "rowsComment", rowName="nick", type="VARCHAR",
16                  size="32")
17    ET.SubElement(t_database, "rowsComment", rowName="comment", type="TEXT",
18                  size="", isText="true") #No ponemos a true isText
19    ET.SubElement(t_database, "rowsComment", rowName="date", type="VARCHAR",
20                  size="32")
21
22    ET.SubElement(t_info, "url", urlBase="",
23                  urlProducts="http://www.marca.com/futbol/cadiz.html",
24                  urlComments="http://www.marca.com/servicios/noticias/comentarios/comunidad/"+
25                  "listar.html?noticia={numberproduct}&portal=5&pagina={npage_comments}")
26
27    arbol = ET.ElementTree(t_info)
28    arbol.write("test.xml")
29
30    with self.assertRaises(SystemExit) as cm:
31        main()
32        self.assertEqual(cm.exception.code, 1)
```

```
1 #Prueba que el XML contiene urlProducts
2 def test_urlProducts(self):
3     """Test that XML file contains urlProducts variable."""
4     #Preparacion de prueba
5     sys.argv = ['TFG.py', '-x', 'test.xml']
6     t_info = ET.Element("info")
7     t_charset = ET.SubElement(t_info, "charset")
8     t_charset.text = "utf-8"
9     t_database = ET.SubElement(t_info, "database", host="localhost", db="tfsg_test",
10                               user="admin", password="admin")
11    ET.SubElement(t_database, "rowsProduct", rowName="name", type="VARCHAR",
12                  size="128")
13    ET.SubElement(t_database, "rowsProduct", rowName="price", type="FLOAT",
14                  size="10")
15    ET.SubElement(t_database, "rowsComment", rowName="nick", type="VARCHAR",
16                  size="32")
17    ET.SubElement(t_database, "rowsComment", rowName="comment", type="TEXT",
18                  size="", isText="true") #No ponemos a true isText
19    ET.SubElement(t_database, "rowsComment", rowName="date", type="VARCHAR",
20                  size="32")
21
22    ET.SubElement(t_info, "url", urlBase="http://www.marca.com",
23                  urlComments="http://www.marca.com/servicios/noticias/comentarios/comunidad/" +
24                  "listar.html?noticia={numberproduct}&portal=5&pagina={npage_comments}")
25
26    arbol = ET.ElementTree(t_info)
27    arbol.write("test.xml")
28
29    with self.assertRaises(SystemExit) as cm:
30        main()
31        self.assertEqual(cm.exception.code, 1)
```

```
1 #Prueba que el XML contiene urlComments
2 def test_urlComments(self):
3     """Test that XML file contains urlComments variable."""
4     #Preparacion de prueba
5     sys.argv = ['TFG.py', '-x', 'test.xml']
6     t_info = ET.Element("info")
7     t_charset = ET.SubElement(t_info, "charset")
8     t_charset.text = "utf-8"
9     t_database = ET.SubElement(t_info, "database", host="localhost", db="tfsg_test",
10                               user="admin", password="admin")
11    ET.SubElement(t_database, "rowsProduct", rowName="name", type="VARCHAR",
12                  size="128")
13    ET.SubElement(t_database, "rowsProduct", rowName="price", type="FLOAT",
14                  size="10")
15    ET.SubElement(t_database, "rowsComment", rowName="nick", type="VARCHAR",
16                  size="32")
17    ET.SubElement(t_database, "rowsComment", rowName="comment", type="TEXT",
18                  size="", isText="true") #No ponemos a true isText
19    ET.SubElement(t_database, "rowsComment", rowName="date", type="VARCHAR",
20                  size="32")
21
22    ET.SubElement(t_info, "url", urlBase="http://www.marca.com",
23                  urlProducts="http://www.marca.com/futbol/cadiz.html",
24                  urlComments="")
25
26    arbol = ET.ElementTree(t_info)
27    arbol.write("test.xml")
28
29    with self.assertRaises(SystemExit) as cm:
30        main()
31        self.assertEqual(cm.exception.code, 1)
```

Código	CP-004
Nombre	Funcionamiento de la función is_repit
Descripción	Se comprueba el correcto funcionamiento de la función is_repit, a la cual se le proporciona un comentario y el id de un producto y devolverá si ese comentario ya está introducido en la base de datos para ese producto.

Tabla 52: CP-004 – Funcionamiento de la función is_repit

```

1 def test_repit(self):
2     """Testing is_repit function."""
3     #Debe dar verdadero
4     repit = is_repit('Este comentario es una prueba de verificación para el ',
5                      +'método que comprueba si hay comentarios repetidos.', '123456')
6     self.assertTrue(repit)
7
8     #Debe dar falso por el id del product
9     repit = is_repit('Este comentario es una prueba de verificación para el método ',
10                      +'que comprueba si hay comentarios repetidos.', '213456')
11    self.assertFalse(repit)
12
13    #Debe dar falso por el comentario
14    repit = is_repit('Este comentario es una prueba de verificacion para el metodo ',
15                      +'que comprueba si hay comentarios repetidos.', '123456')
16    self.assertFalse(repit)

```

A continuación se relatan las pruebas realizadas a la aplicación web sobre funcionalidades concretas:

Código	CP-005
Nombre	Login
Descripción	Se prueba que solo se accede a la aplicación introduciendo las credenciales de un usuario registrado y que haya sido activado. Además, se prueba que el Login conduce hacia una vista diferente dependiendo del rol asignado a dicho usuario.

Tabla 53: CP-005 – Login

Código	CP-006
Nombre	Registro y validación
Descripción	Se comprueba que se necesita llenar todo el formulario de registro para ser aceptado y que si se introduce un nombre de usuario ya existente se avisa al usuario para que pruebe con otro. Se comprueba que al proceder a un registro correcto se envía un email automático al correo electrónico introducido y que el usuario que se guarda en la base de datos está sin activar, es decir, que con ese usuario recién creado no se puede acceder a la aplicación web.

Tabla 54: CP-006 – Registro y validación

Código	CP-007
Nombre	Olvido de contraseña
Descripción	Se prueba que para restablecer la contraseña es necesario introducir un nombre de usuario y un email correcto y coincidentes, y que la nueva contraseña aleatoria es enviada a dicho email.

Tabla 55: CP-007 – Olvido de contraseña

Código	CP-008
Nombre	Cambio de contraseña
Descripción	Se prueba que, en el menú de configuración de un usuario logueado, se puede modificar correctamente la contraseña, y que para que la modificación sea satisfactoria hay que introducir la nueva contraseña en dos ocasiones y que ambas sean coincidentes.

Tabla 56: CP-008 – Cambio de contraseña

Código	CP-009
Nombre	Activar usuario
Descripción	Se prueba que la lista de usuarios para activar son los usuarios que en la base de datos tienen el atributo de 'active' a 0, y se comprueba que al pulsar sobre el botón 'Activar' de uno de estos usuarios ese atributo es modificado en la base de datos y desaparece de la lista.

Tabla 57: CP-009 – Activar usuario

Código	CP-010
Nombre	Modificar datos de otro usuario
Descripción	Se prueba que desde un usuario administrador se modifica correctamente cualquiera de los datos de otro usuario. Y que al modificar un atributo solo ese atributo y sobre el usuario correcto se efectúa el cambio en la base de datos.

Tabla 58: CP-010 – Modificar datos de otro usuario

Código	CP-011
Nombre	Registrar administrador
Descripción	Se comprueba que desde un usuario administrador se puede proceder a registrar a otro administrador. Se confirma que es necesario llenar todos los campos del formulario, que el usuario creado tiene por defecto asignado el rol de administrador y está activo y que se valida correctamente el hecho de que no exista algún usuario con el nombre de usuario que se está introduciendo.

Tabla 59: CP-011 – Registrar administrador

Código	CP-012
Nombre	Registrar cliente usuario
Descripción	Se prueba que desde el rol 'admin_customer' se puede registrar nuevos usuarios, que para el registro es necesario llenar todos los campos del formulario y que su creación asigna, por defecto, el rol 'user_customer' al usuario creado.

Tabla 60: CP-012 – Registrar cliente usuario

Código	CP-013
Nombre	Modificar datos de otro usuario desde 'admin_customer'
Descripción	Se prueba que desde un usuario con el rol de administrador de cliente se modifica correctamente cualquiera de los datos de otro usuario asignado a ese mismo cliente. Y que al modificar un atributo solo ese atributo y sobre el usuario correcto se efectúa el cambio en la base de datos. Además se comprueba que en lo que respecta al rol solo se puede asignar los roles 'admin_customer' y 'user_customer' en la modificación.

Tabla 61: CP-013 – Modificar datos de otro usuario desde 'admin_customer'

Código	CP-014
Nombre	Puntuar comentarios
Descripción	Se comprueba que aparecen un máximo de diez comentarios en la lista a puntuar, pero que pueden ser menos. Se prueba que los comentarios son cargados en su zona correcta, es decir, si nunca han sido valorados en la sección 'Comentarios no valorados' y si ya han sido valorados alguna vez en la sección 'Comentarios ya valorados'. Se prueba su correcta introducción en la tabla 'analysis' de la base de datos y que los datos de esta tabla son cargados correctamente cuando se entra en 'Comentarios ya valorados'.

Tabla 62: CP-014 – Puntuar comentarios

Código	CP-015
Nombre	Generar Excel
Descripción	Se comprueba que el fichero Excel generado se corresponde con los datos almacenados en la base de datos y que los filtros opcionales son tenido en cuenta. Se comprueba que el formato del Excel generado y su visualización son correctas.

Tabla 63: CP-015 – Generar Excel

Código	CP-016
Nombre	Crear y editar informes
Descripción	Se comprueba que el formulario de los informes se muestra correctamente y la obligatoriedad de que se introduzcan los datos relacionados con la descripción del informe y el 'select' de la consulta. Se comprueba que cada casilla del formulario coincide con su atributo correspondiente en la base de datos, y que al entrar en modificar un informe, aparecen los atributos correspondiente a dicho informe cargados.

Tabla 64: CP-016 – Crear y editar informes

Código	CP-017
Nombre	Mostrar gráficos
Descripción	Se comprueba que ante un informe con una consulta bien creada se muestra correctamente el gráfico seleccionado (barras, pastel o líneas) con sus datos correspondiente.

Tabla 65: CP-017 – Mostrar gráficos

Código	CP-018
Nombre	Reportar error en un informe
Descripción	Se comprueba que al reportar el error en un informe seleccionado se envía correctamente un correo electrónico a los administradores de dicho cliente que ha reportado el error, y que en el correo electrónico viene introducida la identificación del informe seleccionado así como la descripción del problema que haya redactado el usuario que ha realizado el reporte.

Tabla 66: CP-018 – Reportar error en un informe

5.9.4. Pruebas de integración

En las pruebas de integración se verifica el correcto funcionamiento de la unión de varios modulos. Para la aplicación del web scraping se ha vuelto a utilizar el framework unittest con una última prueba que verifica el funcionamiento de toda la aplicación al completo. En este documento se mostrará el código de ejecución de la prueba, pero no el código de preparación ni de limpieza anterior y posterior respectivamente. El código completo está referenciado en la sección 5.8.2. Para las pruebas realizadas sobre la aplicación web se relatará el proceso llevado a cabo para su verificación.

Código	CP-019
Nombre	Prueba completa de la aplicación de web scraping
Descripción	Se comprueba el correcto funcionamiento de todos los modulos de la aplicación que realiza el web scraping. Para ello, utilizando unittest y Element Tree XML de Python, se ha prepara un escenario inicial donde se genera un archivo XML con una estructura y datos de una página de prueba la cual se conoce su contenido y se crea una base de datos de prueba, todo esto será eliminado tras la realización del test. En el código de prueba se comprobará que dicha base de datos está vacía antes de empezar la aplicación y que, tras finalizar, se habrá llenado correctamente la base de datos y que dicho código habrá finalizado su ejecución sin arrojar ningún error.

Tabla 67: CP-019 – Prueba completa de la aplicación de web scraping

