



Treball fi de carrera

GRAU D'ENGINYERIA INFORMÀTICA

**Facultat de Matemàtiques
Universitat de Barcelona**

**Complaints management service with
responsive web.**

Sergi Carmona Castro

Director/s: Mireia Ribera
Realitzat a: Departament de
Matemàtica Aplicada i
Anàlisi. UB

Barcelona, 27 de Junio de 2015

Índice

Abstract (English)	4
Abstract (Castellano).....	5
1 Introducción	6
1.1 Motivación	7
1.2 Objetivos	7
1.3 Organización de la memoria.....	8
2. Análisis de los requisitos.....	9
2.1. Análisis de requisitos y funcionalidades	9
2.2. Requisitos funcionales.....	9
2.2.1 Casos de uso	10
2.2.1.1. Usuario anónimo.	20
2.2.1.2. Usuario registrado.	21
2.2.1.3. Usuario administrador	22
2.3. Requisitos no funcionales.....	23
3. Diseño del modelo de datos	29
3.1. Modelo Entidad / Relación (ER).....	29
3.2. Modelo de datos, tablas de la base de datos.....	30
4. Requerimientos generales: software y hardware.....	30
4.1. Framework Twitter Bootstrap 3	30
4.2. HML5 & CSS3.....	39
5. Implementación	48
5.1. Servidor XAMPP	48
5.2. Estructura FrontEnd	49
5.3. Estructura BackEnd.....	52
6. Planificación.....	54
7. Conclusiones y trabajo futuro	55
7.1 Conclusión.....	55
7.2 Trabajo futuro	55
7.2.1 Encriptacion de contraseñas con sha512	55
7.2.2 Evitar inyección SQL	56
7.2.3 Minimizar el contenido CSS/HTML.....	56
7.2.4 Realizar las peticiones a la base de datos con Ajax.....	56
7.2.5 Gráficos FrontEnd y BackEnd.....	57
7.2.6 Realizar pruebas de usuario	57

7.2.7	Accesibilidad	57
7.2.8	Iniciar sesión con perfiles sociales	58
7.2.9	Publicación de quejas con Drag & Drop	58
7.2.10	Posicionamiento SEO	58
7.2.11	Dinamismo en la sección de categorías	58
8.	Bibliografía.....	59
9.	Anexo	60
9.1	Manual de usuario administrador (BackEnd).....	60
9.2	Prototipo Inicial.....	74

Abstract (English)

The Project called “Complaints management Service with a responsive web” is a web application specially designed for the management and evaluation of complaints proposed by citizens.

The main function of the web application is to provide citizens a fast, suitable, indicative and easy tool to send their complaints. Our specialized team will manage the complaints and make them available to the rest of citizens who want to contribute to their resolution.

In order to perform our objective, we have proposed the creation of an start-up as we want to create a new company based on an innovative idea and looking for complicated processes to be easier.

In the short term, the idea of this project is to create a small community which responds to the complaints of citizens. This community will consist on citizens who want to contribute to the resolution of complaints, but also on a team of professionals who would be responsible to provide verified information to the user for the resolution of his complaint or to deal with it according to the user’s requirements.

At the beginning of the process, a responsible person is needed for the maintenance of the web application. In the long term, if a complaint is processed formally, we will need some lawyers to carry out the complaint according to the procedures established by law. This personalized management will have a cost for the citizen as our specialized team will have to make the necessary arrangements to get this complaint solved as soon as possible, fulfilling our purpose of helping citizens in solving their problem.

The web application has been developed with the aim of providing citizens with the possibility to make a complaint from anywhere, making it adaptable to the type of device they use and to enhance the experience from devices such as notebooks, tablets or smartphones.

In this document, the design, analysis, planning and architecture process of the web application is detailed.

Abstract (Castellano)

El proyecto “Complaints management Service with responsive web” es una aplicación web destinada a la gestión y valoración de las quejas propuestas por los ciudadanos.

La función principal de la aplicación, es ofrecer al ciudadano una herramienta rápida, adaptada, orientativa y fácil de utilizar para poder enviar sus quejas y que nuestro equipo especializado pueda gestionarlas y ponerlas al alcance del resto de ciudadanos que quieran colaborar con la resolución de la queja.

Para poder llevar a cabo nuestro objetivo, se ha propuesto la creación de una empresa “Start-Up” [24], ya que lo que se persigue es emprender un nuevo negocio aportando una idea innovadora del mercado y buscando hacer que los procesos complicados sean más fáciles de realizar.

La idea de este proyecto es crear a corto plazo, una pequeña comunidad que dé respuesta a las quejas de los ciudadanos.

Esta comunidad estará formada por ciudadanos que quieran contribuir en la resolución de las quejas, pero sobretodo y lo más importante, con un equipo de profesionales que se encargarían de facilitar información contrastada al usuario, para la resolución de su queja o bien tramitarlas en el caso de que así lo prefiera el ciudadano.

Ello supondrá inicialmente, una persona encargada del mantenimiento de la aplicación web.

En el caso de tramitar oficialmente la queja, a largo plazo necesitaríamos contar con una serie de abogados para llevar a cabo estas gestiones según los procedimientos establecidos por la ley. Ésta gestión personalizada tendrá un coste para el ciudadano, ya que nuestro equipo especializado tendrá que realizar los trámites necesarios para que ésta queja sea atendida con la mayor brevedad posible, cumpliendo así nuestro propósito de ayudar al ciudadano en la resolución de su problema.

La aplicación web ha sido desarrollada con la voluntad de facilitar al ciudadano la resolución de sus quejas, ágilmente, con la mayor brevedad posible y que sea gestionable desde cualquier sitio, haciéndola así adaptable al tipo de dispositivo que utilice y poder mejorar la experiencia desde dispositivos portátiles, como tablets y móviles.

En el presente documento se detalla el proceso de diseño, análisis, planificación y arquitectura de la plataforma proyectada.

1 Introducción

En España, existe una gran diversidad de comportamientos o actitudes de los ciudadanos a la hora de realizar reclamaciones en los servicios.

Los sectores que reciben un mayor índice de reclamaciones son en su conjunto las comunicaciones electrónicas, la telefonía y el acceso a internet.

Por norma general, ante un problema, el consumidor recurre directamente al proveedor de servicios.

Si ésta opción no funciona, recurre a otros medios más cercanos que suelen ser las organizaciones de consumidores.

Según un estudio publicado por el Instituto Nacional de Consumo(INC) sobre el índice de insatisfacción del consumidor, podemos resumir que las reclamaciones se canalizan de la siguiente manera:

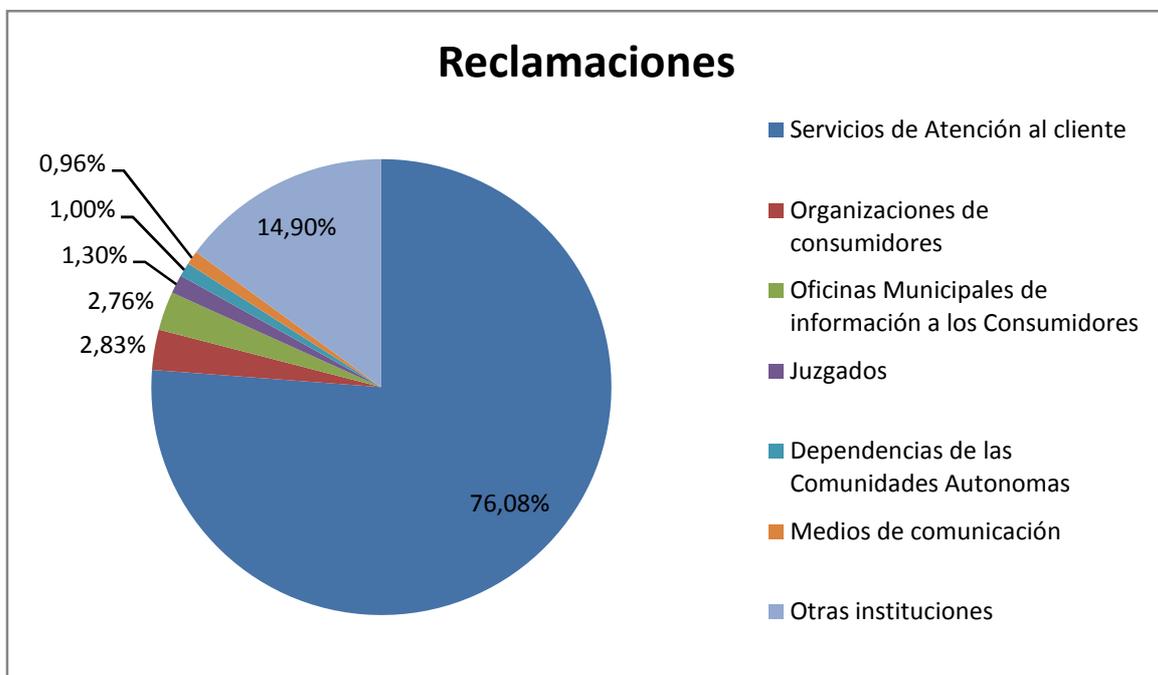


Ilustración 1: Canalización de las reclamaciones en España. Fuente [25]

Según el INC el 41,2% de las quejas efectivamente presentadas, no han sido resueltas a través de los canales previstos.

Un dato importante que podemos sacar de éste estudio es que prácticamente no se recurre al uso de las herramientas de la información tales como páginas web dedicadas a la gestión exclusiva de las quejas o reclamaciones de los ciudadanos.

Utilizando esta carencia como oportunidad para el desarrollo de éste proyecto, se quiere ofrecer tanto a los ciudadanos con un alto grado de insatisfacción en los que encuentran ese 41.2% de las quejas efectivamente presentadas, como para el resto de los ciudadanos, un canal alternativo para la resolución de sus quejas.