```

1 #Dado un archivo de entrada completo debe llenar la base de datos
2 def test_complete_ok(self):
3     """Given a complete input file you must fill in the database."""
4     #Comprobamos que la base de datos esta vacia
5     query = "Select count(c.autoid), count(p.autoid) from comments as c join"+
6             "products as p on c.idProduct = p.id"
7     result = run_query(query)
8     for row in result:
9         count_comment = row[0]
10        count_product = row[1]
11
12    #Comprobamos que no hay comentarios ni productos antes de comenzar la pruebas
13    self.assertEqual(count_comment, 0)
14    self.assertEqual(count_product, 0)
15
16    #Ejecutamos el programa con el fichero recien creado y activando el modo detallado
17    sys.argv = ['TFG.py', '-v', '-x', 'test.xml']
18    main()
19
20    #Volvemos a comprobar el numero de comentarios y productos ahora
21    query = "Select count(c.autoid), count(p.autoid) from comments as c"+
22            "join products as p on c.idProduct = p.id"
23    result = run_query(query)
24    for row in result:
25        count_comment = row[0]
26        count_product = row[1]
27
28    #Comprobamos que se ha introducido correctamente comentarios y productos
29    self.assertGreater(count_comment, 0)
30    self.assertGreater(count_product, 0)

```

Código	CP-020
Nombre	Prueba completa de la aplicación web para administrador
Descripción	Se comprueba que al iniciar sesión con el rol de administrador solo se puede acceder a las funcionalidades particulares de clientes a la espera de confirmación, activación de clientes, visualización y edición de usuarios y registra nuevos administradores. Se comprueba que la navegación entre las diferentes funcionalidades es correcta.

Tabla 68: CP-020 – Prueba completa de la aplicación web para administrador

Código	CP-021
Nombre	Prueba completa de la aplicación web para cliente administrador
Descripción	Se comprueba que al iniciar sesión con el rol de cliente administrador solo se puede acceder a las funcionalidades particulares de usuarios activos del cliente actual, edición de estos, registro de nuevos usuarios, descarga y valoración de comentarios y visualización, edición y creación de informes. Se comprueba que la navegación entre las diferentes funcionalidades es correcta.

Tabla 69: CP-021 – Prueba completa de la aplicación web para cliente administrador

Código	CP-022
Nombre	Prueba completa de la aplicación web para cliente usuario
Descripción	Se comprueba que al iniciar sesión con el rol de cliente usuario solo se puede acceder a las funcionalidades particulares de visualización y reporte de informes activos y descarga de comentarios. Se comprueba que la navegación entre las diferentes funcionalidades es correcta.

Tabla 70: CP-022 – Prueba completa de la aplicación web para cliente usuario

5.9.5. Pruebas de sistema

En esta sección se verifica que el sistema que se ha construido cumple los objetivos iniciales propuestos. Para ello se ha comprobado el debido cumplimiento de las secciones 5.2.1 y 5.2.2 en los equipos mencionados en la sección 5.9.2, para de esta forma verificar el funcionamiento y la visualización correcta de la aplicación web en PC y smartphone.

5.9.6. Pruebas de aceptación

El panel de administración que proporciona la aplicación web es un panel clásico y simple, que hará que el usuario tenga facilidad para acceder a las funcionalidades deseada.

Las pruebas de aceptación han sido enfocadas principalmente a la tarea más compleja del proceso, que es la generación de un archivo de configuración XML funcional, para ello, se ha realizado la creación del archivo XML con varias webs diferentes entre ellas. A continuación se expondrá dos archivos XML de ejemplo de dos webs clientes objetivos muy diferentes entre sí,

por un lado tendremos la web <http://www.marca.com/> donde los productos son las noticias de la sección de un equipo concreto y los comentarios a almacenar son los de cada una de las noticias. Por otro lado tendremos la web <https://www.pcccomponentes.com/>, donde los productos serán productos generales de la web y los comentarios a almacenar serán las opiniones de los clientes para cada producto.

A continuación se muestra el código de los archivos de configuración XML para las webs <http://www.marca.com/> y <https://www.pcccomponentes.com/> respectivamente, ambos código junto con el resto de la información también están disponible en los enlaces mencionados en la sección 5.8.2.

```
1 <info>
2   <charset>iso-8859-15</charset>
3   <database host="localhost" db="tfg_marca" user="admin" password="admin">
4     <rowsProduct rowName="name" type="VARCHAR" size="128"></rowsProduct>
5     <rowsComment rowName="nick" type="VARCHAR" size="32"></rowsComment>
6     <rowsComment rowName="comment" type="TEXT" size="" isText="true"></rowsComment>
7     <rowsComment rowName="date" type="VARCHAR" size="32"></rowsComment>
8   </database>
9   <url urlBase="http://www.marca.com"
10    urlProducts="http://www.marca.com/futbol/cadiz.html"
11    urlComments="http://www.marca.com/servicios/noticias/comentarios/comunidad/
12    listar.html?noticia={numberproduct}&portal=5&pagina={npage_comments}">
13  </url>
14  <urlProducts categorypage_max=""> <!-- ultima pagina a la que debe llegar -->
15    <!-- Indicar como llegar hasta el enlace para cada producto -->
16    <linkProduct tag="header" attr="class" valueAttr="mod-header"></linkProduct>
17    <linkProduct tag="h3" attr="class" valueAttr="mod-title"></linkProduct>
18    <linkProductFinal tag="a" attr="itemprop" valueAttr="url" attrGoal="href">
19      <!-- Indicar los atributos que se desean guardar de la web de cada producto -->
20      <attributeProducts tag="meta" attr="name" valueAttr="twitter:title"
21        tagID="false" rowName="name"></attributeProducts>
22      <!-- Indicar como llegar hasta la ID necesaria para entrar en los comentarios -->
23      <idProduct tag="aside" attr="class"
24        valueAttr="aside-comments right-panel comments-panel"></idProduct>
25      <idProduct tag="div" attr="class"
26        valueAttr="no-visible comentarios comentarios-retro"></idProduct>
27      <idProductFinal tag="div" attr="class"
28        valueAttr="js-comments-container" attrGoal="data-commentid"></idProductFinal>
29  </urlProducts>
30  <urlComments>
31    <attributeComments tag="span" attr="class"
32      valueAttr="nombre_usuario" tagID="false" rowName="nick"></attributeComments>
33    <attributeComments tag="span" attr="class" valueAttr="fecha"
34      tagID="false" rowName="date"></attributeComments>
35    <attributeComments tag="div" attr="class" valueAttr="comentario"
36      tagID="false" rowName="comment"></attributeComments>
37  </urlComments>
38 </info>
```

```
1 <info>
2     <charset>utf-8</charset>
3
4     <database host="localhost" db="tfg_1" user="admin" password="admin">
5         <rowsProduct rowName="name" type="VARCHAR" size="128"></rowsProduct>
6         <rowsProduct rowName="price" type="FLOAT" size="10"></rowsProduct>
7         <rowsProduct rowName="brand" type="VARCHAR" size="64"></rowsProduct>
8         <rowsComment rowName="nick" type="VARCHAR" size="32"></rowsComment>
9         <rowsComment rowName="comment" type="TEXT" size="">
10            isText="true"></rowsComment>
11         <rowsComment rowName="rating" type="FLOAT" size=""></rowsComment>
12         <rowsComment rowName="date" type="VARCHAR" size="32"></rowsComment>
13     </database>
14
15     <url urlBase="https://www.pccomponentes.com">
16         urlProducts="https://www.pccomponentes.com/listado/
17         ajax?page={npage_products}&order=new&idFamilies[]={categorypage}"
18         urlComments="https://www.pccomponentes.com/a
19         /comments?page={npage_comments}&order=new&idArticle={numberproduct}">
20     </url>
21
22     <urlProducts categorypage_max="3"> <!-- ultima pagina a la que debe llegar -->
23         <!-- Indicar como llegar hasta el enlace para cada producto -->
24         <linkProduct tag="div" attr="class"
25             valueAttr="tarjeta-articulo__elementos-basicos"></linkProduct>
26         <linkProduct tag="header" attr="class"
27             valueAttr="tarjeta-articulo__nombre"></linkProduct>
28         <linkProduct tag="h3" attr="itemprop" valueAttr="name"></linkProduct>
29         <linkProductFinal tag="a" attr="class"
30             valueAttr="GTM-productClick enlace-disimulado" attrGoal="href">
```

```
1      <!-- Indicar los atributos que se desean guardar de cada producto -->
2      <!-- Nombre -->
3      <attributeProducts tag="h1" attr="itemprop" valueAttr="name" tagID="false"
4          rowName="name"></attributeProducts>
5      <!-- Precio -->
6      <attributeProducts tag="span" attr="class" valueAttr="baseprice"
7          tagID="false" rowName="price"></attributeProducts>
8      <!-- Marca -->
9      <attributeProducts tag="a" attr="itemprop" valueAttr="brand"
10         tagID="false" rowName="brand"></attributeProducts>
11      <!-- Indicar como llegar hasta la ID para entrar en los comentarios -->
12      <idProduct tag="body" attr="id" valueAttr="ficha-producto"></idProduct>
13      <idProductFinal tag="div" attr="id"
14         valueAttr="contenedor-principal" attrGoal="data-id"><idProductFinal>
15      <urlProducts/>
16
17      <urlComments>
18          <attributeComments tag="p" attr="class"
19              valueAttr="user-data m-b-0" tagID="false" rowName="nick">
20          </attributeComments>
21          <attributeComments tag="div" attr="class"
22              valueAttr="opinion" tagID="false" rowName="comment">
23          </attributeComments>
24          <attributeComments tag="input" attr="class"
25              valueAttr="rating-input" tagID="value" rowName="rating">
26          </attributeComments>
27          <attributeComments tag="p" attr="class"
28              valueAttr="fecha pull-sm-right hidden-sm-down m-b-0"
29              tagID="false" rowName="date"></attributeComments>
30
31      </urlComments>
32  </info>
```

En las pruebas se ha visto que para llenar el archivo de configuración XML se requiere de ciertos conocimientos informáticos para el análisis del contenido y funcionamiento de la web objetivo. Además de la obligación de mantener actualizado dicho archivo cuando la web objetivo modifique su estructura. Por esto se propone como opción al cliente una versión alternativa donde el mantenimiento y la construcción del fichero de configuración XML sea cargo del equipo de administración de la aplicación, elevando el pago mensual. Las dos opciones que se ofrecen a los clientes están explicadas en la sección 1.3.5.

6. Epílogo

6.1. Manual de instalación

A continuación se detalla el proceso de descarga e instalación. Todos los archivos necesarios están disponibles en los enlaces mencionados en el capítulo 5.8.2. Se recomienda utilizar servidores Linux para el despliegue de las aplicaciones relacionadas con este proyecto.

6.1.1. Despliegue y configuración de la aplicación web

Para el despliegue de la aplicación web se requiere de un servidor Apache y de la implementación, en el mismo servidor o en otro, de las bases de datos MySQL asociadas a la web. También será necesario tener instalado PHP 5.6 o superior. Para proceder a instalar estos componentes necesarios, en un servidor Linux se haría de la siguiente forma:

```
1 $ sudo apt-get update
2 $ sudo apt-get install apache2
3 $ sudo apt-get install mysql-server php5 mysql
4 $ sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql
```

Con esto ya tendremos instalado Apache, PHP y MySQL.

Se deberá configurar los datos de acceso al servidor MySQL así como a la base de datos donde se almacenará la información de los usuarios de la aplicación web. La configuración de la base de datos, tanto en el servicio MySQL como en el archivo CodeIgniter, es esencial para garantizar la seguridad de la aplicación web. En MySQL se crearán dos usuarios, uno que pueda ver todas las bases de datos de la aplicación y que tenga permitido el SELECT, INSERT, UPDATE y DELETE en todas las tablas, y otro usuario administrador con las mismas acciones permitidas pero que solo pueda ver las bases de datos de los clientes, es decir, no pueda tener acceso ni a la base de datos donde se almacena los nombres de usuarios y contraseñas de la aplicación web ni a las bases de datos propias de MySQL. Con esto evitamos que en la zona de mayor peligro en cuanto a seguridad, que es la zona de creación de informes, se pueda obtener información delicada a través de la tabla 'mysql.user' o de funciones como 'current_user()', por ejemplo. Posteriormente en CodeIgniter, el usuario con acceso a todas las bases de datos se configurará en la variable \$db['default'], y el usuario que solo puede acceder a las bases de datos de clientes se indicará en la variable \$db['personal'], ambas variables están situadas en application/config/database.php. En la variable \$config['base_url'] definida en application/config/config.php se indicará la ruta donde está situada la aplicación web.

Cuando se registre un cliente nuevo se le creará una base de datos y un usuario asociado a esa base de datos. Ese usuario solo podrá ver y manejar su propia base de datos, y los datos de acceso de ese nuevo usuario son los que se enviará al cliente.

6.1.2. Despliegue y configuración de la aplicación de web scraping

Para poder ejecutar la aplicación web en Python es necesario tener instalado Python3 y los componentes requeridos, que son BeautifulSoup, el modulo Requests, MySQLdb y el procesa-

dor de archivos XML lxml. Para obtener esto instalaremos en primer lugar Python3, aunque en servidores Linux suelen venir instalados por defecto. Podemos comprobar si lo tenemos instalado de la siguiente forma:

```
1 $ python3 --version
```

Y en caso de que no lo tengamos, procedemos a instalarlo:

```
1 $ sudo apt-get install python3.4
```

Ahora instalaremos el gestor de paquetes pip, el cual utilizaremos para instalar el resto de paquetes necesarios. Para instalar pip introducimos:

```
1 $ sudo apt-get install python3-pip
```

Haciendo uso de pip procedemos a instalar Beautiful Soup y el module Requests:

```
1 $ sudo pip3 install beautifulsoup4
2 $ sudo pip3 install requests
```

Por último instalamos MySQLdb y el procesador XML para Python de la siguiente forma:

```
1 $ sudo apt-get install python3-mysqldb
2 $ sudo apt-get install python3-lxml
```

Con todo lo anterior instalado podemos pasar a ejecutar correctamente la aplicación haciendo uso de Python 3. Para su correcto funcionamiento es necesario que el archivo de configuración XML esté bien redactado y los datos sean válidos. La forma de redacción y el funcionamiento del archivo de configuración se explica detalladamente en la sección 6.2.

6.2. Manual de uso

El programa ejecutable depende de un archivo de configuración XML que puede ser completado por el usuario y que se explicará a continuación. Este archivo de configuración requiere de conocimientos informáticos para ser completado correctamente, por esto mismo se ofrece al cliente la opción de una versión ampliada de la aplicación en la que ellos no tendrían que encargarse de la realización ni del mantenimiento del archivo de configuración.