1.1 Motivación

Este proyecto surge principalmente por diferentes motivos:

- Proporcionar al ciudadano un canal alternativo para la resolución de sus quejas, ofreciendo así una herramienta efectiva y que le permita ser consultada desde cualquier lugar.
- Ofrecer al ciudadano una aplicación adaptable al dispositivo que utilice. Para ello éste proyecto se centrará en el diseño responsivo[1], evitando así tener que crear diferentes versiones de la aplicación para las diversas resoluciones de los dispositivos actuales.
- Las “Start-Up”, por lo general comienzan como una idea de negocio creativo y el paso inmediato es agregar diferenciación a dicha idea a través de la innovación, para finalmente emprender el negocio. Para poder convertir la idea en un proyecto real se ha decidido crear una pequeña compañía “Start Up”.
- **Motivación personal:**

Con éste proyecto quiero aprender el uso de las nuevas tecnologías de maquetado y diseño web como HTML5 y CSS3 [13], así como el manejo del lenguaje de programación PHP [26]. En cuanto a la gestión y optimización de la base de datos he decidido utilizar el open source con mayor aceptación mundial que permite la creación asequible y fiable de aplicaciones de bases de datos integradas basadas en web de alto rendimiento y fácilmente ampliables, conocido como MySQL [27].

Por último, quiero aplicar los conocimientos obtenidos a lo largo de mi formación en un proyecto real a largo plazo.

1.2 Objetivos

Creación de una aplicación web para poder facilitar al ciudadano un derecho tan básico como el poder quejarse.

La idea de este proyecto es crear a corto plazo, una pequeña comunidad donde se plasmen las quejas de los ciudadanos.

Esta comunidad estará formada por ciudadanos que quieran contribuir en la resolución de las quejas, pero sobretodo y lo más importante, con un equipo de profesionales que se encargarían de facilitar información al usuario para la resolución de su queja o bien tramitarlas en el caso de imposibilidad del ciudadano. En el caso de tramitar oficialmente la queja, a largo plazo necesitaríamos contar con una serie de abogados para llevar a cabo estas gestiones de forma legal.

Con ésta aplicación, se quiere ofrecer un aplicativo rápido y fiable a la hora de poder hacer llegar las quejas de los ciudadanos.

Otra de las funciones principales será la de compartir las quejas que envían los ciudadanos con el resto de la sociedad.

Habrà un administrador web, el cual se encargará de gestionar todo el aplicativo, ya sean las quejas que reciba de los ciudadanos como el servicio técnico para dudas del aplicativo. Ejercerá como moderador de los comentarios que se realicen en las quejas, ya que cualquier ciudadano que se haya registrado puede dar su opinión sobre cualquier queja teniendo siempre en cuenta que el objetivo principal es la resolución del problema del ciudadano por la mejor vía.

El ciudadano sin necesidad de estar registrado en el sistema podrá enviar sus quejas para que se puedan gestionar.

Una vez el ciudadano ingrese en el sistema y forme parte de la comunidad, podrá interactuar con otros usuarios dando su opinión y valorando positiva o negativamente las diferentes quejas propuestas en la aplicación web.

1.3 Organización de la memoria

Se incluyen al detalle los siguientes puntos del proyecto:

- Capítulo 2. Análisis de los requisitos de la web.
- Capítulo 3. Modelo de datos.
- Capítulo 4. Tecnologías utilizadas para este proyecto.
- Capítulo 5. Características más relevantes de Bootstrap, HTML5, CSS3 y MySQL.
- Capítulo 6. Estructura del proyecto.
- Capítulo 7. Planificación inicial y final del proyecto
- Capítulo 8. Conclusiones del proyecto y propuestas de mejoras de cara al futuro.

2. Análisis de los requisitos

2.1. Análisis de requisitos y funcionalidades

Para el desarrollo de este proyecto se analizaron una serie de requisitos y funcionalidades que debía cumplir.

A medida que iba avanzando el proyecto se iban construyendo muchas de las funcionalidades y requisitos de la aplicación web.

El requisito fundamental de este proyecto es proporcionar al ciudadano un canal de comunicación para poder exponer sus quejas.

Este proyecto contará con una amplia base de datos de quejas que los usuarios harán llegar a través de la página web.

Las quejas deberán hacerse de forma intuitiva y cómoda, tanto para el ciudadano que expone su problema como para el gestor o responsable que lo gestiona.

2.2. Requisitos funcionales

Definición de los servicios que el sistema debe proporcionar, cómo debe reaccionar a una entrada particular y cómo se debe comportar ante situaciones particulares.

Separamos en 3 grupos los requisitos de la página web en función del tipo de usuario que ingresa en ella:

- Usuario anónimo: Usuario que accede a nuestra página web sin necesidad de estar registrado en el sistema.
- Usuario registrado: Usuario que accede a nuestra página web identificándose con un usuario y una contraseña.
- Usuario administrador: Usuario que accede a nuestra página identificándose con un usuario y una contraseña en concreto que le permite poder gestionar la web.

A continuación se detallan los diferentes casos de uso separados por tipo de usuario. Se incluye una breve descripción para cada uno de ellos.

2.2.1 Casos de uso

Antes de entrar al detalle con los casos de usos de cada usuario, se indican las tareas que pueden realizar los usuarios que no tengan rol de administrador, independientemente de estar o no registrados en nuestro sistema.

Se utilizan las siglas UCX donde UC significa "Caso de Uso" y "X" es el numero identificativo del caso.

UC0.Login	
Objetivo	Validarse en el sistema
Actores	Usuario anónimo
Precondiciones: -	
Descripción: <ol style="list-style-type: none">1. Usuario: Entrar en el apartado "Login/Registro"2. Sistema: Procesar la petición solicitada3. Usuario: Acceder al formulario de login4. Sistema: Mostrar el formulario de login5. Usuario: Rellenar los datos y enviar el formulario6. Sistema: Validar los datos7. Sistema: Autenticar al usuario en el sistema8. Sistema: Re-dirigir a la pagina inicial	
Escenarios alternativos: <ol style="list-style-type: none">6.1 El usuario no introduce bien los datos. Mostrar mensaje de error si los datos son mal introducidos Volver al paso 5	

UC1. Registro	
Objetivo	Registrarse en el sistema
Actores	Usuario anónimo
Precondiciones: -	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en el apartado "Login/Registro" 2. Sistema: Procesar la información solicitada 3. Usuario: Acceder al formulario de registro 4. Sistema: Mostrar el formulario de registro 5. Usuario: Rellena los datos y enviar el formulario 6. Sistema: Validar los datos 7. Sistema: Re-dirigir al usuario a la página de registrado con éxito 	
Escenarios alternativos:	
<ol style="list-style-type: none"> 6.1 El usuario no introduce bien los datos. Mostrar mensaje de error si los datos son mal introducidos Volver al paso 5 6.2 Usuario duplicado: Redirigir al usuario a la página de login mostrando el mensaje de usuario duplicado. Volver al paso 4 	

UC2. Enviar una queja	
Objetivo	Enviar una queja para publicarla en la página web.
Actores	Usuario anónimo y usuario registrado
Precondiciones: -	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en el apartado "nueva queja" 2. Sistema: Procesar la información solicitada 3. Usuario: Acceder al formulario de envío 4. Sistema: Mostrar el formulario de envío 5. Usuario: Rellenar los datos y enviar el formulario 6. Sistema: Validar los datos 7. Sistema: Re-dirigir al usuario a la página de mensaje enviado con éxito 	
Escenarios alternativos:	
<ol style="list-style-type: none"> 6.1 El usuario no introduce bien los datos. Mostrar mensaje de error si los datos son mal introducidos Volver al paso 5 	

UC3.Comentar queja	
Objetivo	Escribir un comentario en la queja que el usuario desee
Actores	Usuario registrado
Precondiciones: Estar registrado en el sistema	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en el apartado quejas 2. Sistema: Procesar la información solicitada 3. Usuario: Seleccionar la queja que se quiere comentar 4. Sistema: Mostrarla ficha de la queja 5. Usuario: Desplazar hasta la sección “lista de comentarios” 6. Usuario: Escribir el comentario 7. Sistema: Validar los datos 8. Sistema: Re-dirigir al usuario a la página de “quejas” con la alerta de mensaje enviado con éxito 	
Escenarios alternativos:	
6.1 Mostrar mensaje de error si los datos son mal introducidos	

UC4. Ver queja	
Objetivo	Ver las quejas que están dadas de alta en nuestro sistema
Actores	Usuario anónimo y usuario registrado
Precondiciones: -	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en el apartado quejas 2. Sistema: Procesar la información solicitada 	

UC5. Ver comentarios	
Objetivo	Ver los comentarios de una queja que esté dada de alta en nuestro sistema
Actores	Usuario anónimo y usuario registrado
Precondiciones: -	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en el apartado “quejas” 2. Sistema: Procesar la información solicitada 3. Usuario: Seleccionar la queja que quiere ver los comentarios 4. Usuario: Desplazar hasta la sección “lista de comentarios” 	

UC6. Valorar queja	
Objetivo	Valorar positiva o negativamente la queja que deseemos.
Actores	Usuario registrado
Precondiciones: Estar registrado en el sistema	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en el apartado “quejas” 2. Sistema: Procesar la información solicitada 3. Usuario: Seleccionar la queja que quiere valorar 4. Sistema: Mostrar la ficha de la queja 5. Usuario: Seleccionar “me gusta” o “no me gusta”. 6. Sistema: Validar los datos 	
Escenarios alternativos:	
<p>4.1Sistema: Si el usuario ya ha votado, muestra mensaje conforme solo puede valorar una vez.</p> <p>6.1Sistema: Oculta los botones de “me gusta” y “no me gusta”</p>	

UC7. Listar quejas por categorías	
Objetivo	Mostrar las quejas por categorías
Actores	Usuario anónimo y usuario registrado
Precondiciones: -	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en el apartado quejas 2. Sistema: Procesar la información solicitada 3. Usuario: Seleccionar “filtrar por categorías” 4. Sistema: Mostrar el listado de todas las categorías 5. Usuario: Seleccionar la categoría por la cual quiere filtrar 6. Sistema: Mostrarlas quejas filtradas por la categoría seleccionada 	
Escenarios alternativos:	
<ol style="list-style-type: none"> 6.1 Sistema: Si no hay quejas para esa categoría, muestra un mensaje indicándolo. 	