```
Web Info Download
usage: python WID.py [-h] [-v] [-x xPath] [-d] [-c] [-e]
Command Summary:
-h                  help
-v                  verbose mode
-x                  XML file path
-d                  default values
-c                  create database
-e                  create database and delete if exists
```

Figura 14: Pantalla del menú de ayuda del programa de web scraping

El ejecutable, como podemos observar en la imagen 14, cuenta con los siguientes comandos:

- -h 'Ayuda': Muestra el menú de ayuda que se ve en la imagen 14.
- -v 'Modo detallado': Activa el modo detallado, informando por pantalla de todo lo que la aplicación va ejecutando.
- -x 'Ruta del archivo XML': Se utiliza para introducir el nombre y/o ruta del archivo de configuración que se utilizará para ejecutar el código.
- -d 'Valores por defecto': Toma los valores por defecto, es decir, sin creación de base de datos, sin modo detallado y buscando un archivo de configuración XML situado a la misma altura que el código Python y llamado info.xml.
- -c 'Crear las tablas de la base de datos': Crea las tablas de la base de datos utilizando los datos de la etiqueta correspondiente del archivo de configuración. Es necesario que el archivo de configuración esté bien realizado, indicándole en él las tablas que se quieren crear así como los datos de acceso a la base de datos donde se generará la estructura de tablas.
- -e 'Crear las tablas de la base de datos borrando las existentes': Similar a la opción '-c' pero, en caso de que la base de datos objetivo tenga tablas y contenido este será eliminado. Sirve para restablecer el contenido de la base de datos.

Para la utilización del ejecutable es necesario llenar el archivo de configuración, esto se va a tratar de explicar a continuación. El archivo de configuración es un documento XML compuesto por una serie de etiquetas que dará valor a ciertos atributos. En primer lugar se explica la composición del archivo de configuración y, posteriormente, la forma de obtener

los datos necesarios para rellenarlo. Finalmente se presenta un repaso de las capacidades y opciones que ofrece la aplicación web.

6.2.1. Archivo de configuración

El archivo de configuración está compuesto por las siguientes secciones:

- Inicio y codificación:

```
<info>
<charset>utf-8</charset>
```

Figura 15: Codificación

El archivo de configuración se abre con una etiqueta <info> y dicha etiqueta ocupará todo el código del fichero. Dentro de esta etiqueta, lo que tendremos en primer lugar es la codificación de la página web objetivo. Esta codificación estará indicada entre las etiquetas <charset>

- Atributos de la base de datos:

```
<database host="localhost" db="tfg_1" user="admin" password="admin">
  <rowsProduct rowName="name" type="VARCHAR" size="128"></rowsProduct>
  <rowsProduct rowName="price" type="FLOAT" size="10"></rowsProduct>
  <rowsProduct rowName="brand" type="VARCHAR" size="64"></rowsProduct>

  <rowsComment rowName="nick" type="VARCHAR" size="32"></rowsComment>
  <rowsComment rowName="comment" type="TEXT" size="" isText="true"></rowsComment>
  <rowsComment rowName="rating" type="FLOAT" size=""></rowsComment>
  <rowsComment rowName="date" type="VARCHAR" size="32"></rowsComment>
</database>
```

Figura 16: Base de datos

Entre la etiqueta <database> estará toda la información relacionada a la base de datos. En primer lugar, como podemos observar en la imagen, la etiqueta database tendrá unos atributos que indicarán la IP de la base de datos, el nombre y el usuario y password con el que acceder. Esta información es ofrecida al cliente una vez el registro en la aplicación se haya completado. Dentro de la etiqueta <database> se definirán los atributos que queramos obtener de la página web objetivo, separando los atributos relacionados con los productos (etiqueta <rowsProduct>) de los relacionados con los comentarios (etiqueta <rowsComment>). Para cada atributo indicaremos el nombre, el tipo y el tamaño de caracteres que ocupará. Además, indicaremos poniendo el atributo 'isText' a "true" cual de los atributos relacionados con comentarios será un texto (en la imagen es el segundo atributo de las etiquetas <rowsComment>).

- URL de la web objetivo:

```
<url urlBase="https://www.pccomponentes.com" urlProducts="https://www.pccomponentes.com/listado/
ajax?page={npage_products}&order=new&idFamilies[]={categorypage}" urlComments="https://www.pccomponentes.com/
a/comments?page={npage_comments}&order=new&idArticle={numberproduct}">
</url>
```

Figura 17: URL web objetivo

En la etiqueta <url> colocaremos como atributo tres direcciones web, que son:

- urlBase: Dirección principal de la página web objetivo.
- urlProducts: Dirección donde se listarán todos los productos que nos interesan de la web objetivo. Desde aquí se debe obtener el enlace a cada uno de los productos y dentro de ese enlace es donde se saca la información a almacenar de cada producto así como la identificación necesaria del producto para poder acceder a sus comentarios. Como podemos observar en la imagen tenemos disponible dos variables numéricas que podemos utilizar, una es *{npage_products}* que recorrerá las diferentes páginas de la url, y la otra *{categorypage}* que recorrerá las diferentes categorías en caso de que existan.
- urlComments: Dirección donde están listados los comentarios. Aquí también disponemos de dos variables a utilizar, una es *{npage_comments}* que estará situada en el lugar de la URL donde está situado el número de página de los comentarios, y *{numberproduct}* que irá situada en el lugar donde está el ID para identificar el producto cuyos comentarios estamos viendo. Si los comentarios están listados en una única página se puede obviar la variable *{npage_comments}*. *{numberproduct}* es la identificación que se obtendrá de la web del producto, no tiene por qué ser un ID numérico, puede ser parte de una URL, por ejemplo.

■ Productos:

```
<urlProducts categorypage_max="3"> <!-- categorypage_max es la ultima pagina a la que debe llegar -->
<!-- Indicar como llegar hasta el enlace (atributo href necesario) para cada producto -->
<linkProduct tag="div" attr="class" valueAttr="tarjeta-articulo_elementos-basicos"></linkProduct>
<linkProduct tag="header" attr="class" valueAttr="tarjeta-articulo_nombre"></linkProduct>
<linkProduct tag="h3" attr="itemprop" valueAttr="name"></linkProduct>
<linkProductFinal tag="a" attr="class" valueAttr="GTM-productClick enlace-disimulado" attrGoal="href">

<!-- Indicar los atributos que se desean guardar de la web de cada producto -->
<!-- ID -->
<!-- Nombre -->
<attributeProducts tag="h1" attr="itemprop" valueAttr="name" tagID="false" rowName="name"></attributeProducts>
<!-- Precio -->
<attributeProducts tag="span" attr="class" valueAttr="baseprice" tagID="false" rowName="price"></attributeProducts>
>
<!-- Marca -->
<attributeProducts tag="a" attr="itemprop" valueAttr="brand" tagID="false" rowName="brand"></attributeProducts>

<!-- Indicar como llegar hasta la ID necesaria para entrar en los comentarios (el numberproduct) -->
<idProduct tag="body" attr="id" valueAttr="ficha-producto"></idProduct>
<idProductFinal tag="div" attr="id" valueAttr="contenedor-principal" attrGoal="data-id"></idProductFinal>

</urlProducts>
```

Figura 18: Productos

Como se indicó anteriormente, en el siguiente punto de este manual se explicará como obtener los valores de los atributos de la página web objetivo. En este punto nos centramos

en la composición de las etiquetas. En este caso, entre la etiqueta <urlProducts> va definido como localizar dentro de la página web objetivo cada uno de los atributos de los productos que queremos almacenar, así como el enlace a la web de cada producto y el identificador. Además, urlProducts tiene un atributo llamado 'categorypage_max' que es el número de categorías de productos que tiene la web (en caso de que la web se divida en categorías), es decir, sería el número máximo al que debe llegar la variable {categorypage} en caso de haberse utilizado dicha variable.

Dentro de la etiqueta urlProducts tenemos varias etiquetas más, estas son:

- <linkProduct>: Las etiquetas linkProduct nos indica como llegar hasta el enlace individual de cada uno de los productos que se muestran en el listado del enlace urlProducts. Es decir, usando la imagen 18 como ejemplo, en la primera etiqueta linkProduct se obtiene todas las etiquetas de la página web que sean 'div' y que tengan un atributo 'class' cuyo valor sea 'tarjeta-articulo_elementos-basicos'. Luego, dentro de los que cumpla este requisito buscará el segundo linkProduct. Podemos poner tantos linkProduct como necesitemos hasta llegar a la etiqueta linkProductFinal, que será con la que concluyamos e indicará donde está el enlace que necesitamos (el enlace a la web de cada producto), que en el ejemplo de la imagen sería en la etiqueta 'a' que tenga un atributo 'class' cuyo valor sea 'GTM-productClick enlace-disimulado' y nos quedamos con el valor del atributo 'href' de esa etiqueta.
- <attributeProducts>: Esta etiqueta nos indica como, dentro de la web de cada uno de los productos, se obtiene la información que queremos almacenar en la base de datos. En el ejemplo de la imagen 18 almacenamos tres atributos relacionados con los productos, el primero de ellos es el nombre ('name' definido en el atributo 'rowName' de la etiqueta 'attributeProducts') y está situado en la etiqueta 'h1' que tiene un atributo llamado 'itemprop' cuyo valor es 'name'. El atributo 'tagID' sirve para indicar si tomamos el valor de algún atributo concreto de la etiqueta puesta en 'tag' (que habría que indicarse en 'tagID') o si tomamos el valor del contenido que está dentro de la etiqueta de 'tag' (en ese caso ponemos false).

De esta misma forma obtendríamos cada valor que queramos almacenar en la base de datos. El 'rowName' debe coincidir con el indicado en la definición de la base de datos para los productos.

Los atributos 'attr' y 'valueAttr' pueden estar vacíos si la etiqueta en la página web objetivo no tiene nada que la identifique, pero en caso de que esto sea así habría que vigilar si solo con el 'tag' se puede obtener el dato correctamente de la web, sin ser confundido con otro.

- <idProduct>: Esta etiqueta señala como llegar hasta la identificación que necesitamos para entrar en los comentarios de cada uno de los productos. Dicho valor tendremos que localizarlo en la página web de la información de un producto concreto, y será el que se utilizará para la variable {numberproduct} citada anteriormente. Su funcionamiento es igual al de la etiqueta <linkProduct>, es decir, iremos indicando con las etiquetas <idProduct> como llegar hasta la zona donde está el valor que necesitamos y finalizaremos con la etiqueta <idProductFinal> indicando en qué etiqueta está el valor que queremos y señalando en la variable 'attrGoal' en qué atributo de dicha etiqueta está el valor que vamos a utilizar para acceder a los comentarios de ese producto.

La identificación puede ser un valor numérico o una parte de una URL, en cualquier caso, debe ser algo que cuando se incluya en la 'urlComments' identifique los comentarios del producto concreto en cuestión.

- Comentarios:

```

<urlComments>
  <attributeComments tag="p" attr="class" valueAttr="user-data m-b-0" tagID="false" rowName="nick"></
  attributeComments>
  <attributeComments tag="div" attr="class" valueAttr="opinion" tagID="false" rowName="comment"></attributeComments>
  <attributeComments tag="input" attr="class" valueAttr="rating-input" tagID="value" rowName="rating"></
  attributeComments>
  <attributeComments tag="p" attr="class" valueAttr="fecha pull-sm-right hidden-sm-down m-b-0" tagID="false" rowName
  ="date"></attributeComments>

</urlComments>
</info>

```

Figura 19: Comentarios

En la etiqueta <urlComments> se definirá cada uno de los atributos y valores con respecto a los comentarios que queremos almacenar en la base de datos. Para ello utilizaremos la etiqueta <attributeComments> de la misma forma que anteriormente hemos utilizado la etiqueta <attributeProducts>. Es decir, indicamos en rowName el nombre con el que almacenaremos dicho valor, y localizaremos el valor dentro de la página web de los comentarios indicando que está situado en una etiqueta de la página web ('tag') que tiene un atributo ('attr') cuyo valor es el señalado en 'valueAttr', y finalmente indicamos si el valor a almacenar es el contenido de la etiqueta de la página web que se ha señalado en el 'tag' (en ese caso marcamos 'tagID' como false) o si está en un atributo de dicha etiqueta (en ese caso marcamos en que atributo está, como es el caso del 'rating' en el ejemplo de la imagen 19). El 'rowName' debe coincidir con el indicado en la definición de la base de datos para los comentarios.

6.2.2. Obtener la información de la página web objetivo

A continuación se explica como obtener la información necesaria de la página web sobre la que vayamos a utilizar el programa. Para ello se debe utilizar una herramienta de desarrollo para el navegador, que puede ser las que traen por defecto cada navegador o plugins para este propósito como, por ejemplo, 'Firebug'. Una vez que tengamos abierta esta herramienta comenzaremos a obtener cada uno de los datos que necesitamos para llenar el archivo de configuración. Los ejemplos que se mostrarán a continuación son principalmente obtenidos de la web de Marca y de PCComponente. En la web de Marca los productos serían las noticias, mientras que en la de PCComponente serían los productos a la venta en la web. Comencemos:

- **Codificación:** La codificación suele estar en la cabecera (etiqueta <head>) de la página web principal, y las codificaciones más comunes son utf-8 e iso-8859-15. En las imágenes 20 y 21 tenemos un par de ejemplos:

```

<!DOCTYPE html>
<html class="webFont" lang="es">
  <head>
    <title>PcComponentes.com | Tienda de Informática y Tecnología</title>
    <meta charset="utf-8">
    <script src="https://js-agent.newrelic.com/nr-1039.min.js"></script>
    <script src="https://www.salesmanago.pl/dynamic/xf7kphig7t5r69m7/custom.j_.Inform%C3%Altica%20y%20Tecnolog%C3%ADa%20online&banners=true" asvnc="" tvoe="text/javascript"></script>
  
```

Figura 20: Codificación

```

<!DOCTYPE html>
<html lang="es">
  <head>
    <link href="http://e00-marca.uecdn.es/assets/v7/css/core-marca-desktop.css" type="text/css" rel="stylesheet">
    <meta content="text/html; charset=iso-8859-15" http-equiv="Content-Type">
    <meta content="IE=edge;chrome=1" http-equiv="X-UA-Compatible">
    <title data-ue-ue="title" data-ue-c="innerHTML">MARCA - Diario online líder en información deportiva</title>
    <script type="text/javascript" asvnc="" src="https://active.cache.el-mundo.net/hannerc/cvence/cv_ie">
  
```

Figura 21: Codificación

- URL de los productos:** Los productos que queramos obtener pueden estar representados en la página web de forma muy diferente dependiendo del lenguaje utilizado y de los programadores. Por ejemplo, en el caso de Marca los productos serían las noticias que deseamos almacenar y estarían representados en la web donde se muestran dichas noticias:



Figura 22: Enlace a productos

En cambio en otras webs, como es el caso de PCComponentes, al utilizar javascript el enlace donde se muestran todos los productos, los cuales están divididos por páginas, no resulta tan evidente, puesto que utiliza una API a la cual llama cada vez que cambiamos de página u ordenamos los productos. Para comprobar si la web objetivo utiliza esto simplemente, con la consola abierta, procedemos a cambiar de página o a ordenar la página actual:



Figura 23: Enlace a productos

Una vez ordenado o cambiado de página, en la consola debería aparecer un nuevo enlace que es la llamada a la API que la web realiza para obtener los productos que va a mostrar:



Figura 24: Enlace a productos

Procedemos a copiar esa URL y pegarla en el navegador, eliminando de ella los parámetros pasados por GET que no nos sirvan para nuestro propósito y comprobando en qué parámetro se envía el cambio de página y en qué parámetro se envía el cambio de categoría para indicarlo como variables en nuestro archivo de configuración:



QNovedad

[Gigabyte GA-H110-D3A](#)

129 €
Recíbelo el **martes 11 de julio**
Añadir al carrito



Figura 25: Enlace a productos

En el ejemplo mostrado en la imagen 25, es en el parámetro 'page' donde se indica la página y en el parámetro 'idFamilies[]' donde se indica la categoría.

- **URL de los comentarios:** La idea es igual que la anterior. Primero, con la consola abierta, buscamos donde están los comentarios:

```

Consola ▾ HTML CSS Script DOM Red Cookies
Limpiar Mantener Analizar Todo Errores Advertencias Información Información de depuración Cookies
/gigabyte-ga-h110m-s2h
GET https://eul-search.doofinder.com/5/options/d9c752526286837ecd93ee20ff18249e?www.pccomponentes.com 200 C
GET https://www.pccomponentes.com/essentials/91283 200 OK 136ms
GET https://www.pccomponentes.com/ajax_nc/availability/91283 200 OK 615ms
GET https://www.pccomponentes.com/va/complements/91283 200 OK 87ms
POST https://www.pccomponentes.com/comethunter 200 OK 119ms
popups.js (línea 24)
doofind....min.js (línea 1)
jquery....2.1.js (línea 1)
    
```

Figura 26: Enlace a comentarios

Luego comprobamos si al entrar en los comentarios se ha realizado alguna llamada a la API que debe estar reflejada en la consola:

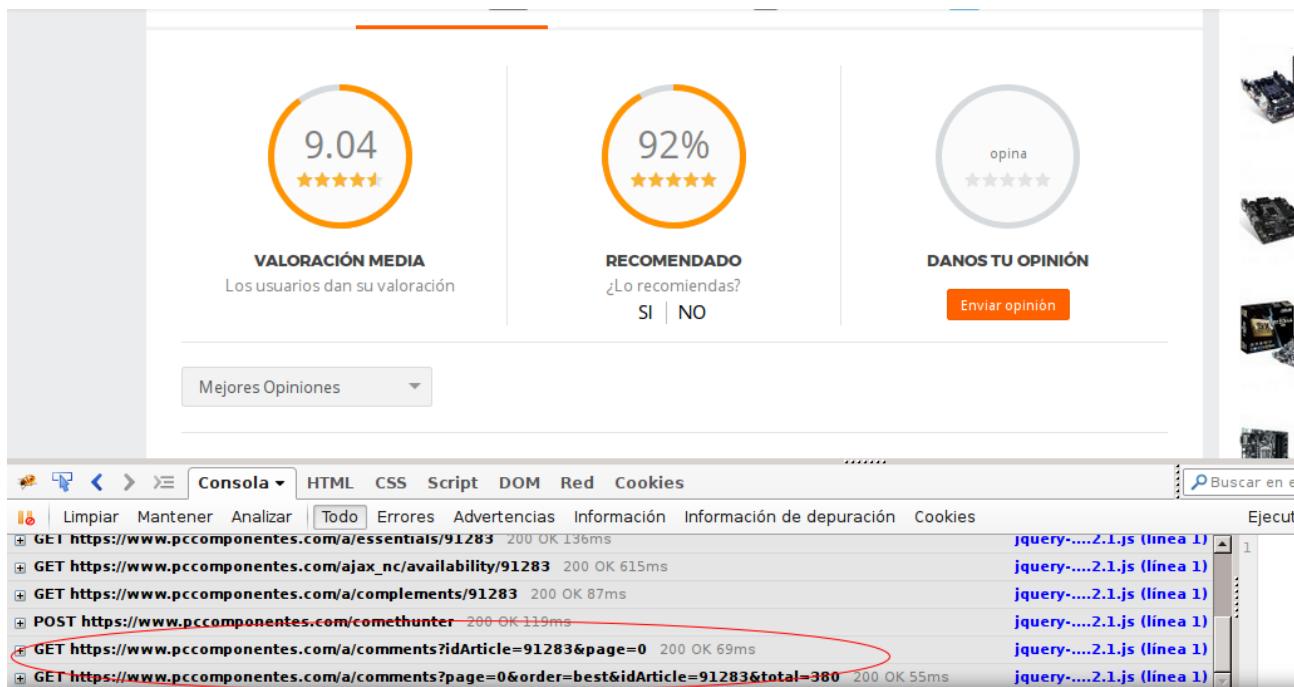


Figura 27: Enlace a comentarios

En caso de que al mostrar los comentarios no aparezca llamada a la API, pues hacemos alguna acción que recarguen los comentarios mostrados como puede ser cambiar de página o reordenar los comentario, para de esta forma obligar a recargar los comentarios llamando a la API:



Figura 28: Enlace a comentarios

De este modo ya tendremos el enlace a la web donde se obtienen los comentarios y solo tendremos que buscar los parámetros que determinan la página y el producto relacionado a esos comentarios para utilizar nuestras variables en el archivo de configuración.

También puede ocurrir que el enlace a los comentarios no esté en la misma vista, sino que sea otra página, como por ejemplo en el caso de la web Mubis, donde el enlace a un producto de la web es <http://www.mubis.es/titulos/origen-inception-blu-ray/> y el de los comentarios de ese producto es <http://www.mubis.es/titulos/origen-inception-blu-ray/opiniones?page=1>.

- **URL de cada producto:** En el enlace obtenido anteriormente donde se muestran todos los productos debemos encontrar donde está situado el enlace a cada uno de esos productos.



Figura 29: Enlace a producto

Una vez encontrado el enlace, inspeccionamos para ver donde está la URL de ese producto en el código HTML:

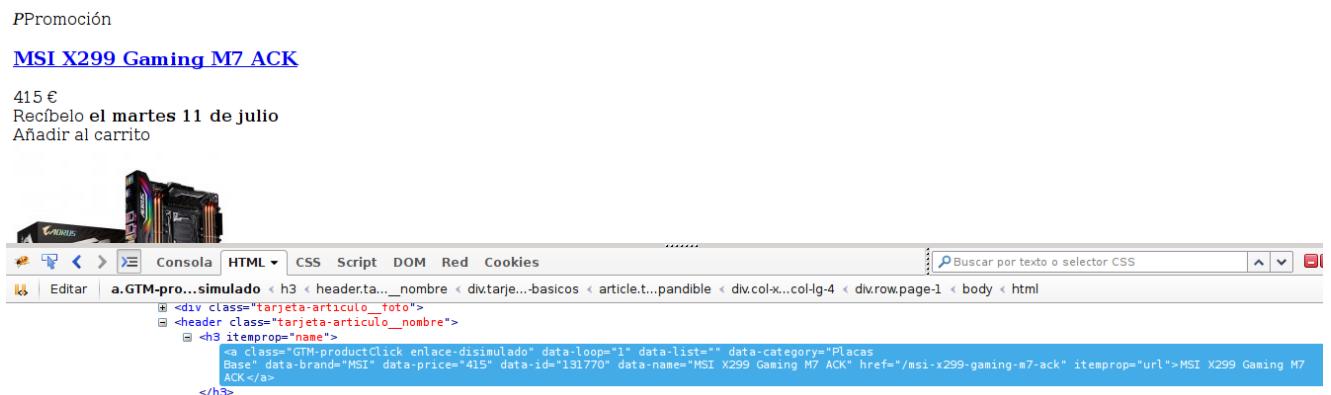


Figura 30: Enlace a producto

En el ejemplo de la imagen 30 vemos que el enlace al producto está situado dentro de la etiqueta 'header' cuyo atributo 'class' es 'tarjeta-articulo_nombre', y dentro de esta en la etiqueta 'h3' cuyo atributo 'itemprop' es 'name'. Por lo que esta información se introduce en el 'linkProduct' de nuestro archivo de configuración y terminaríamos indicando la etiqueta donde está lo que buscamos, esto sería la etiqueta 'a' que tiene un atributo 'class' que es igual a 'GTM-productClick enlace-disimulado', y el valor que queremos está situado en el atributo 'href' de dicha etiqueta, por lo que esto lo ponemos en la etiqueta 'linkProductFinal' de nuestro archivo de configuración utilizando el atributo 'attrGoal' para indicarle que lo que buscamos está situado en 'href'.

- **Atributos de productos:** Dentro de la URL de cada producto debemos obtener todas

las características que deseemos almacenar. Para ello, buscamos dentro del código HTML de dicha página web en qué etiquetas se posiciona el valor a almacenar:

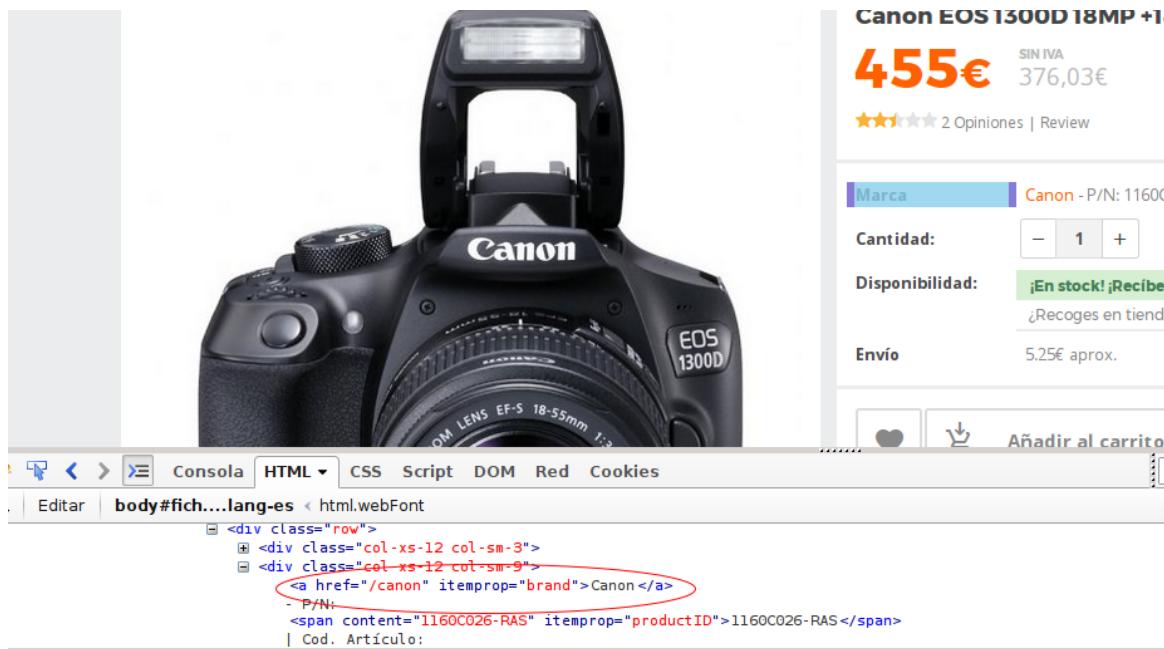


Figura 31: Atributos de productos

Habiendo localizado esta información procedemos a rellenar en el archivo de configuración el lugar donde se sitúa dicho valor, indicándolo entre las etiquetas <attributeProducts>:

```
<!-- Marca -->
<attributeProducts tag="a" attr="itemprop" valueAttr="brand" tagID="false" rowName="brand">
</attributeProducts>
```

Figura 32: Atributos de productos

Aquí indicamos que en la etiqueta 'a' que tiene un atributo llamado 'itemprop' cuyo valor es 'brand' queremos obtener dicho valor y almacenarlo en el atributo "brand" de nuestra base de datos. Esto mismo se haría con cada uno de los atributos relacionados con el producto que se vaya a almacenar. Los atributos 'attr' y 'valueAttr' del archivo de configuración se pueden dejar vacíos en caso de ser necesario, pero podría perjudicar la precisión de la aplicación haciendo que al obtener el dato este sea confundido con otra información de la web.

- **Identificación del producto:** Para entrar en los comentarios relacionados con cada producto necesitaremos la ID del producto. Para indicarla en el archivo de configuración utilizaremos las etiquetas <idProduct> finalizándolas con <idProductFinal> de la forma que se explicó en el apartado correspondiente de la sección anterior. Cuando obtengamos la URL de los comentarios relacionados con el producto, en dicha URL se enviará la ID del producto. Podemos utilizar esta ID para buscarla en el código HTML de la página web del producto y localizar en qué posición está esta información:

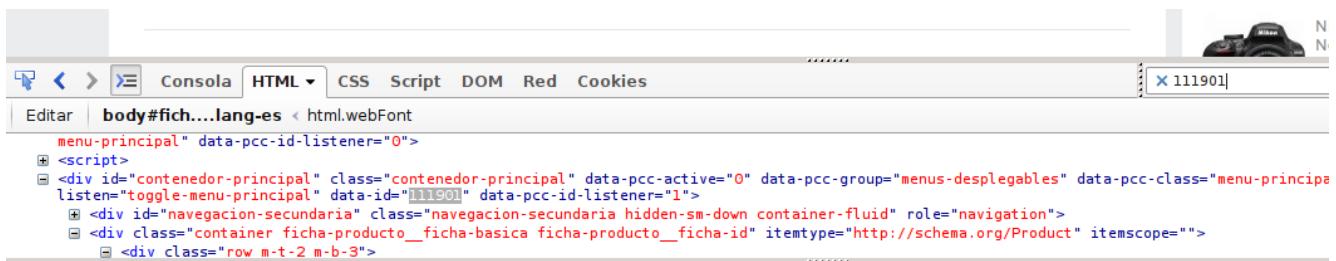


Figura 33: Identificación del producto

Una vez localizado el ID del producto solo quedará indicarle como llegar hasta ese dato en el archivo de configuración:

```

<!-- Indicar como llegar hasta la ID necesaria para entrar en los comentarios (el
numberproduct) -->
<idProduct tag="body" attr="id" valueAttr="ficha-producto"></idProduct>
<idProductFinal tag="div" attr="id" valueAttr="contenedor-principal" attrGoal="data-id">
<idProductFinal>

```

Figura 34: Identificación del producto

También puede ocurrir que la identificación no sea numérica, como es el caso de la web <http://www.mubis.es>, donde el enlace a los comentarios de un producto sería el siguiente: <http://www.mubis.es/titulos/origen-inception-blu-ray/opiniones?page=1>, por lo que lo que identifica al producto en el enlace a sus comentarios es 'origen-inception-blu-ray', y, por lo tanto, habría que buscar la etiqueta de la página web del producto en donde está esta información.

- Atributo tagID del archivo de configuración:** Dentro del código HTML los valores que queremos almacenar pueden estar mostrados en la web o ser parte del atributo de una etiqueta HTML. Para diferenciar estas dos opciones tenemos en el archivo de configuración el atributo 'tagID' para las etiquetas <attributeProducts> y <attributeComments>. Dicho atributo será "false" cuando el dato que queremos obtener esté en la web, pero hay algunos casos en los el valor está en un atributo, como en el siguiente ejemplo:

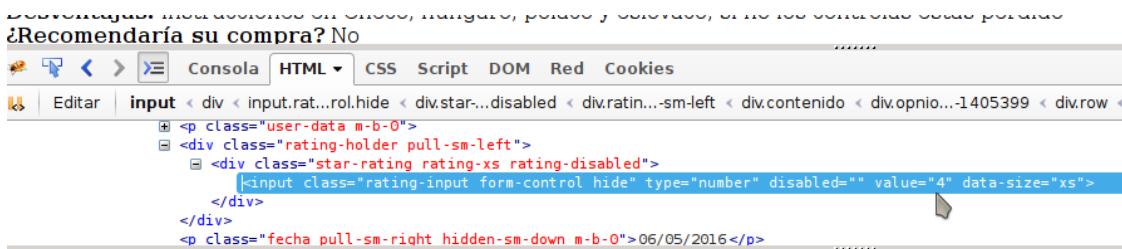


Figura 35: Atributo tagID

En la imagen 35 queremos almacenar el valor del 'rating' con respecto al comentario. Dicho valor está situado en el atributo 'value' de la etiqueta <input>. Por lo tanto, indicaremos esto en nuestro archivo de configuración:



```
<urlComments>
  <attributeComments tag="p" attr="class" valueAttr="user-data m-b-0" tagID="false" rowName="nick"></attributeComments>
  <attributeComments tag="div" attr="class" valueAttr="opinion" tagID="false" rowName="comment"></attributeComments>
  <attributeComments tag="input" attr="class" valueAttr="rating-input" tagID="value" rowName="rating"></attributeComments>
  <attributeComments tag="p" attr="class" valueAttr="fecha pull-sm-right hidden-sm-down m-b-0" tagID="false" rowName="date"></attributeComments>
</urlComments>
```

Figura 36: Atributo tagID

En el tercer atributo que almacenamos en la imagen 36 podemos observar que indicamos que dicho atributo está en la etiqueta 'input' que tiene un atributo 'class' cuyo valor es 'rating-input', y utilizamos el 'tagID' para indicar que queremos almacenar el valor almacenado en el atributo 'value' de dicha etiqueta. Finalmente indicamos que esta información la almacenaremos en nuestra base de datos en la columna con el nombre 'rating'.

6.2.3. Aplicación web

En esta sección se explican todas las pantallas y opciones disponibles en la aplicación web.

6.2.3.1 Login, registro y recuperación de la contraseña

La pantalla de Login es sencilla y completamente intuitiva. Introduciendo el nombre de usuario y la contraseña correctamente se entra en la aplicación.

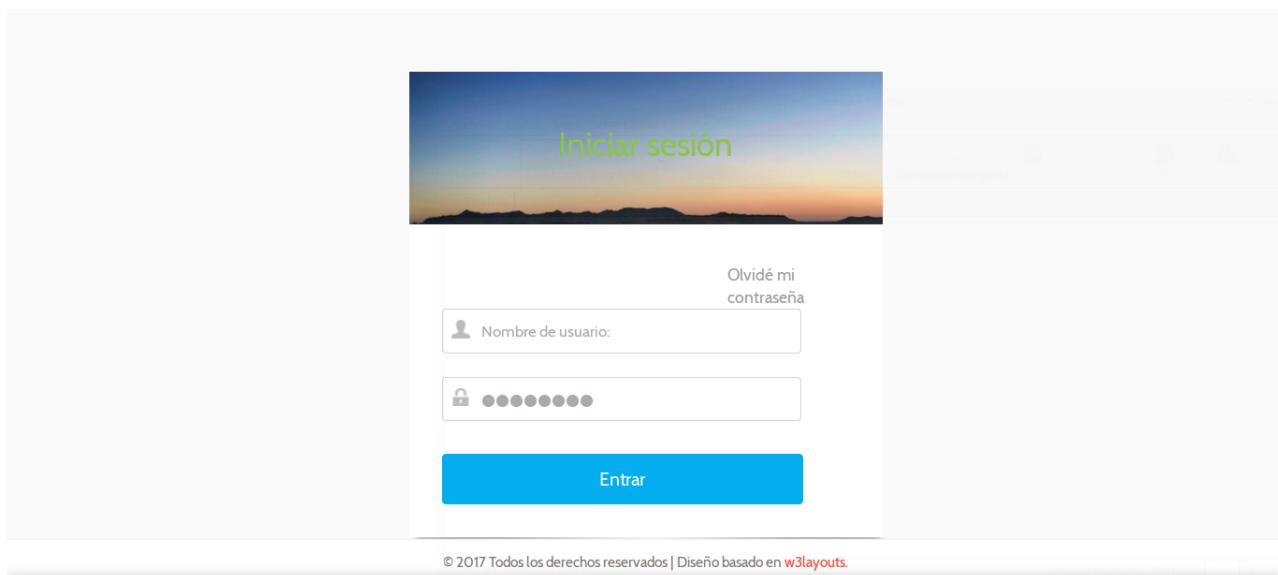


Figura 37: Pantalla del Login

Además, desde esta pantalla tenemos acceso a la opción de recuperar la contraseña y al registro. Para recuperar la contraseña se necesita introducir el nombre de usuario y el correo electrónico del registro, posteriormente se enviará una nueva contraseña a dicho correo electrónico con la que se podrá acceder a la aplicación.

The screenshot shows a web page titled "Recuperación de la contraseña". Below the title, a note says "Recibirás un correo electrónico con la nueva contraseña." A red header "Datos del usuario" is followed by two input fields: "Nombre de usuario*" and "Correo electrónico*". At the bottom left is a red button labeled "Completar", and at the bottom right is a link "Volver al Inicio". The footer contains the copyright notice "© 2017 Todos los derechos reservados | Diseño basado en w3layouts."

Figura 38: Pantalla de recuperación de contraseña



Figura 39: Email de recuperación de contraseña

En el registro se introducen los datos del cliente y el sistema enviará un correo electrónico automáticamente confirmando el registro. El administrador de la aplicación deberá aprobar dicho registro e informará al cliente cuando todo esté listo para su utilización.

Formulario de registro

Regístrate para solicitar la aplicación y comenzar a disfrutar de sus estadísticas e informes cuanto antes.

Datos del usuario

Nombre de usuario*:

Correo electrónico*:

Contraseña*:

Sobre la página web

Nombre de la web*:

© 2017 Todos los derechos reservados | Diseño basado en [w3layouts](#).

Figura 40: Pantalla de registro inicial



Figura 41: Email de registro

El diseño de la web es 'responsive' y se adapta a dispositivos móviles como se muestra en la pantalla de login de la imagen siguiente, realizada en un smartphone, por lo que la conexión a la aplicación web por parte de los clientes puede ser realizada desde cualquier dispositivo que cuente con un navegador web y conexión a Internet.

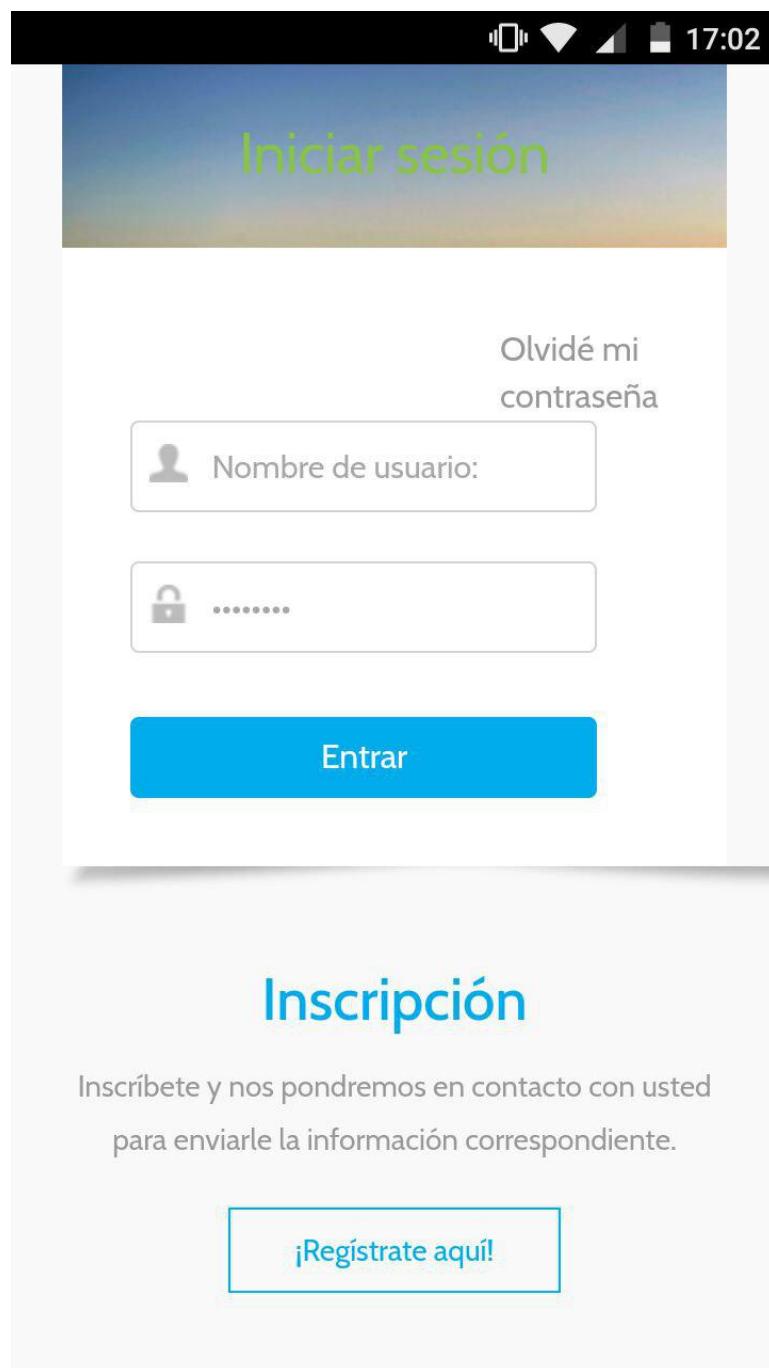


Figura 42: Pantalla de Login en dispositivo móvil

6.2.3.2 Pantalla de configuración. Información y cambio de contraseña

La pantalla de configuración de usuario es común a todos los roles de usuarios que componen la aplicación. En ella se muestra la información relacionada al usuario logueado y también se da la posibilidad a modificar la contraseña actual.

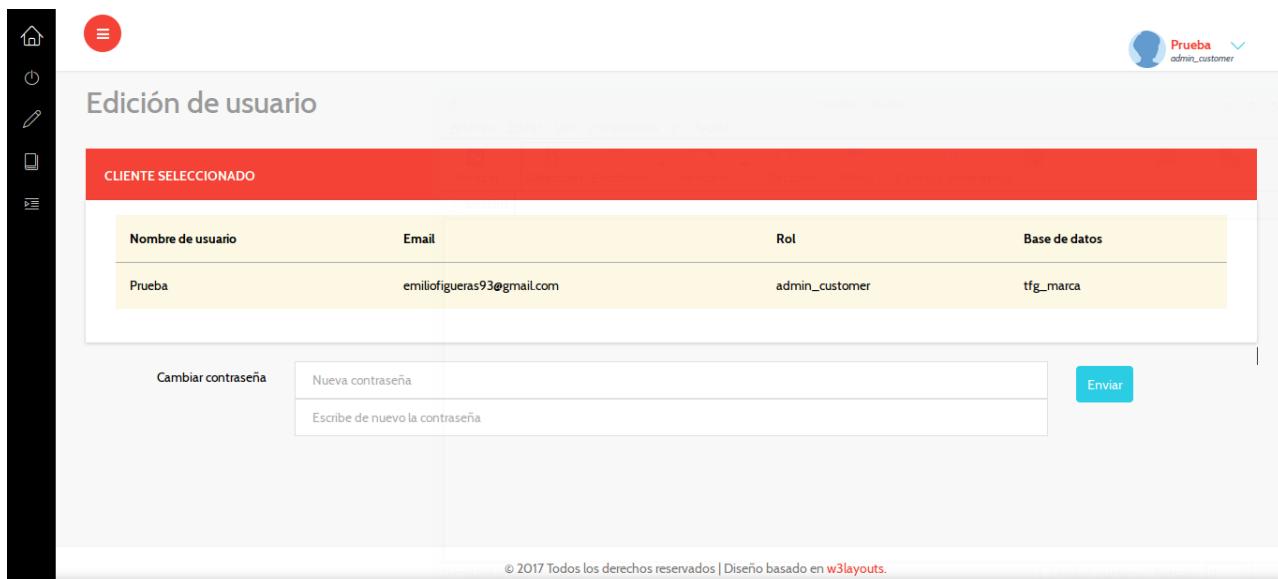


Figura 43: Pantalla de configuración de usuario

6.2.3.3 Pantalla inicial para el rol 'Administrador'

Al iniciar sesión con un usuario con rol de 'Administrador' se puede ver la pantalla con los clientes que se han registrado pero están inactivos, es decir, están a la espera de confirmación.

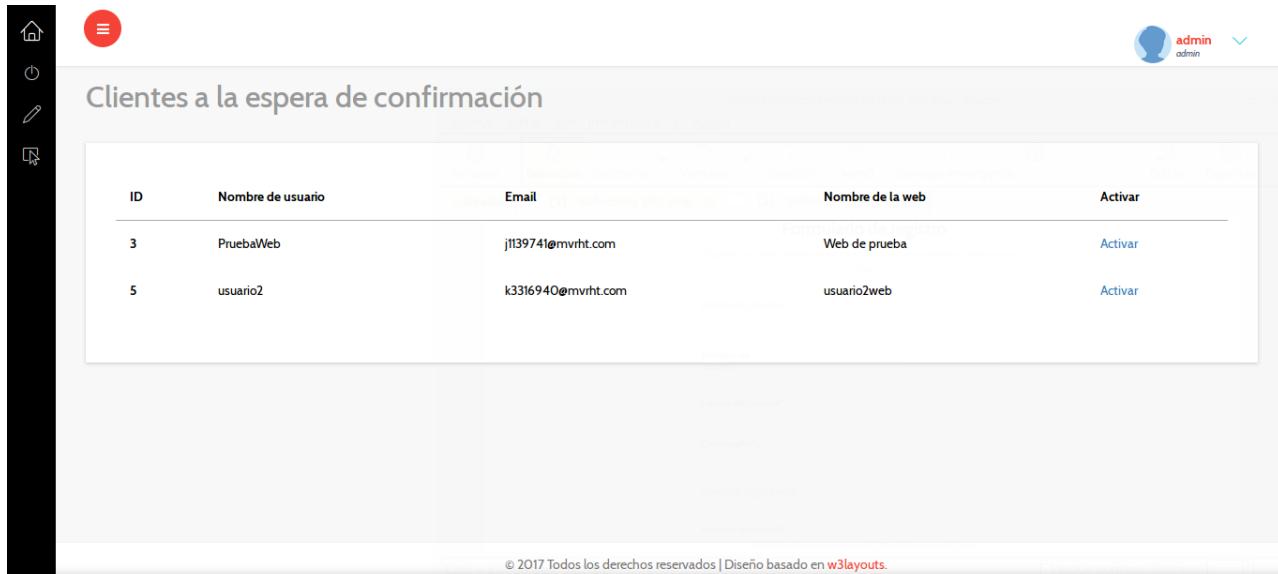


Figura 44: Pantalla de inicio del Administrador

6.2.3.4 Registrar administrador

En la barra lateral del rol 'Administrador' contamos con dos opciones más, una de ellas es registrar administrador. Introduciendo un nombre de usuario, un correo electrónico y una contraseña se creará y activará un nuevo usuario con el rol de 'Administrador'.

Formulario de registro

Crear un nuevo administrador global de la aplicación web.

Datos del usuario

Nombre de usuario*:

Correo electrónico*:

Contraseña*:

© 2017 Todos los derechos reservados | Diseño basado en w3layouts.

Figura 45: Pantalla de registro de un nuevo administrador

6.2.3.5 Visualización de usuarios activos

La otra opción de la barra lateral para el rol de 'Administrador' es 'Ver usuarios'. En esta pantalla aparecen todos los usuarios activos de todos los clientes.

Usuarios activos

ID	Nombre de usuario	Email	Nombre de la web	Rol	Editar
1	EmilioFigueroa	emilio@emilio.com		admin	Editar
2	Prueba	emiliofigueras93@gmail.com	Cliente de prueba	admin_customer	Editar
4	nuevoUsuario	k3316940@mvrht.com	nuevoCliente	admin	Editar
6	admin	admin		admin	Editar
7	torotoro	j1559222@mvrht.com	la web de toro	admin_customer	Editar
8	admin2	admin@admin.com		admin	Editar
13	marca_usuario	ninguno@ninguno.com	Cliente de prueba	user_customer	Editar

© 2017 Todos los derechos reservados | Diseño basado en w3layouts.

Figura 46: Pantalla de visualización de usuarios activos

Desde la pantalla de 'Ver usuarios' se puede ver la información de todos los usuarios activos así como acceder a la pantalla de edición del usuario, en la que podremos modificar la base de datos asignada al usuario, modificar el rol, cambiarle la contraseña o desactivarlo.

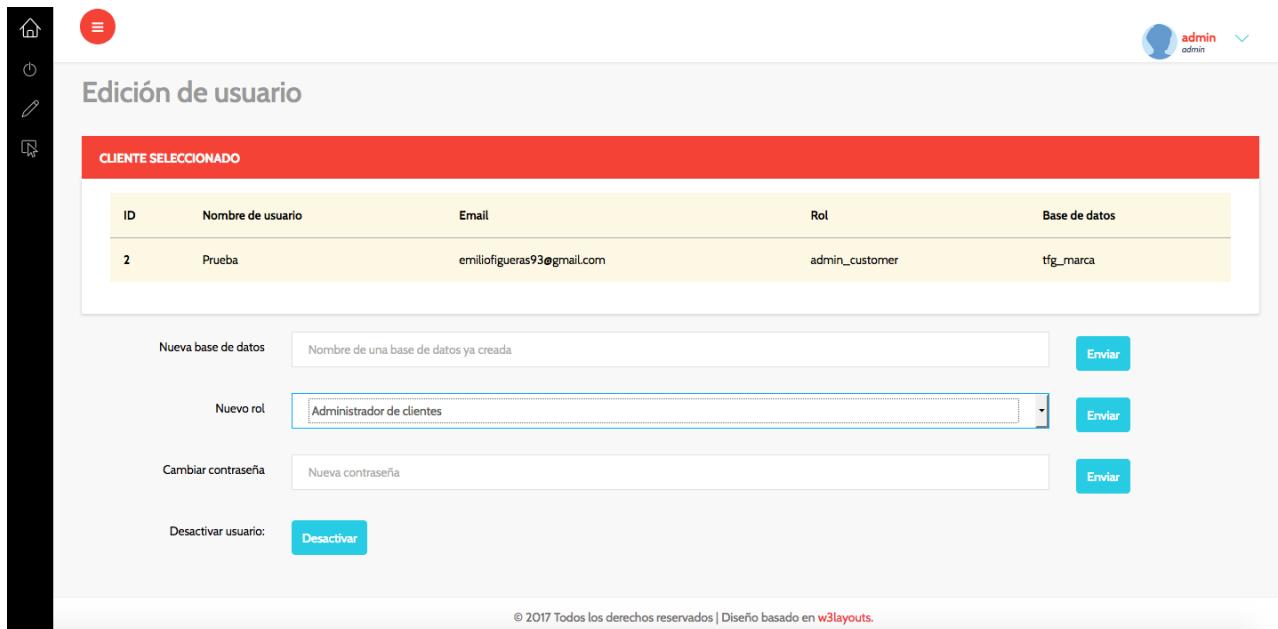


Figura 47: Pantalla de edición de usuarios desde la administración

6.2.3.6 Pantalla inicial para el rol 'Cliente Administrador'

Al iniciar sesión con un usuario que tenga como rol 'Cliente Administrador' se muestran los usuarios activos del cliente al que pertenezca el usuario logueado. En la imagen siguiente, realizada en un dispositivo móvil, se puede ver la versión 'responsive' de la aplicación web.

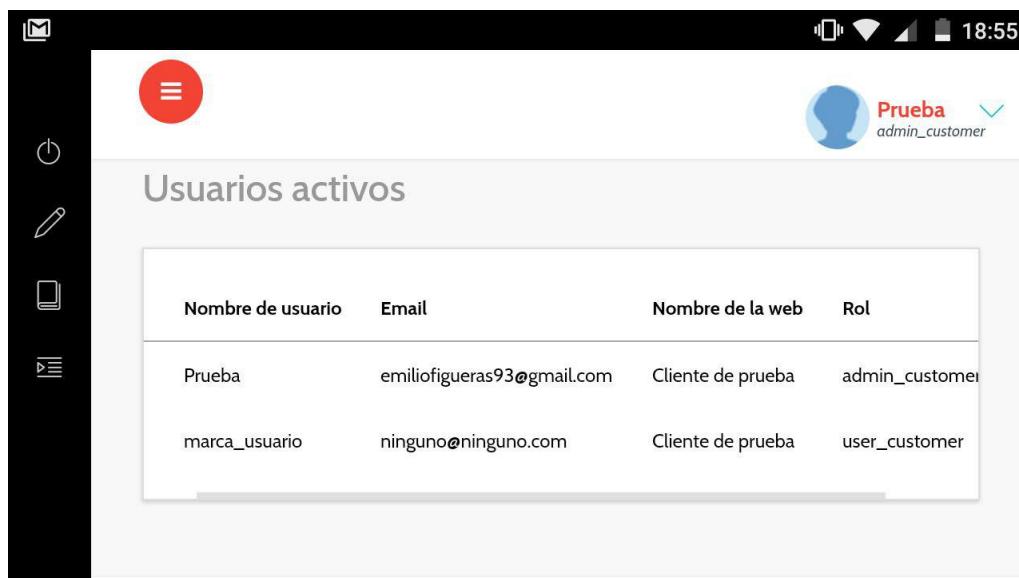


Figura 48: Pantalla de inicio para el rol 'Cliente Administrador'

Desde esta vista se puede acceder a la edición de los usuarios, donde se podrá eliminar el usuario, modificar el rol y modificar la contraseña.

The screenshot shows a user interface for editing a selected client. At the top right, there is a user profile icon labeled "Prueba" and "admin_customer". On the left, a vertical sidebar contains icons for home, refresh, edit, and list. The main title is "Edición de usuario". Below it, a red header bar says "CLIENTE SELECCIONADO". A table displays the selected client's details:

ID	Nombre de usuario	Email	Rol	Base de datos
2	Prueba	emiliofigueras93@gmail.com	admin_customer	tfg_marca

Below the table, there are three input fields with buttons:

- "Nuevo rol": "Administrador de clientes" with a dropdown arrow and a "Enviar" button.
- "Cambiar contraseña": "Nueva contraseña" with a "Enviar" button.
- "Eliminar usuario": "Eliminar" button.

At the bottom, a footer note reads: "© 2017 Todos los derechos reservados | Diseño basado en w3layouts."

Figura 49: Pantalla de edición de usuario para 'Cliente Administrador'

6.2.3.7 Registrar usuario desde el rol 'Cliente Administrador'

La primera opción de la barra lateral del rol 'Cliente Administrador' es la opción de registrar un nuevo usuario relacionado con el cliente que lo crea. La imagen que se muestra es de la visualización del formulario para la creación de un nuevo usuario desde un dispositivo móvil.

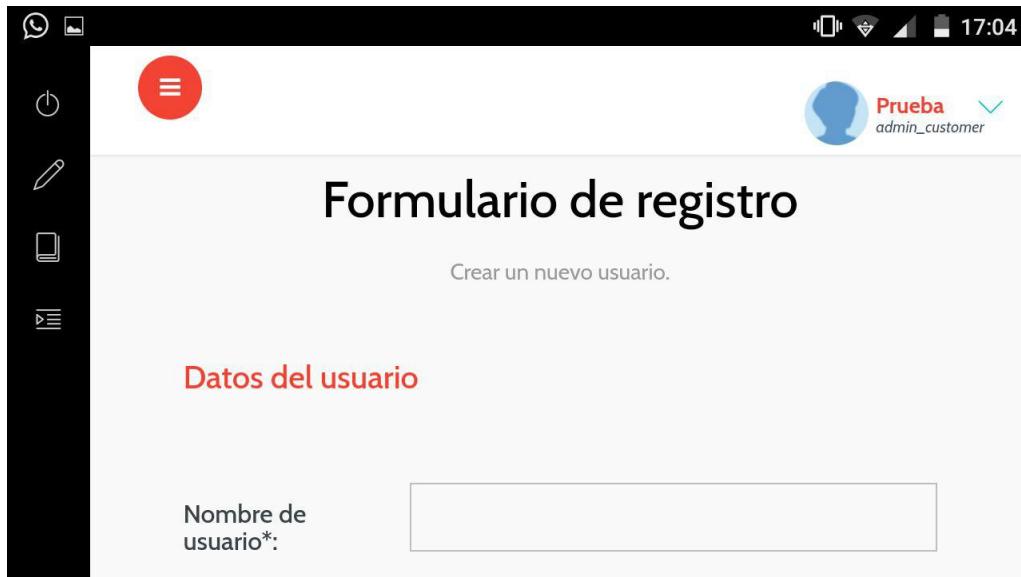


Figura 50: Pantalla de registro de usuario para 'Cliente Administrador'

6.2.3.8 Comentarios

La siguiente opción para el rol 'Cliente Administrador' son los comentarios. En esta pantalla se tiene la posibilidad de valorar los comentarios de los diferentes productos almacenados en la base de datos, así como la opción de descargar los comentarios aplicando diferentes filtros, ya sea a través de una lista de palabras prohibidas como a través de consultas según alguno de los atributos de productos o comentarios almacenados en la base de datos.

The screenshot shows a web-based application interface for managing comments. At the top, there's a navigation bar with icons for home, search, and other functions. On the right, a user profile is shown with the name "Prueba" and the role "admin_customer". Below the header, the main content area has a title "Valorar comentarios" (Rate comments). There are two radio buttons: one for "Comentarios no valorados" (Unrated comments) and another for "Comentarios ya valorados" (Rated comments), with the latter being selected. A large blue button labeled "Valorar" (Rate) is prominently displayed. Below this, a section titled "Descargar comentarios" (Download comments) is shown, featuring a list of comments with checkboxes for rating. The comments listed are:

- 34: "¿Por qué el Getafe usa 'trending topic' y no 'tendencia' o 'tema de momento'? Claro, en inglés queda más 'cool' o 'fashion' que en el clásico castellano. No te j... Denunciar Valorar: Positivo Negativo Responder" with "Si" and "No" radio buttons.