UC8. Ver quejas más comentadas	
Objetivo	Ver un ranking de las quejas más comentadas
Actores	Usuario anónimo y usuario registrado
Precondiciones: -	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en la página de inicio 2. Sistema: Procesar la información solicitada 3. Sistema: Mostrar la página de inicio 	

UC9. Ver los usuarios que más quejas han comentado	
Objetivo	Ranking de los usuarios que más quejas han comentado
Actores	Usuario anónimo y usuario registrado
Precondiciones: -	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en la página de inicio 2. Sistema: Procesar la información solicitada 3. Sistema: Mostrar la página de inicio 	

A continuación se indicarán las tareas que puede realizar el usuario con rol de administrador. A parte de todas las tareas anteriores, el **administrador** puede:

UC10.Crear queja	
Objetivo	Dar de alta nuevas quejas en el sistema
Actores	Usuario administrador
Precondiciones: Entrar al panel de administración	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en la tabla “quejas” 2. Sistema: Procesar la información solicitada 3. Usuario: Pulsar en “nueva queja” 4. Sistema: Mostrar formulario para crear una nueva queja 5. Usuario: Rellenar los datos y enviar el formulario 6. Sistema: Validar los datos 7. Sistema: Re-dirigir a la tabla de quejas 	
Escenarios alternativos:	
<ol style="list-style-type: none"> 6.1 El usuario no introduce bien los datos. Mostrar mensaje de error si los datos son mal introducidos Volver al paso 5 6.2 Sistema: Queja duplicada, se mostrará un mensaje de alerta. 	

UC11. Modificar queja	
Objetivo	Modificar la queja seleccionada
Actores	Usuario administrador
Precondiciones: Entrar al panel de administración	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en la tabla “quejas” 2. Sistema: Procesar la información solicitada. 3. Usuario: Seleccionar la queja que quiere modificar 4. Sistema: Mostrar el formulario de modificación de la queja 5. Usuario: Rellenar los datos y enviar el formulario 6. Sistema: Validar los datos 7. Sistema: Re-dirigir a la tabla de quejas 	
Escenarios alternativos:	
<ol style="list-style-type: none"> 6.1 El usuario no introduce bien los datos. Mostrar mensaje de error si los datos son mal introducidos Volver al paso 5 6.2 Sistema: Queja duplicada, se mostrará un mensaje de alerta. 	

UC12. Eliminar queja	
Objetivo	Dar de baja una queja del sistema
Actores	Usuario administrador
Precondiciones: Entrar al panel de administración	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en la tabla “quejas” 2. Sistema: Procesar la información solicitada 3. Usuario: Seleccionar la queja que desea eliminar 4. Sistema: Mostrar un aviso para confirmarla eliminación 5. Usuario: Confirmar la eliminación 6. Sistema: Validar los datos 7. Sistema: Redirigir a la tabla de quejas 	

UC13. Ocultar queja	
Objetivo	Ocultar una queja que ya ha quedado obsoleta
Actores	Usuario administrador
Precondiciones: Entrar al panel de administración	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en la tabla “quejas” 2. Sistema: Procesar la información solicitada 3. Usuario: Seleccionar la queja que quiere ocultar y “modificar” 4. Sistema: Mostrar el formulario de modificación 5. Usuario: Escribir una “s” minúscula en el campo “ocultar” y envía el formulario. 6. Sistema: Validar los datos 7. Sistema: Redirigir a la tabla de quejas 	

UC14. Crear usuario	
Objetivo	Dar de alta un nuevo usuario en el sistema
Actores	Usuario administrador
Precondiciones: Entrar al panel de administración	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en la tabla “usuarios” 2. Sistema: Procesar la información solicitada 3. Usuario: Pulsar en “nuevo usuario” 4. Sistema: Mostrar formulario para crear un nuevo usuario 5. Usuario: Rellenar los datos y enviar el formulario 6. Sistema: Validar los datos 7. Sistema: Redirigir a la tabla de usuarios 	
Escenarios alternativos:	
<ol style="list-style-type: none"> 6.1 El usuario no introduce bien los datos. Mostrar mensaje de error si los datos son mal introducidos Volver al paso 5 6.2 Sistema: Usuario duplicada, se mostrará un mensaje de alerta. 	

UC15. Modificar usuario	
Objetivo	Modificar el usuario seleccionado
Actores	Usuario administrador
Precondiciones: Entrar al panel de administración	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en la tabla “usuarios” 2. Sistema: Procesar la información solicitada. 3. Usuario: Seleccionar el usuario que quiere modificar 4. Sistema: Mostrar formulario de modificación del usuario 5. Usuario: Rellenar los datos y enviar el formulario 6. Sistema: Validar los datos 7. Sistema: Redirigir a la tabla de usuarios 	
Escenarios alternativos:	
<ol style="list-style-type: none"> 6.1 El usuario no introduce bien los datos. Mostrar mensaje de error si los datos son mal introducidos Volver al paso 5 6.2 Sistema: Usuario duplicado, se mostrará un mensaje de alerta. 	

UC16. Eliminar usuario	
Objetivo	Dar de baja un usuario del sistema
Actores	Usuario administrador
Precondiciones: Entrar al panel de administración	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en la tabla “usuarios” 2. Sistema: Procesar la información solicitada 3. Usuario: Seleccionar el usuario que desea eliminar 4. Sistema: Mostrar un aviso para confirmarla eliminación 5. Usuario: Confirmar la eliminación 6. Sistema: Validar los datos 7. Sistema: Redirigir a la tabla de usuarios 	

UC17. Crear categoría	
Objetivo	Dar de alta una nueva categoría en el sistema
Actores	Usuario administrador
Precondiciones: Entrar al panel de administración	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en la tabla “categorías” 2. Sistema: Procesar la información solicitada 3. Usuario: Pulsar en “nueva categoría” 4. Sistema: Mostrar formulario para crear una nueva categoría 5. Usuario: Rellenar los datos y envía el formulario 6. Sistema: Validar los datos 7. Sistema: Re-dirigir a la tabla de categorías 	
Escenarios alternativos:	
<ol style="list-style-type: none"> 6.1 El usuario no introduce bien los datos. Mostrar mensaje de error si los datos son mal introducidos Volver al paso 5 6.2 Sistema: Categoría duplicada, se mostrará un mensaje de alerta. 	

UC18. Modificar categoría	
Objetivo	Modificar la categoría seleccionada
Actores	Usuario administrador
Precondiciones: Entrar al panel de administración	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en la tabla “categorías” 2. Sistema: Procesar la información solicitada. 3. Usuario: Seleccionar la categoría que quiere modificar 4. Sistema: Mostrar formulario de modificación del usuario 5. Usuario: Rellenar los datos y enviar el formulario 6. Sistema: Validar los datos 7. Sistema: Re-dirigir a la tabla de categorías 	
Escenarios alternativos:	
<ol style="list-style-type: none"> 6.1 El usuario no introduce bien los datos. Mostrar mensaje de error si los datos son mal introducidos Volver al paso 5 6.2 Sistema: Categoría duplicada, se mostrará un mensaje de alerta. 	

UC19. Eliminar categoría	
Objetivo	Dar de baja una categoría del sistema
Actores	Usuario administrador
Precondiciones: Entrar al panel de administración	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en la tabla “categorías” 2. Sistema: Procesar la información solicitada 3. Usuario: Seleccionarla categoría que desea eliminar 4. Sistema: Mostrar un aviso para confirmarla eliminación 5. Usuario: Confirmar la eliminación 6. Sistema: Validar los datos 7. Sistema: Re-dirigir a la tabla de categorías 	

UC20. Logout	
Objetivo	Desconectar al usuario de la sesión actual
Actores	Usuario registrado
Precondiciones: -	
Descripción:	
<ol style="list-style-type: none"> 1. Usuario: Entrar en la web 2. Sistema: Procesar la información solicitada 3. Usuario: Hacer click sobre la opción "CERRAR SESION" 4. Sistema: Procesa la información y des-autentifica al usuario 	

2.2.1.1. Usuario anónimo.

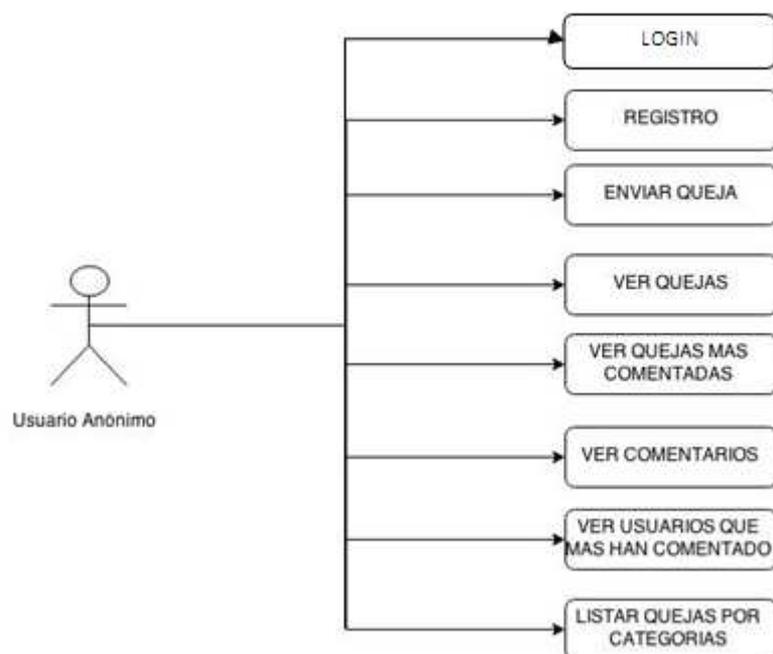


Ilustración 2: Diagrama casos de uso usuario anónimo.

Acciones que puede realizar:

- **Login:** El usuario anónimo pasa a ser usuario registrado cuando se valida en nuestro sistema. [\[UC0\]](#)
- **Registro:** El usuario anónimo puede ingresar en nuestro sistema utilizando un sistema de registro. [\[UC1\]](#)

- **Enviar queja:**El usuario anónimo puede enviar una queja para que el moderador del sitio web pueda aprobarla y publicarla en la web.[UC2]
- **Ver quejas:** El usuario anónimo puede ver todas las quejas publicadas en nuestro sistema.[UC4]
- **Ver quejas más comentadas:** El usuario anónimo puede ver un ranking con las quejas más comentadas.[UC8]
- **Ver comentarios:** El usuario anónimo puede ver los comentarios en las quejas publicadas en nuestro sistema.[UC5]
- **Ver usuarios que más han comentado:** El usuario anónimo puede ver un ranking con los usuarios que han dejado más comentarios.[UC9]
- **Listar quejas por categorías:** El usuario anónimo puede seleccionar una categoría y ver cuáles son las quejas que tienen asignada esa categoría.[UC7]

La ilustración 2 muestra los casos de uso de éste usuario.

2.2.1.2. Usuario registrado.

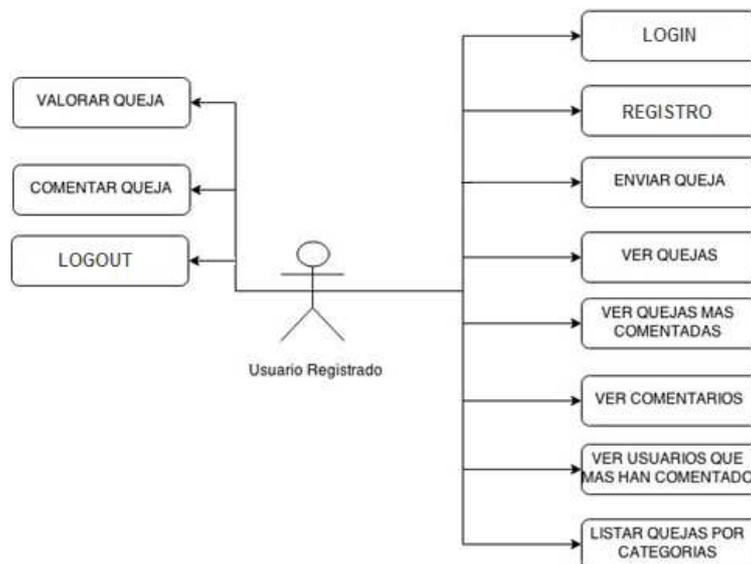


Ilustración 3: Diagrama casos de uso usuario registrado.

El usuario anónimo una vez se registra en nuestro sistema, pasa a ser un usuario registrado donde puede realizar las mismas tareas que el usuario anónimo con la excepción de que también puede:

- **Valorar queja:** El usuario registrado puede valorar las quejas de otros usuarios.[UC6]
- **Comentar queja:** El usuario registrado puede comentar la queja que seleccione.[UC3]
- **Logout:** El usuario registrado puede salir del sistema .[UC20]

La ilustración 3 muestra los casos de uso de éste usuario.

2.2.1.3. Usuario administrador

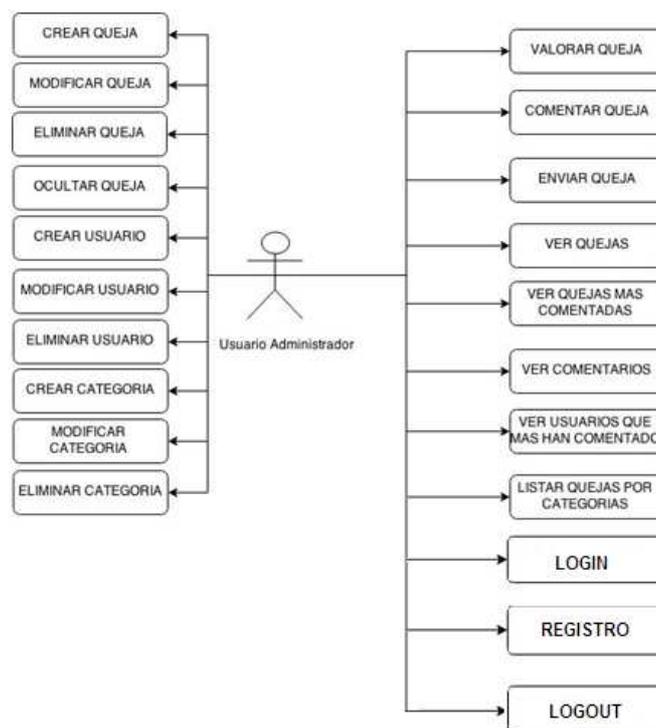


Ilustración 4: Diagrama casos de uso usuario administrador.

El usuario administrador, es el único usuario con unos roles especiales que le permiten entrar en el panel de administración o “backend” para poder gestionar el contenido de la aplicación web.

A parte de todas las funciones anteriores, estos roles le permiten:

- **Crear queja:** El administrador puede crear nuevas quejas.[UC10]
- **Modificar queja:** El administrador puede modificar quejas.[UC11]
- **Eliminar queja:** El administrador puede eliminar quejas.[UC12]
- **Ocultar queja:** El administrador puede ocultar quejas.[UC13]
- **Crear usuario:** El administrador puede crear usuarios.[UC14]
- **Modificar usuario:** El administrador puede modificar usuarios.[UC15]
- **Eliminar usuario:** El administrador puede eliminar usuarios.[UC16]
- **Crear categoría:** El administrador puede crear quejas.[UC17]
- **Modificar categoría:** El administrador puede modificar quejas.[UC18]
- **Eliminar categoría:** El administrador puede eliminar quejas.[UC19]

2.3. Requisitos no funcionales

Los requisitos no funcionales tienen que ver con las características que de una u otra forma puedan limitar el sistema. Describen una restricción sobre el sistema que limita nuestra elección en la construcción de una solución.

- **Arquitectura:** El sitio web de la aplicación deberá poderse explotar y administrar empleando cualquier navegador web
Los datos de la aplicación deberán estar almacenados en un sistema gestor de bases de datos, sobre el cual puedan realizarse futuras consultas no previstas en la actualidad.
Todas las funcionalidades de la aplicación deberán estar al alcance a través de la interfaz de usuario.
- **Escalabilidad:** El sistema debe de permitir en el futuro el desarrollo de nuevas funcionalidades, y también modificar o eliminar funcionalidades.
Las peticiones concurrentes de acceso a la base de datos deben dejar la aplicación en un estado consistente.

- **Usabilidad:** Es la disciplina que estudia la forma de diseñar sitios web para que los usuarios puedan interactuar con ellos de la forma más fácil, cómoda e intuitiva posible.

La mejor forma de crear un sitio web usable es realizando un diseño centrado en el usuario, diseñando para y por el usuario, en contraposición a lo que podría ser un diseño centrado en la tecnología o uno centrado en la creatividad u originalidad.

El foco de este proyecto ha sido la arquitectura por lo tanto solo se ha realizado una revisión heurística y un análisis de la competencia, pero en el futuro se realizarán pruebas con usuarios.

- **Principios heurísticos de Molich y Nielsen [11]:** La experiencia de usuario, es la respuesta emocional y satisfacción a la hora de interactuar con un sitio web. Cuánto mejor se siente un usuario navegando por la web, más posibilidades hay de que vuelva. Por otra parte, si un usuario tiene que dar muchas vueltas por la web para encontrar el contenido que le interesa, o tiene problemas con la página, es más que probable que no vuelva.

Hay que tener muy en cuenta la experiencia del usuario, para ello a continuación se detallarán los 10 principios de diseño basados en el usuario que se usarán para evaluar la usabilidad de un sitio web.

Visibilidad del Estado del Sistema: El sistema debe tener siempre informado al usuario sobre lo que está ocurriendo.

Lenguaje de los usuarios: Utilizar un lenguaje familiar para el usuario, la información debe aparecer por orden lógico y natural.

Control y libertad para el usuario: El usuario debe disponer de una “ruta de escape” si ha escogido una opción errónea, pudiendo así abandonar el estado no deseado en el que se encuentra.

Consistencia y Estándares: El usuario no tiene porque saber que diferentes palabras o acciones significan lo mismo. Hay que seguir las normas y las convenciones de la plataforma sobre la cual se está implementando el sistema.

Ayuda a los Usuarios para Reconocimiento, Diagnostico y recuperación de errores: Los mensajes de error deben expresarse en un lenguaje sencillo y conocido por el usuario, indicando con exactitud el problema y sugiriendo alternativas o soluciones.

Prevención de errores: Prevenir errores mediante un diseño adecuado, aún así los mensajes de error deben contar con la

confirmación por parte del usuario antes de realizar las acciones de corrección.

Reconocimiento antes que recuerdo: Para evitar que el usuario tenga que recordar la información de las diferentes secciones del sitio web, los objetos, las acciones y opciones deben hacerse visibles así como las instrucciones de uso también deben serlo o bien ser fácilmente localizables.

Flexibilidad y eficiencia de uso: Los atajos de teclado pueden hacer más rápida la interacción para usuarios expertos, de tal forma que la aplicación o sitio web sea útil y fácil de utilizar tanto para usuarios noveles como avanzados.

Diseño estético y minimalista: Los sitios web no deben contener información innecesaria. Cada información extra dificulta la visibilidad de la información relevante.

Ayuda general y Documentación: Lo más correcto es que el usuario pueda utilizar el sitio web sin necesidad de ayuda, pero en algunos casos puede ser necesario proveer cierto tipo de ayuda.

En éste caso, debe ser fácil de localizar y no ser muy extensa.

- **Análisis de la competencia:** Se han analizado diferentes páginas web para determinar si era viable la realización de éste proyecto y para extraer buenas prácticas y oportunidades de mejora. Entre ellas:

-QuejasOnline(<http://quejasonline.com/>):Se puede apreciar un diseño minimalista y muy poco cuidado. La página web se estructura en tres partes bien diferenciadas:

A la izquierda podemos encontrar un menú sobrecargado para el usuario, que contiene las diferentes categorías de las quejas, enlaces a webs interesantes y un módulo de facebook.

A la derecha podemos encontrar una sección que consta de un apartado para que puedas publicar la queja, un buscador que quejas, unos enlaces a facebook, feed y twitter, publicidad y un ranking de las quejas donde no se puede apreciar que criterio utiliza para determinar que queja va por encima de la otra.

En el centro podemos encontrar una sección donde se exponen las quejas, el formato de la queja es: Título, Introducción, Puntuación y un enlace para ampliar la queja. Cada queja está separada por una línea horizontal y al final de la pagina encontramos un paginador para navegar entre las distintas quejas que hay en el sistema.

Procedemos añadiendo una nueva queja, en la sección superior

aparece la normativa a cumplir a la hora de realizar los comentarios, más abajo aparece un formulario con unos campos a rellenar, es un formulario rudimentario donde no se controla si se han rellenado los campos de manera individual, es decir, si envías el formulario sin cumplimentar, simplemente te muestra un mensaje de “faltan datos por rellenar”.

Para un usuario poco familiarizado puede ser complicado determinar que parte del formulario se ha dejado en blanco.

Se puede apreciar que esta web no sigue un diseño responsivo, por lo tanto no está adaptado a dispositivos móviles ni tablets, cosa que dificultará utilizar ésta herramienta a no ser que cuentes con un ordenador.

En definitiva, ésta página web no cumple con los principios heurísticos de:

Visibilidad del Estado del Sistema: Una vez navegas por la página, no hay lugar que te diga dónde te encuentras.