- 35: "CrPenaldo #9 02.nov.2016 | 09:15 Y por que no nombran al Español y al Getafe? Los 2 actuales filiales del Madrid Denunciar Valorar: Positivo Negativo Responder" with "Si" and "No" radio buttons.
- 36: "Cdlugosiempre #130.oct.2016 | 19:08 No es mérito del...digo tú el gol.fue al gran error garrafal de nuestra defensa. Denunciar Valorar: Positivo Negativo Responder" with "Si" and "No" radio buttons.
- 37: "SPAINX #2 30.oct.2016 | 19:08 Y pensar q no le querian cuando llegó a Cádiz.Guiña ganó una Eurocopa.no lo olvidemos.y para golazo suyo tambien.el del 4-0 del Getafe al Barcelona en aquella eliminatoria q hizo inútil aquel golazo de Messi Denunciar Valorar: Positivo Negativo Responder" with "Si" and "No" radio buttons.
- 38: "Diferencia de potencial #3 30.oct.2016 | 19:09 Y? En serio? Cualquiera diría... Denunciar Valorar: Positivo Negativo Responder" with "Si" and "No" radio buttons.
- 39: "C4dista71 #4 30.oct.2016 | 19:23 Genio!!! Denunciar Valorar: Positivo Negativo Responder" with "Si" and "No" radio buttons.
- 40: "Ganja-SanPedro #5 30.oct.2016 | 19:34 Si esta bien pero que poco exigente el trending topin ese Denunciar Valorar: Positivo Negativo Responder" with "Si" and "No" radio buttons.

At the bottom of the page, there's a footer note: "© 2017 Todos los derechos reservados | Diseño basado en w3layouts."

Figura 51: Pantalla de comentarios para 'Cliente Administrador'

6.2.3.9 Valoración de comentarios

Al seleccionar la opción de valorar comentarios desde la pantalla anterior nos lleva a un formulario donde se van mostrando los comentarios almacenados de diez en diez, y se debe dar una valoración binaria, entre positivo o negativo, a cada uno de ellos.

This screenshot shows the same "Valorar comentarios" (Rate comments) page as Figure 51, but it displays a different set of comments from the list. The comments shown are:

- 34: "¿Por qué el Getafe usa 'trending topic' y no 'tendencia' o 'tema de momento'? Claro, en inglés queda más 'cool' o 'fashion' que en el clásico castellano. No te j... Denunciar Valorar: Positivo Negativo Responder" with "Si" and "No" radio buttons.
- 35: "CrPenaldo #9 02.nov.2016 | 09:15 Y por que no nombran al Español y al Getafe? Los 2 actuales filiales del Madrid Denunciar Valorar: Positivo Negativo Responder" with "Si" and "No" radio buttons.
- 36: "Cdlugosiempre #130.oct.2016 | 19:08 No es mérito del...digo tú el gol.fue al gran error garrafal de nuestra defensa. Denunciar Valorar: Positivo Negativo Responder" with "Si" and "No" radio buttons.
- 37: "SPAINX #2 30.oct.2016 | 19:08 Y pensar q no le querian cuando llegó a Cádiz.Guiña ganó una Eurocopa.no lo olvidemos.y para golazo suyo tambien.el del 4-0 del Getafe al Barcelona en aquella eliminatoria q hizo inútil aquel golazo de Messi Denunciar Valorar: Positivo Negativo Responder" with "Si" and "No" radio buttons.
- 38: "Diferencia de potencial #3 30.oct.2016 | 19:09 Y? En serio? Cualquiera diría... Denunciar Valorar: Positivo Negativo Responder" with "Si" and "No" radio buttons.
- 39: "C4dista71 #4 30.oct.2016 | 19:23 Genio!!! Denunciar Valorar: Positivo Negativo Responder" with "Si" and "No" radio buttons.
- 40: "Ganja-SanPedro #5 30.oct.2016 | 19:34 Si esta bien pero que poco exigente el trending topin ese Denunciar Valorar: Positivo Negativo Responder" with "Si" and "No" radio buttons.

A blue "Enviar" (Send) button is located at the bottom right of the comment list. The footer note "© 2017 Todos los derechos reservados | Diseño basado en w3layouts." is also present.

Figura 52: Pantalla de valoración de comentarios para 'Cliente Administrador'

6.2.3.10 Informes en el rol 'Cliente Administrador'

La última opción de la barra lateral para el rol 'Cliente Administrador' es la de los informes. En esta pantalla se pueden ver los detalles de todos los informes creados. Desde aquí se puede acceder a la visualización, edición o eliminación dichos informes, así como crear nuevos.

ID	Título	Descripción	Tipo	Activo	Editar	Ver	Eliminar
1	Número de comentarios por noticia	Muestra el número de comentarios que tienen las noticias que más comentarios tienen.	bar	1	Editar	Ver	Eliminar
2	Comentarios favorables vs desfavorables	Una comparativa entre los comentarios que son favorables contra los que son desfavorables según el análisis propio.	pie	1	Editar	Ver	Eliminar
4	Bien vs Mal	Comparativa entre las apariciones de la palabra 'buen' vs la palabra 'mal'	pie	1	Editar	Ver	Eliminar

Figura 53: Pantalla de lista de informes para 'Cliente Administrador'

6.2.3.11 Visualización de informes

Al pulsar sobre la opción de visualización de un informe se muestra una pantalla donde la consulta de dicho informe se ejecuta y se visualiza en forma de gráfica, los datos de la gráfica aparecen sobre esta. En la parte inferior podemos observar la leyenda de la gráfica seleccionada.

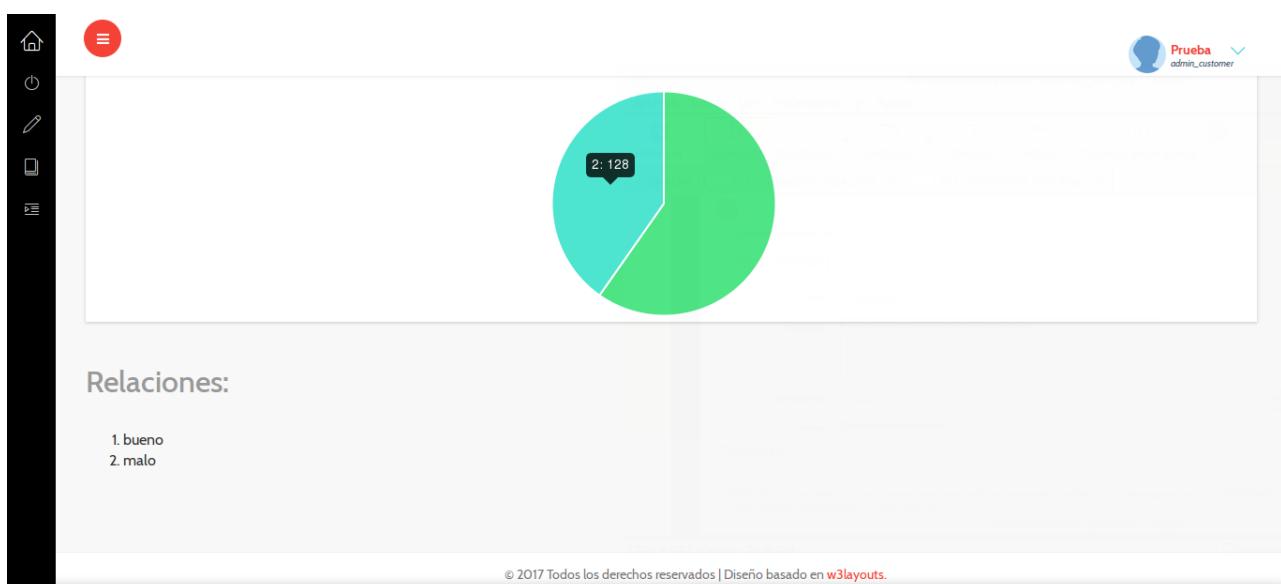


Figura 54: Pantalla de visualización de informe

6.2.3.12 Creación y edición de informes

Al crear un nuevo informe o editar uno ya existente se nos muestra una pantalla compuesta por varios apartados, estos son:

- Información: Breve descripción del proceso de creación de un informe. Se explica las tablas y atributos disponibles para la realización de la consulta que genere dicho informe.
- Crear informe: Se introducen los detalles del informe, es decir, el nombre, la descripción, el tipo de gráfico y si estará activo de entrada.
- Consulta: Se explica las reglas que hay que seguir para construir la consulta adecuadamente. Se introduce, en cada campo, la parte específica de la consulta, es decir, el 'select' obligatoriamente y, en caso de necesitarlos, los 'where', 'group by' y 'order by'.

The screenshot shows a web-based reporting application. On the left, there's a vertical sidebar with icons for home, refresh, edit, and search. The main area has a title 'Crear informe:' and several input fields:

- Título:** A text input field with placeholder 'Título del informe...'.
- Descripción:** A text area with placeholder 'Escriba aquí una descripción de la consulta realizada...'.
- Tipo de gráfico:** A dropdown menu currently set to 'Barra'.
- Activar:** A checkbox labeled 'Activar/Desactivar visualización'.

Below these fields, there's a section titled 'Consulta' containing a note in a green box:

IMPORTANTE: Las tres tablas de las que se pueden sacar datos (productos, comentarios y análisis) ya han sido agregadas, por ello ES IMPORTANTE USAR LOS ALIAS asignados a la hora de trabajar con los atributos en el select, estos son:

At the very bottom of the page, there's a footer with the text '© 2017 Todos los derechos reservados | Diseño basado en w3layouts.'

Figura 55: Pantalla de la creación/edición de informes

6.2.3.13 Pantalla inicial para el rol 'Cliente Usuario'

Al iniciar sesión con un usuario con rol 'Cliente Usuario' se cargará la pantalla de la visualización de los informes. En esta pantalla se describen todos los informes activos para el usuario logueado. Desde aquí se da la posibilidad de ver los informes, como ya se ha mostrado anteriormente y se puede ver en la figura 54, así como también se puede reportar los informes en caso de encontrar algún tipo de error.

The screenshot shows the 'WID Panel' homepage. On the left, there is a sidebar with a green header titled 'WID Panel'. It contains two items: 'Ver informes' (with a document icon) and 'Descargar comentarios' (with a download icon). At the top right, there is a user profile icon labeled 'mar' and 'User'. The main content area has a title 'Informes' and a subtitle 'Se presentan todos los informes en orden alfabético'. Below this is a table listing three reports:

ID	Título	Descripción	Tipo	Ver	Notificar
4	Bien vs Mal	Comparativa entre las apariciones de la palabra 'buen' vs la palabra 'mal'	pie	Ver	Reportar error
2	Comentarios favorables vs desfavorables	Una comparativa entre los comentarios que son favorables contra los que son desfavorables según el análisis propio.	pie	Ver	Reportar error
1	Número de comentarios por noticia	Muestra el número de comentarios que tienen las noticias que más comentarios tienen.	bar	Ver	Reportar error

At the bottom of the page, there is a footer note: '© 2017 Todos los derechos reservados | Diseño basado en w3layouts.'

Figura 56: Pantalla de inicio para el rol 'Cliente Usuario'

6.2.3.14 Reportar informe

Desde aquí se puede enviar directamente un correo electrónico al administrador del usuario. En este correo electrónico se indica el informe que contiene el error y, desde esta pantalla, se puede agregar, si el usuario lo desea, una descripción de dicho error que irá incluida en el correo que recibirá el administrador.

INFORMACIÓN: Se va a enviar un correo electrónico a su administrador, indicando la información del informe seleccionado y agregando su comentario adicional describiendo el problema. No guardaremos ninguna información de este correo electrónico.

Describe el problema

Escriba aquí una descripción del problema...

Enviar

© 2017 Todos los derechos reservados | Diseño basado en w3layouts.

Figura 57: Pantalla de reportar informe

6.2.3.15 Descargar comentarios