Control y libertad para el usuario: El usuario no dispone de ruta de escape, tiene que hacer “atrás” en el navegador para volver a la página anterior.

Ayuda a los Usuarios para Reconocimiento, Diagnostico y recuperación de errores: Los mensajes de error no especifican claramente cuál es el problema, como cuando intentas enviar un formulario vacío, no te informa de que campo es el que te falta por rellenar.

Prevención de errores: Mucha publicidad colocada en lugares inapropiados, como por ejemplo, antes de la votación de la queja hay una gestión de anuncios que te confunde con la página.

Diseño estético y minimalista: El diseño no es estético y no ofrece nada diferente en ninguna de las diferentes pantallas en las que se navega. Cabe mencionar que no cumple con el diseño responsivo necesario para que sea fácil de utilizar en dispositivos móviles.

Ayuda general y Documentación: No ofrece ninguna sección de ayuda ni de contacto en el caso de tener algún problema.

Otra página analizada ha sido:

-Quejas y reclamaciones (<http://www.quejas-y-reclamaciones.net/>):

Al igual que la página web anterior, podemos apreciar un diseño minimalista y con muy poco detalle.

Se estructura en tres partes no tan diferenciadas:

En la cabecera de la página podemos encontrar un link “Alta Queja”

para dar de alta una nueva queja, a la derecha de la cabecera podemos encontrar tres botones, enlace a Facebook, enlace a RSS y el botón superior sirve para publicar queja.

Empezando por aquí no hace diferenciación entre un botón que es un link externo a Facebook como una parte tan esencial de la pagina web como es el hecho de publicar una queja.

La sección central de la página cuenta con el título de la queja, una pre visualización del contenido de la queja seguido de puntos suspensivos. Debajo de cada queja se encuentra la fecha que se publicó, el número de comentarios que tiene, la empresa de la cual se queja el usuario, la categoría de la queja y dependiendo del tipo de queja que sea aparece un campo llamado “marca”, para identificar la marca del vehículo, producto, etc.

No hay línea de separación entre las quejas, están puestas una detrás de otra.

Después de la última queja se puede apreciar un paginador para navegar entre las distintas quejas.

En la sección de la derecha contamos con, las diferentes categorías de las quejas y un enlace a “+ categorías” permitiendo al usuario poder ver todas las categorías posibles.

A continuación vemos una serie de ranking y secciones donde podemos ver “Top empresas con quejas”, “Artículos con + reclamaciones”, “Reclamaciones por localidades”, “Últimos comentarios”.

En conclusión, desde un inicio el planteamiento de la página web es incorrecto ya que en la publicación de una nueva queja, permite al usuario añadir lo que quiera en el campo de categoría.

Esto es un grave error, ya que si le das libertad al usuario para crear una categoría, tendrás infinitas categorías diferentes y no tendrás correctamente agrupadas las quejas por su verdadera categoría.

No te indica cómo puedes visualizar los comentarios que tiene asociados esa queja, tienes que pulsar en el título para que se te abra la ficha de la queja.

Se puede apreciar que esta web no sigue un diseño responsivo, por lo tanto no está adaptado a dispositivos móviles ni tablets, cosa que dificultará utilizar ésta herramienta a no ser que cuentes con un ordenador.

En definitiva, ésta página web no cumple con los principios heurísticos de:

Control y libertad para el usuario: Dependiendo de la página en la que se encuentre, como por ejemplo, “Ayuda Rápida”, el usuario no dispone de ruta de escape, tiene que hacer “atrás” en el navegador para volver a la página anterior.

Consistencia y Estándares: Para poder ver los comentarios tienes que entrar en la queja en cuestión, en cambio el creador de la página web, en cada queja ha hecho un enlace a los comentarios que está inactivo. Por ejemplo te dice que hay 1 comentario, intentas acceder allí y no hace nada. El usuario no debe saber que ese link no funciona, puede llegar a pensar que no contiene comentarios y ha debido de ser un error.

La barra de navegación está en el pie de página cuando por lógica se debería poner en la cabecera de la página ya que es el mecanismo de navegación donde se ponen datos como “Página Inicial”, “quiénes somos”, “ayuda”, etc.

En todas las páginas hay dos formas muy diferentes para publicar una queja, la primera es lo que llaman “Alta queja” y la segunda “Publicar queja”. Alta queja, es un enlace a la página para dar de alta una nueva queja y publicar queja es un botón con el enlace a la página para dar de alta una nueva queja.

Los dos mecanismos hacen lo mismo, no tiene sentido ponerlo en la misma página y que tengan un diseño diferente si tienen la misma función.

Ayuda a los Usuarios para Reconocimiento, Diagnóstico y recuperación de errores: Los mensajes de error no siguen un estándar lógico, por ejemplo si quieres enviar un comentario sin rellenar nada, te re-dirige a una página php donde te dice “usuario vacío”. En cambio a la hora de dar de alta una queja, si no rellenas nada, te avisa conforme tienes que rellenar los campos obligatorios y te los resalta en naranja.

Prevención de errores: Mucha publicidad colocada en lugares inapropiados, como por ejemplo, antes de la paginación, hay una gestión de anuncios que te confunde con la página.

Reconocimiento antes que recuerdo: Por ejemplo, para añadir un nuevo comentario en una queja, debajo de la publicidad se encuentra un link llamado “Añadir comentario +”. Es de esperar que una acción tan importante como poder comentar una queja, tenga una posición más notoria en la página o bien sea más vistosa.

Diseño estético y minimalista: No cumple con un diseño estético, fondo de la página en negro con las secciones en blanco, donde la página está completamente sobrecargada y llena de publicidad, por lo tanto tampoco cumple el diseño minimalista.

- **Validación:** Obligatoriedad de los campos, manejo del tipo de datos y validación de contraseñas.

3. Diseño del modelo de datos

3.1. Modelo Entidad / Relación (ER)

A continuación se describirán las cardinalidades para que pueda entenderse mejor el diseño de la ilustración 5.

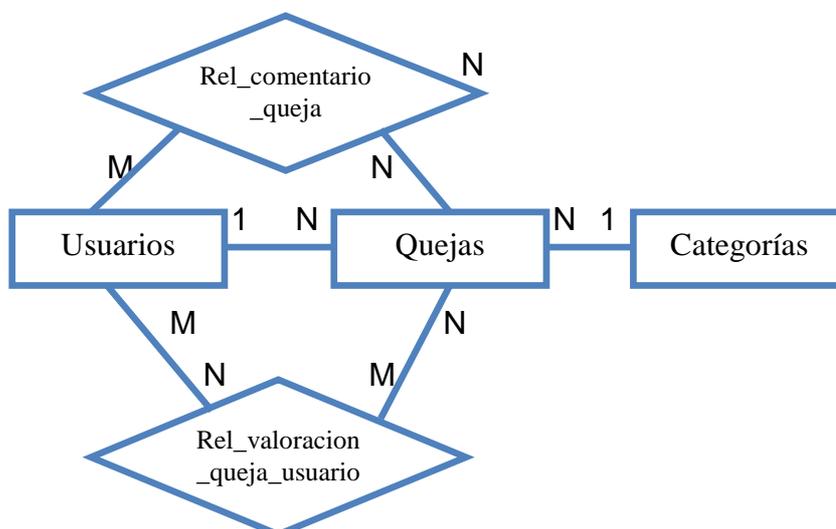


Ilustración 5: Diseño del Modelo Entidad Relación (ER)

Relación tabla “usuarios” con “quejas”: La cardinalidad es Usuarios 1 Quejas N, ya que un usuario puede publicar varias quejas, pero una queja solo puede ser publicada por un usuario.

Para la valoración de una queja se ha creado la tabla “rel_valoracion_queja_usuario”: La cardinalidad es, Usuarios M – N rel_valoracion_queja_usuario M – N queja, con la creación de esta tabla se consigue limitar el número de votaciones a una votación por usuario sobre una misma queja. Ya que solo se le permite valorar una vez la queja.

Para registrar el número de comentarios por queja se ha creado la tabla

“rel_comentario_queja”: La cardinalidad es, comentarios M – N rel_comentario_queja M – N queja, con la creación de esta tabla se consigue obtener el numero de comentarios total que tiene cada queja, para después poder utilizar estos datos en nuestra aplicación.

Relación tabla “categorías” con “quejas”: La cardinalidad es, categorías 1 –M quejas, ya que varias quejas pueden tener una misma categoría, pero una única queja solo puede tener una única categoría.

3.2. Modelo de datos, tablas de la base de datos

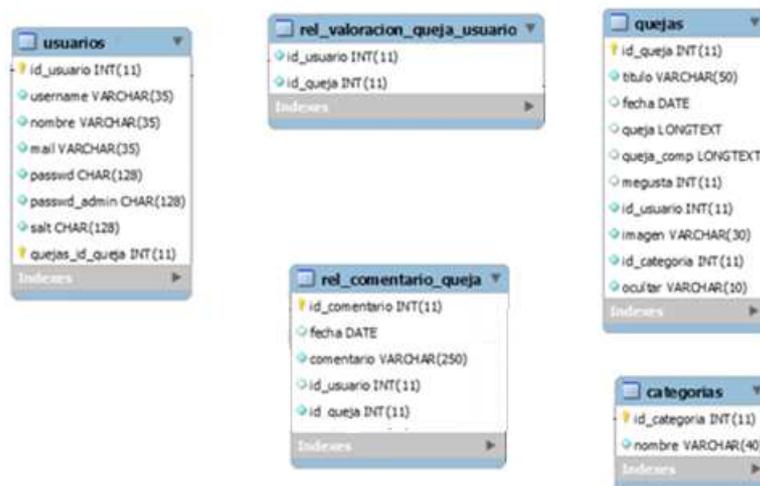


Ilustración 6: Diseño del Modelo de datos

4. Requerimientos generales: software y hardware

4.1. Framework Twitter Bootstrap 3

Tras revisar diversos frameworks de desarrollo web se ha decidido utilizar para la realización del proyecto el framework de Twitter Bootstrap 3.

A modo de ejemplo, a continuación muestro una tabla comparativa entre Bootstrap [3] y otro framework muy adoptado, Foundation [4]:

	Bootstrap 3	Foundation 4
URL	http://getbootstrap.com/	http://foundation.zurb.com/
GRID	4 tamaños diferentes	3 tamaños diferentes + rejilla
BASE CSS	Estilos standard para formularios, botones, imágenes, cabeceras, sistemas de navegación, etc.	Estilos standard para formularios, botones, imágenes, cabeceras, sistemas de navegación, etc.
PRE-PROCESADORES CSS	LESS CSS	SASS AND COMPASS
JQUERY	Dropdown menus, pop-ups, alertas, carrusel de imágenes, paneles, etc	Dropdown menus, pop-ups, alertas, carrusel de imágenes, paneles, etc
ICONOS	Fuente opcional de iconos para incorporar imágenes vectoriales, distribuidas con Bootstrap, 180 iconos	Fuente personalizable de iconos para incorporar imágenes vectoriales, disponibles para descargar por separado
DISTRIBUIDO	Apache 2	MIT

Ilustración 7: Tabla comparativa entre Bootstrap 3 y Foundation 4.
Fuente[28]

Comparación entre pre-procesadores CSS Sass y LESS

- **Instalación:**

Sass[12] funciona con Ruby, los usuarios de Mac no tendrán que instalarlo ya que viene pre-instalado para empezar a utilizarlo, en cambio los usuarios de Windows tendrán que instalar Ruby para poder utilizarlo.

Independientemente del Sistema Operativo, para poder utilizar SASS, hay que hacerlo a través de línea de comandos o bien por terminal. Existen herramientas para facilitar el uso de éste pre-procesador, pero en general tienen un coste.

LESS[12] al estar escrito en Javascript [33] es tan sencillo como importar la librería a tu documento HTML, existen muchas herramientas gratuitas que te ayudan a compilar los archivos.

- **Sintaxis:**

Ambos utilizan la sintaxis estándar de CSS, lo que resulta sencillo convertir una hoja de estilos en formato Sass o LESS.

Sass también acepta ésta sintaxis omitiendo, las llaves “{}” y el punto y coma “,”

```
h1
color:#4D926F
```

Ilustración 8: Sintaxis Sass

```
h1 {
color:#4D926F;
}
```

Ilustración 9: Sintaxis LESS

- **Variables**

Sass:Las variables se asignan con el símbolo de “\$” antes de la variable.

```
$color: #4D926F;
body {
color:$color;
}
```

Ilustración 10; Variables Sass

LESS:Igual que en Sass pero en vez de “\$” hay que agregar el símbolo “@”.

```
@color: #4D926F;

#header {
color: @color;
}
h2 {
color: @color;
}
```

Ilustración 11: Variables LESS

- **Código anidado:**

Podemos escribir los selectores de los hijos dentro del elemento padre y se puede, además, utilizar el símbolo “&” para referirse al mismo.

Tanto Sass como LESS utilizan la misma sintaxis para los elementos anidados:

```
section {
margin: 10px;
nav {
height: 25px;
```

```

a {
  color: #0982C1;
  &:hover {
    text-decoration: underline;
  }
}

```

Ilustración 12: Código anidado Sass/LESS

- **Mixins**

Son parte fundamental de los estilos en LESS CSS y sirven para reutilizar y parametrizar de forma que pueden simplificar las tareas de diseño.

Que se puede hacer con un Mixin:

- Definir un estilo básico y reutilizarlo dentro de toda la hoja de estilos.
- Crear estilos de color.
- Utilizar parámetros para crear estilos.
- Combinar diversas clases para crear una tercera de estilos de un elemento.

En cuanto al uso de los “Mixins”, tanto Sass como LESS son igual de buenas.

```

@mixin error($anchoBorde: 2px) {
  border: $anchoBorde solid #F00;
  color: #F00;
}

.error-generico {
  padding: 20px;
  margin: 4px;
  @include error();
}

.error-login {
  left: 12px;
  position: absolute;
  top: 20px;
  @include error(5px);
}

```

Ilustración 13: Ejemplo de Mixin Sass

```

@mixin error(@anchoBorde: 2px) {
  border: @anchoBorde solid #F00;
  color: #F00;
}

```

```

.error-generico {
  padding: 20px;
  margin: 4px;
  .error();
}

.error-login {
  left: 12px;
  position: absolute;
  top: 20px;
  .error(5px);
}

```

Ilustración 14: Ejemplo de Mixin Less

Salvo la pequeña diferencia en la forma que tienen de aplicar los estilos del mixin error los pre-procesadores Sass(@include mixin_name()) y Less (.mixin_name();), el código compilado es el mismo para los dos:

```

.error-generico {
  padding: 20px;
  margin: 4px;
  border: 2px solid #f00;
  color: #f00;
}

.error-login {
  left: 12px;
  position: absolute;
  top: 20px;
  border: 5px solid #f00;
  color: #f00;
}

```

Ilustración 15: Código compilado ejemplo anterior

- **Herencia**

Hablamos de herencia cuando quieres aplicar el mismo estilo a varios elementos de una hoja de estilos.

La principal diferencia entre LESS y Sass es que en el pre-procesador LESS no genera un código CSS óptimo ya que no agrupa las herencias. En este sentido Sass es mejor que LESS.

```

.bloque, p, ul, ol {
  margin: 10px 5px;
  padding: 2px;
}

```

```
}  
p {  
border: 1px solid #EEE;  
}  
ul, ol {  
color: #333;  
text-transform: uppercase;  
}
```

Ilustración 16: Compilado Sass

```
.bloque {  
margin: 10px 5px;  
padding: 2px;  
}  
p {  
margin: 10px 5px;  
padding: 2px;  
border: 1px solid #EEE;  
}  
ul,  
ol {  
margin: 10px 5px;  
padding: 2px;  
color: #333;  
text-transform: uppercase;  
}
```

Ilustración 17: Compilado LESS

- **Importación:**

La importación en un documento CSS común no es óptima, ya que requiere hacer múltiples peticiones HTTP, lo que supone más lentitud.

En cambio los pre-procesadores CSS cargan los distintos archivos importados y los compilan en un solo archivo.

- **Funciones de colores**

Estas funciones sirven para transformar el color en tiempo de ejecución.

Se pueden utilizar en cualquier lugar donde el color sea una propiedad CSS válida.

```
$color: #0982C1;  
  
h1 {
```

```
background: $color;
border: 3px solid darken($color, 50%);
}
```

Ilustración 18: Colores Sass

```
@color: #0982C1;

h1 {
background: @color;
border: 3px solid darken(@color, 50%);
}
```

Ilustración 19: Colores LESS

- **Operaciones**

Tanto Sass como LESS permiten realizar operaciones matemáticas.

La principal diferencia es que LESS es menos restrictivo en cuanto a permitir operaciones entre diferentes unidades.

- **Media Queries**

Las media queries son la base del “responsive design”. Se componen de un media type y una o más expresiones, las cuales implican características del medio y se resuelven como verdadera o falsa. Los media type permiten indicar los estilos que se aplicarán según el tipo del medio, los más típicos son: all (por defecto), print (vista previa impresión y a la hora de imprimir), screen (pantallas de ordenador), tv (televisores). La consulta de la media query será verdadera, si el tipo de medio concuerda con el tipo de dispositivo que se está mostrando.

```
<link rel="stylesheet" type="text/css" media="all"
href="style.css">
<link rel="stylesheet" type="text/css" media="print"
href="print.css">
```

Ilustración 20: Ejemplo de media type

```
body {
background: green ;
}

@media (max-width: 600px) {
body {
background: red ;
}
}
```

```
}
```

Ilustración 21: Ejemplo de media querie

A diferencia de Less, Sass puede utilizar el comando “respond-to” para llamar a las media queries de una forma más clara.

```
=respond-to($name)

@if $name == pantalla-pequena
  @media only screen and (min-width: 320px)
    @content

@if $name == pantalla-grande
  @media only screen and (min-width: 800px)
    @content
```

Ilustración 22: Comando “respond-to”

Soporte por los navegadores más extendidos:

Bootstrap 3	Foundation 4
Últimas versiones de :	Últimas versiones de Chrome, Firefox, Safari
Chrome (Mac, Windows, IOS y Android)	IOS(Iphone y Ipad)
Safari(Mac y IOS únicamente, Windows da incongruencias)	Android 2, 4 (tablet y móvil)
Firefox(Mac, Windows)	Windows phone 7+
Opera(Mac, Windows)	Surface
IE 9 y 10. IE 8 soportado con respond.js	IE 9 y IE 10. IE 8 soportado con el sistema adicional de “gridcode”

Ilustración 23: Tabla comparativa soporte con navegadores. Fuente [29]

En este aspecto, Twitter Bootstrap gana terreno al dar soporte a una versión más antigua de Internet Explorer, el cual está aún muy presente en muchas empresas en sus versiones 7 y 8.

Bootstrap

Bootstrap es la piedra angular del proyecto, es el framework de Twitter que permite el desarrollo web con CSS y Javascript que adaptan la interfaz dependiendo de las dimensiones y resolución de los diferentes dispositivos en que se visualice.

Sigue un diseño responsivo y “Mobile First”[2].

El concepto “Mobile First” es una filosofía, una forma de facilitarnos la labor durante el diseño responsivo, dando prioridad o empezando siempre por los dispositivos móviles/tablets.

En un móvil no cabe toda la información de la que disponemos y no tiene sentido pretender que un usuario haga un scroll casi infinito para llegar a la información a la que desea acceder.

Por tanto, debemos intentar reducir todo aquello que es superfluo o prescindible.

- **Licencia de Bootstrap**

Bootstrap se publica bajo la licencia "Apache 2 License" y está protegido por el siguiente copyright: "copyright 2013 Twitter".

Ésta licencia te permite descargar y usar gratuitamente Bootstrap entero o alguna de sus partes, tanto para uso personal, empresarial y/o comercial.

También te permite utilizar e incluir Bootstrap en cualquier paquete o distribución de software que crees.

Por otra parte, prohíbe la distribución de cualquier parte sin mencionar quienes son los autores. No permite utilizar ningún elemento de propiedad Twitter de forma que parezca que tiene alguna relación con tu proyecto. Del mismo modo tampoco permite utilizar ningún elemento que parezca que tú has creado ese elemento de Twitter.

Ésta licencia te obliga a incluir una copia de la licencia de Bootstrap en cualquier distribución de software que crees e incluya bootstrap.

Te obliga a indicar de forma clara quienes son los autores de Bootstrap.

A lo que no te obliga la licencia es a incluir en tu software, el código fuente de Bootstrap. No es necesario hacer públicos los cambios y mejoras que hayas hecho de Bootstrap.