La otra opción que aparece en la barra lateral para los usuarios con rol de 'Cliente Usuario' es la de descargar comentarios. Desde esta pantalla se puede obtener un Excel con los comentarios valorados o no valorados y toda la información relacionada con cada uno de los comentarios. También se puede filtrar la información que aparecerá en dicho Excel a través de una lista de palabras prohibidas, las cuales serán ignoradas del resultado, así como a través de consultas sobre algún atributo almacenado de los productos o los comentarios.

WID Panel

Ver informes

Descargar comentarios

Comentarios no valorados

Comentarios valorados

Palabras prohibidas (palabras que serán ignoradas)

Separar por comas (los espacios serán tenidos en cuenta)

Cuya/o id de los productos sea igual

Cuya/o autoid de los comentarios sea igual

Descargar

© 2017 Todos los derechos reservados | Diseño basado en w3layouts.

Figura 58: Pantalla de descargar comentarios

6.3. Conclusiones y trabajo futuro

Tras el desarrollo del proyecto aquí se expresa las conclusiones a las que se ha llegado así como el trabajo futuro a realizar sobre este proyecto.

6.3.1. Objetivos alcanzados

El objetivo principal de la aplicación era aportar la posibilidad de generar diferentes tipos de informes sobre unos datos recogidos. La base del proyecto es la utilización del web scraping para obtener la información de la página web objetivo sin la necesidad de tener algún permiso para acceder a los datos deseados, y a partir de esta base se proporcionan herramientas para la manipulación y manejo de los datos, dando lugar a la obtención de diferentes tipos de informes a partir de estos datos.

Otro objetivo importante era que, además de los informes gráficos dentro de la aplicación web, también se pudiera generar un informe que fuese utilizable para herramientas avanzadas de análisis de sentimientos, como por ejemplo Weka. Este objetivo se consigue a través de ofrecer al usuario la posibilidad de valorar personalmente los comentarios obtenidos y de descargar un archivo Excel con dichos comentarios y la valoración, positiva o negativa, dada. Este Excel servirá para otras herramientas externas de análisis.

El resto de objetivos que han sido satisfechos están explicados y detallados en la sección 5.2.

6.3.2. Lecciones aprendidas

El desarrollo del proyecto ha conllevado dar uso a todo lo aprendido durante el Grado de Ingeniería Informática. Lo que considero más importante es que en el Grado he aprendido a enfrentarme a cualquier lenguaje de programación sin necesidad de conocerlo, considero que he adquirido las herramientas necesarias para afrontar cualquier problema en cualquier lenguaje, y ser capaz de aprender y darle una solución.

Por esto mismo, quería que el proyecto incluyese Python, un lenguaje de programación que he utilizado en muy pocas ocasiones y para cosas muy básicas, pero que serviría para mostrar la capacidad de utilizar un lenguaje que no conozco en profundidad para resolver una problemática dada. Además me informaron de la herramienta Beautiful Soup, que me ofrecía dentro de Python las opciones que necesitaba para mi proyecto.

Por otro lado he utilizado el framework CodeIgniter para la aplicación web. La utilización de este framework, además de que encajaba en la idea que quería trasladar a la página web, es que a nivel personal iba a profundizar en un framework que, por un lado ya había utilizado en varios pequeños proyectos durante el Grado, y por otro lado estaba utilizando en mi empleo. También en relación con mi empleo hice uso de la herramienta de JavaScript Chart.js, que nunca la había usado pero iba a tener que usarla en mi ámbito profesional, y en el desarrollo de este proyecto aprendí su funcionamiento para implementarla al generar la visualización de las gráficas a partir de las consultas de los clientes.

6.3.3. Trabajo futuro

El funcionamiento del proyecto implica la constante atención de los administradores, puesto que en muchos casos habrá que generar archivos de configuración en determinados clientes, así como estar atentos a su mantenimiento.

Respecto a la aplicación web tiene un código simple y estructurado que facilita la escalabilidad, es decir, el agregar nuevas funcionalidades de forma cómoda. Por la experiencia que he tenido en otros proyectos, cuando este sale al mercado los clientes suelen pedir funcionalidades extras que le faciliten las tareas comunes que ofrece la aplicación, funcionalidades que no suele verse durante el desarrollo. Por lo que una de estas tareas a futuro sería mantener la comunicación con los primeros clientes e implementar las peticiones más genéricas e interesantes que los primeros clientes aportasen.

Una vez afianzado en el mercado español y haber conseguido generar una aplicación que satisfaga las futuras peticiones más habituales de estos clientes, habría que implementar otros idiomas para buscar clientes a nivel internacional.

Algunas de las funcionalidades extras que pueden implementarse son, por ejemplo:

- Implementar un número máximo de intentos fallidos al iniciar sesión y/o un sistema CAPTCHA.
- Mejorar el diseño gráfico de los correos electrónicos automatizados agregándole algún banner y logotipo.
- Administrar desde la aplicación web el programa del web scraping, dando la opción de ejecutarlo, programar las fecha de ejecución y editar el archivo de configuración.
- Agregar nuevos tipos de gráficas en los informes que los clientes generan, como por ejemplo, una tabla de datos.
- Agregar concurrencia a la aplicación que realiza el web scraping para que disminuya el tiempo de obtención de los datos, pero proporcionando una forma de control para que la web objetivo no lo considere un ataque.
- Proporcionar una opción al programa de Python para poder agregar o eliminar columnas de las tablas de la base de datos, es decir, poder agregar o eliminar datos a obtener de la página web objetivo, sin tener que reiniciar toda la estructura y contenido de las tablas.

Referencias

- [1] FORO DE PREGUNTAS Y RESPUESTAS, Stackoverflow
http://stackoverflow.com/
Fecha de último acceso: 6 de marzo de 2018

- [2] THE WORLD'S LARGEST WEB DEVELOPER SITE
http://www.w3schools.com/
Fecha de último acceso: 14 de enero de 2018

- [3] REFERENCIA DE MySQL
http://dev.mysql.com/doc/refman/5.7/en/
Fecha de último acceso: 15 de octubre de 2017

- [4] HOSTINGS GRATUITOS
http://www.hostinger.es/
Fecha de último acceso: 14 de enero de 2018

- [5] DOCUMENTACIÓN TEÓRICA SOBRE PYTHON 3
http://docs.python.org.ar/tutorial/3/index.html
Fecha de último acceso: 22 de septiembre de 2017

- [6] USO DE MySQLDB EN PYTHON
http://librosweb.es/libro/python/capitulo_12.html
Fecha de último acceso: 9 de abril de 2017

- [7] EJEMPLO PRÁCTICO DE CREACIÓN DE LOGIN EN CODEIGNITER
https://www.formget.com/form-login-codeigniter/
Fecha de último acceso: 9 de septiembre de 2017

- [8] MANUALES Y CURSOS TEÓRICOS Y PRÁCTICOS DE CODEIGNITER
https://www.uno-de-piera.com
Fecha de último acceso: 20 de diciembre de 2017

- [9] PHPExcel CON CODEIGNITER
https://arjunphp.com/how-to-use-phpexcel-with-codeigniter/
Fecha de último acceso: 8 de diciembre de 2017

- [10] USO DE CHARTJS
http://blog.hostdime.com.co/visualizar-estadisticas-de-forma-elegante-con-chart-js/
Fecha de último acceso: 17 de diciembre de 2017

- [11] PARALELISMO EN PYTHON
http://chriskiehl.com/article/parallelism-in-one-line/
Fecha de último acceso: 13 de noviembre de 2016
- [12] PRUEBAS UNITARIAS EN PYTHON
https://code.tutsplus.com/es/tutorials/write-professional-unit-tests-in-python--cms-25835
Fecha de último acceso: 28 de julio de 2017
- [13] PRUEBAS EN PYTHON
http://mundogeek.net/archivos/2008/09/17/pruebas-en-python/
Fecha de último acceso: 28 de julio de 2017
- [14] GUÍA DE USO DEL FRAMEWORK UNITTEST
http://www.w3ii.com/es/unittest_framework/unittest_framework_quick_guide.html
Fecha de último acceso: 29 de julio de 2017
- [15] CREACIÓN DE ARCHIVOS XML DESDE PYTHON
http://www.decodigo.com/python-crear-archivo-xml
Fecha de último acceso: 30 de julio de 2017
- [16] TEORÍA BÁSICA SOBRE UNITTEST
http://www.elmundoenbits.com/2016/01/python-unittest-I.html
Fecha de último acceso: 25 de junio de 2017
- [17] TEORÍA SOBRE REQUISITOS NO FUNCIONALES
http://www.pmoinformatica.com/2015/05/requerimientos-no-funcionales-ejemplos.html
Fecha de último acceso: 10 de diciembre de 2017
- [18] TEORÍA SOBRE REQUISITOS FUNCINALES
http://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html
Fecha de último acceso: 10 de diciembre de 2017
- [19] GENERACIÓN DE DIAGRAMAS DE CASOS DE USO
https://yuml.me/diagram/scruffy/usecase/draw
Fecha de último acceso: 10 de septiembre de 2017
- [20] GENERACIÓN DE DIAGRAMAS DE CLASES
https://yuml.me/diagram/nofunky/class/draw
Fecha de último acceso: 10 de septiembre de 2017

- [21] TEORÍA SOBRE EL ANÁLISIS ESTÁTICO DEL CÓDIGO A TRAVÉS DE PYLINT
<https://milmazz. uno/article/2010/03/13/pylint-analisis-estatico-del-codigo-en-python/>
Fecha de último acceso: 7 de julio de 2017
- [22] DOCUMENTACIÓN DE PYLINT
<https://docs.pylint.org/en/1.6.0/features.html>
Fecha de último acceso: 9 de julio de 2017
- [23] DOCUMENTACIÓN SOBRE LOS CÓDIGOS QUE DEVUELVE PYLINT
<http://pylint-messages.wikidot.com/>
Fecha de último acceso: 23 de julio de 2017
- [24] BUENAS PRÁCTICAS EN PYTHON
<https://docs.quantifiedcode.com/python-anti-patterns/index.html>
Fecha de último acceso: 23 de julio de 2017
- [25] LA GUÍA DE ESTILOS DE PYTHON
<https://www.python.org/dev/peps/pep-0008/>
Fecha de último acceso: 23 de julio de 2017
- [26] MANUAL DE LATEX
https://es.wikibooks.org/wiki/Manual_de_LaTeX
Fecha de último acceso: 6 de marzo de 2018