El contenido completo de la licencia se encuentra en el repositorio de Github, en el archivo README.

- **Mobile First**

La versión anterior de Bootstrap 3 incluía algunas utilidades para hacer que las páginas se adaptasen a los móviles.

Bootstrap 3 se ha creado desde cero pensando en los móviles.

Para que las páginas se muestren correctamente y funcionen bien en los dispositivos móviles es importante añadir la siguiente etiqueta dentro de la cabecera <head> de todas las páginas:

```
<metaname="viewport"content="width=device-width, initial-scale=1">
```

Ilustración 24: Mobile First

Los dispositivos móviles escalan automáticamente los documentos HTML ocasionando resultados no deseados.

Como se puede apreciar en el código anterior, se utiliza el comando “initial-scale=1” para optimizar el viewport al tamaño de los dispositivos móviles. Con esto aseguramos mayor compatibilidad con mayor cantidad de pantallas y navegadores.

- **Imágenes responsive**

Bootstrap 2 adaptaba automáticamente el tamaño de las imágenes, cosa que no continúa su sucesor. Para mantener el mismo comportamiento de antes, debes añadir la clase `.img-responsive` a cada imagen que quieras que sea responsive.

- **Formularios**

Los formularios son uno de los elementos más importantes para el diseño de aplicaciones web. Bootstrap3 permite diseñar formularios muy variados.

4.2. HTML5 & CSS3

HTML



Este proyecto ha sido realizado con las tecnologías más actuales para la realización de aplicaciones y sitios web.

Al hablar de HTML5, estamos hablando de un lenguaje usado para estructurar y presentar el contenido para web. Con HTML5 los navegadores pueden saber cómo mostrar una determinada web, como están ubicados los elementos, donde ubicar texto, imágenes, etc. En este sentido HTML4 y HTML5 no son tan diferentes, la diferencia principal viene determinada por la sofisticación del código que podemos llegar a construir con HTML5.

Además de la simplicidad de código, HTML5 es soportado de forma nativa por los navegadores y trae consigo una serie de nuevas posibilidades de las cuales podemos mencionar las siguientes:

- **Nueva estructura:**



Ilustración 25: Estructura HTML5

- **Soporte de audio y video**

No hace mucho, la mayor parte del contenido multimedia se transmitía mediante reproductores flash embebidos en nuestro código HTML.

Dependíamos de una tecnología de terceros llamada Flash, se han evidenciado carencias a la hora de personalizar las interfaces en los distintos reproductores Flash y para desarrolladores web no familiarizados con ésta tecnología, resultaba una gran pérdida de tiempo.

Utilizando la tecnología Flash se llegaba a consumir muchos recursos llegando así a bloquear el navegador que se estaba utilizando.

En el 2011 Adobe anunció que no habría más versiones de Flash para móviles y que se pondrían a trabajar en soluciones HTML5.

Desarrollar plugins para móviles es mucho más complicado y requiere de la ayuda de los desarrolladores de sistemas operativos y fabricantes de hardware.

Ahora con HTML5, todo es mucho más sencillo, podemos especificar distintas fuentes de un mismo video, podemos personalizar la interfaz, etc.

Para ello se han creado las dos nuevas etiquetas: audio y video.

- **Almacenamiento en Base de datos Offline**

HTML5 ofrece dos posibilidades a la hora de almacenar datos en el lado del cliente.

LocalStorage: Almacena datos sin tiempo limite

SessionStorage: Almacena datos por lo que dure la sesión

Esto sustituye a las famosas “cokkies”.

- **Tecnología Drag and Drop**

Drag and Drop es una característica muy común. Es cuando se arrastra un objeto de una ubicación a otra diferente.

En HTML5 ésta tecnología se hace por medio de JavaScript.

Ejemplo:

```
<div id="cont0">
  <div id="cont1" ondragover="evdragover(event)" ondrop="evdrop(event)">
    <p draggable="true" ondragstart="evdragstart(event)" id="e1">
      Elementoarrastrable</p>
    <h5 draggable="true" ondragstart="evdragstart(event)" id="e2">
      Otro elemento arrastrable</h5>
    <ul draggable="true" ondragstart="evdragstart(event) id="e3">
      <li>Primer elemento</li>
      <li>Segundo elemento</li>
    </ul>
  </div>
  <div id="cont2" ondragover="evdragover(event)" ondrop="evdrop(event,this)">
    <imgsrc="objetos/espana.gif" alt="españa"
    draggable="true" ondragstart="evdragstart(event)" id="e4"/>
    <imgsrc="objetos/europa.gif" alt="europa"
    draggable="true" ondragstart="evdragstart(event)" id="e5"/>
    <imgsrc="objetos/italia.gif" alt="italia"
    draggable="true" ondragstart="evdragstart(event)" id="e6"/>
  </div>
</div>
```

Ilustración 26: Ejemplo Drag and Drop



CSS3 es la nueva versión del lenguaje CSS, es una ampliación de CSS2, cuenta con algunas mejoras que son demandadas por los diseñadores, entre ellas:

- **Varias imágenes de fondo:** En versiones anteriores a CSS3, se podían poner varias imágenes de fondo utilizando varias capas <div> anidadas, teniendo cada una de ellas su fondo de imagen. Aplicando los estilos correspondientes se logrará que los fondos se superpongan unos a otros.

En CSS3 ya no hace falta recurrir a ésta técnica, ya que podemos aplicarle diferentes imágenes de fondo a un solo elemento.

```
<div id="fondos">
  texto de un único elemento
  ...
</div>
```

Ilustración 27: Ejemplo declaración del <div>

```
<style type="text/css">
#fondos{
  background: url(fondo3.png) bottom right no-
repeat,
  url(fondo2.png) center no-repeat,
  url(fondo1.gif) center repeat;
  width: 300px;
}
</style>
```

Ilustración 28: Ejemplo varias imágenes de fondo

- **Bordes:** CSS3 permite el uso de bordes con esquinas redondeadas, bordes con imágenes, etc.

Con la propiedad "border-radius" se puede definir bordes redondeados en las esquinas, especificando las medidas del radio.

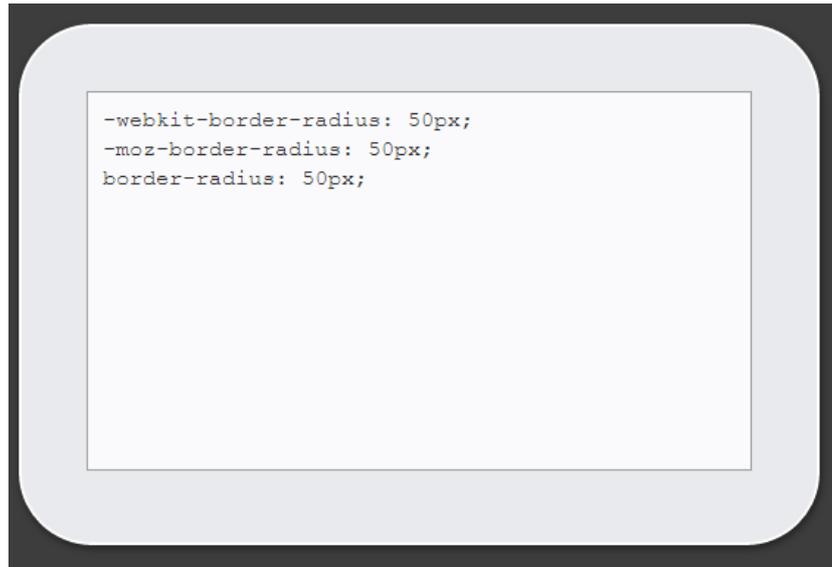


Ilustración 29: Ejemplo de borde redondeado

Con la propiedad “border-image” se puede especificar la imagen que se va a utilizar como borde alrededor del elemento.

```

#borderimg1 {
border: 10px solid transparent;
padding: 15px;
-webkit-border-image: url(border.png) 30 round;
-o-border-image: url(border.png) 30 round;
border-image: url(border.png) 30 round;
}

```

Ilustración 30: Ejemplo de codificación para borde con imagen

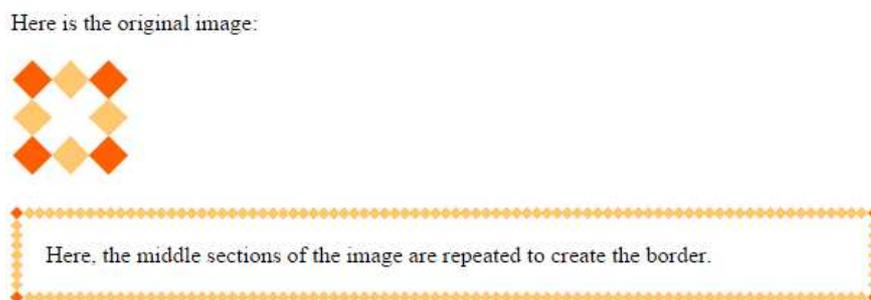


Ilustración 31: Ejemplo de bordes con imagen

- **Sombras:** Con el atributo “box-shadow” se podrá aplicar sombras a los elementos de la página, sin necesidad de utilizar Javascript ni imágenes. Éste atributo necesita cuatro valores para especificar las características de la sombra, son:

Desplazamiento horizontal de la sombra, Desplazamiento vertical, difuminado y color de la sombra.

```
background-color: yellow;
box-shadow: 10px 10px 5px #000000;
```

Ilustración 32: Ejemplo de codificación de caja con sombras



Ilustración: 33 Resultado del ejemplo anterior

- **Transparencias:** Con el atributo “opacity” podemos aplicar transparencias cualquier elemento. El valor por defecto es 1. Como veremos en el ejemplo, el atributo “filter” es para Internet Explorer 8 y anteriores, ya que aún hay incompatibilidades.

```
background-color: red;
opacity: 0.5;
filter: Alpha(opacity=50); /* IE8 and earlier */
```

Ilustración 34: Ejemplo de codificación para aplicar transparencias

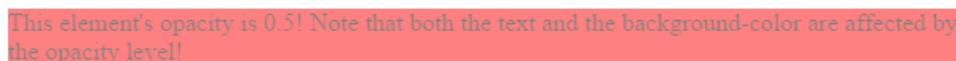


Ilustración 35: Resultado del ejemplo anterior

- **Texto en varias columnas:** Al principio de los tiempos, mucho antes de la aparición de CSS3, para dividir el texto en varias columnas, era necesario el uso de tablas. Uno de los problemas de utilizar tablas, es que, una página maquetada con tablas tarda más en cargar que una página bien maquetada.

Más adelante, con la aparición de CSS2, se abandonó el uso de las tablas para colocar el texto en varias columnas y se sustituyó por div flotantes:

```
<div class="box">
<div class="left">
<div class="content">
Ejemplo de prueba 1
</div>
</div>
<div class="right">
<div class="content">
Ejemplo de prueba 2
</div>
</div>
</div>
```

Ilustración 36: Ejemplo de maquetación con <div> flotante

Solo faltaría aplicar las hojas de estilo correspondientes para darle el formato correcto.

Aún así, este método tampoco sirve, ya que es imposible administrar el contenido, es decir, tienes que ir controlando cuando tienes que realizar el cambio de columna cada vez que se introduce el texto. Éste método tampoco sirve cuando hablamos de multidispositivo, es decir, si se ha pensado en maquetar con una resolución para escritorio, quedará muy mal cuando se navegue desde el móvil o tablet.

La forma correcta de aplicar texto en varias columnas es utilizando CSS3, con las nuevas propiedades column-count, column-gap y column-rule.

- **Column-count:** Con este atributo se indicará el número total de columnas a utilizar. Veamos un ejemplo:

```
<div class="3-columns">
<p> Esto es una prueba con column-count </p>
</div>
```

Ilustración 37: Ejemplo de maquetación

```
.3-columns {
-webkit-column-count: 3; /* Chrome, Safari, Opera */
-moz-column-count: 3; /* Firefox */
column-count: 3;
}
```

Ilustración 38: Ejemplo de aplicación del atributo Column-count



Ilustración 39: Resultado de aplicar column-count

- **Column-gap:** Éste atributo sirve para colocar un margen al texto, es decir, dejar una separación entre las columnas. El valor puede ser en px, pt, em, etc. Lo recomendable es utilizar em, ya que 1 em equivale al tamaño de la fuente del documento por lo que se trata de un valor escalable.

A modo de ejemplo, si se sustituye el código de la ilustración 38 por lo siguiente, obtendremos un espacio de 2em entre las columnas.

```
.3-columns {
-webkit-column-count: 3; /* Chrome, Safari, Opera */
-moz-column-count: 3; /* Firefox */
column-count: 3;
-webkit-column-gap: 20px; /* Chrome, Safari, Opera */
-moz-column-gap: 20px; /* Firefox */
column-gap: 20px;
-webkit-column-gap: 2rem; /* Chrome, Safari, Opera */
-moz-column-gap: 2rem; /* Firefox */
column-gap: 2rem;
}
```

Ilustración 40: Ejemplo de separación de 2em entre columnas



Ilustración 41: Resultado de aplicar 2em de separación

- **Column-rule:** Éste atributo sirve para aplicar un borde separador entre las diferentes columnas, solo hay que añadir lo al código anterior:

```
-webkit-column-rule: 5px solid rgb(75, 180, 149) ; /* Chrome, Safari, Opera */
-moz-column-rule: 5px solid rgb(75, 180, 149) ; /* Firefox */
column-rule: 5px solid rgb(75, 180, 149) ;

-webkit-column-rule: 0.5rem solid rgb(75, 180, 149) ; /* Chrome, Safari, Opera */
-moz-column-rule: 0.5rem solid rgb(75, 180, 149) ; /* Firefox */
column-rule: 0.5rem solid rgb(75, 180, 149) ;

margin: 20px;
margin: 2rem;
```

Ilustración 42: Ejemplo de borde entre columnas



Ilustración 43: Resultado de aplicar column-rule

5. Implementación

5.1. Servidor XAMPP

Para la realización de éste proyecto se ha simulado un entorno real utilizando un entorno virtual.

El entorno real contaría con un dominio y un hosting propio para poder alojar la aplicación web y poder publicarla en internet.

La aplicación se ha desarrollado bajo una arquitectura XAMPP [23]. Es una distribución de Apache[30] gratuita y muy fácil de instalar que consiste básicamente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. Es el entorno más popular de desarrollo con PHP.

El nombre proviene del acrónimo de X “todos los sistemas operativos” Apache, MySQL, PHP, Perl.

Servicios básicos instalados:

Apache: httpd 2.4.12 Released

MySQL: Versión del servidor: 5.6.21 - MySQL Community Server (GPL)

PHP: 5.5.19



Ilustración 44: Servidor XAMPP

5.2. Estructura FrontEnd

Para la realización del FrontEnd se ha utilizado una plantilla gratuita de Bootstrap y se ha adaptado a las necesidades del proyecto.

La estructura de ficheros se compone de:

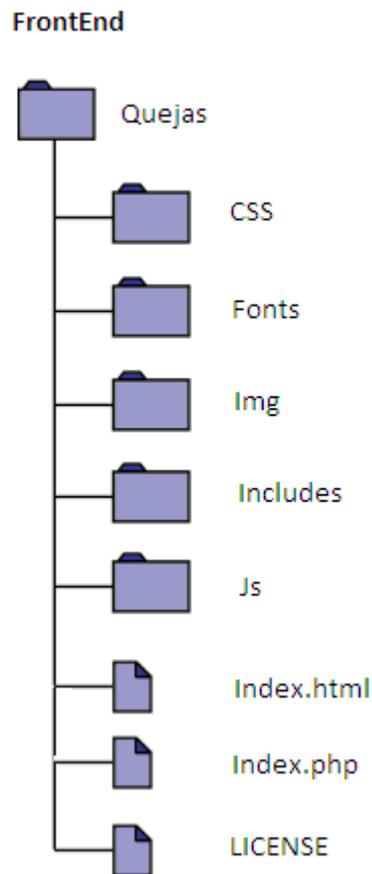


Ilustración 45: Estructura del FrontEnd

La carpeta CSS contiene las diferentes hojas de estilo que se utilizarán para el diseño del FrontEnd.

La carpeta Fonts contiene las fuentes para los Glyphicons de bootstrap.

La carpeta Img contiene las diferentes imágenes utilizadas,

La carpeta Includes, contiene dos librerías llamadas PHPMailer[31] y Mobile-detect[32], donde la primera será la encargada de permitir a la aplicación enviar mails al administrador y la segunda se encargará de detectar si el dispositivo desde el que se navega es un móvil o una tablet. Ésta última librería ha sido utilizada para controlar una serie de impedimentos que daba la aplicación

Ésta carpeta también incluye la lista de archivos PHP y HTML necesarios para el correcto funcionamiento del FrontEnd.

Entre ellos podemos destacar los archivos:

Conexión.php: En este archivo se configura la conexión a la base de datos.

Login.php: Éste archivo mostrará dos formularios, uno para entrar en el sistema y otro para registrarse.

Checklogin.php: En éste archivo se comprueba si el usuario que intenta validarse en el sistema existe. Una vez validado, se comprueba si es administrador de la aplicación web, si lo es, el sistema lo re-dirige al panel de administración o BackEnd, si el usuario no tiene rol de administrador, es re-direccionado a la página inicial de la aplicación web.

Registra_usuario.php: En éste archivo se comprueba si el usuario que quiere registrarse en el sistema ya existe con anterioridad, si existe, será re-dirigido a una página de error indicando que el usuario ya existe, si no existe, será automáticamente dado de alta en el sistema, re-dirigiendo al usuario a una página de bienvenida.

Mail.php: En éste fichero se incluye la clase comentada con anterioridad PHPMailer y se configuran una serie de parámetros necesarios para utilizar el gmail como servidor de correo y poder enviar mails desde la aplicación web.

Entre estos parámetros podemos destacar:

`$correo_emisor:` Correo que se utiliza para autenticarse

`$correo_destino:` Correo de quien recibe

`$mail->Host:` Se establece gmail como servidor SMTP

`$mail->AddReplyTo($correo_emisor, $nombre_emisor):` A que dirección se puede responder el correo.

`$mail->AddAddress($correo_destino, $nombre_destino):` La dirección a donde mandamos el correo.

`$mail->SetFrom($correo_emisor, $nombre_emisor):` De parte de quien es el correo.

`$mail->Send():` Esta función es la encargada de enviar el correo.

Nueva_queja.php: Éste fichero mostrará un formulario para poder enviar las nuevas quejas que tengan los usuarios, utilizando como POST el archivo mail.php explicado anteriormente.

Quejas.php: En éste archivo se mostrarán todas las quejas dadas de alta en el sistema, con el número de votos positivos que tiene cada una. También se podrán filtrar las quejas por categoría.

Queja.php: Éste archivo será el encargado de mostrar la ficha de la queja, ésta ficha contará con el nombre de la queja, la fecha de creación, sistema de valoración, una pequeña introducción de la queja, el contenido de la queja y la lista de comentarios.

Fuera de la carpeta includes se encuentra la carpeta Js que incluye los archivos javascript utilizados para el FrontEnd y los archivos:

Index.html: Este archivo sirve para que el usuario no tenga que escribir toda la ruta entera en el navegador, su función será redireccionar a la página de inicio si el usuario escribe "localhost/quejas".

Index.php: Éste archivo mostrará la página inicial de la aplicación web que contará con una barra de navegación que permitirá al usuario navegar por las diferentes secciones de la web. Las diferentes secciones son:

Inicio: Página inicial de la aplicación.

Login: Formulario para entrar al sistema y formulario para registrarse.

Sobre mí: Ésta sección contendrá información sobre el creador de la aplicación.

Quejas: Ésta sección contendrá las diferentes quejas dadas de alta en el sistema.

Nueva Queja: Formulario para enviar una nueva queja al administrador de la aplicación.

Contacto: Formulario para enviar incidencias o dudas sobre el funcionamiento de la aplicación.

Ésta aplicación también contará con un carousel compuesto de diferentes imágenes y en el cuerpo de la página se encontrará una serie de rankings como: Comentarios por ciudadano, quejas más comentadas y quejas mejor valoradas.

LICENSE: Contiene la licencia de apache, detallando los términos y condiciones de uso, reproducción y distribución. Ésta licencia es la versión 2.0 modificada a Enero de 2004.

5.3. Estructura BackEnd

Para la realización del BackEnd o Panel de Administración, se ha utilizado una plantilla gratuita de Bootstrap y se ha adaptado a las necesidades del proyecto.

La estructura de ficheros se compone de:

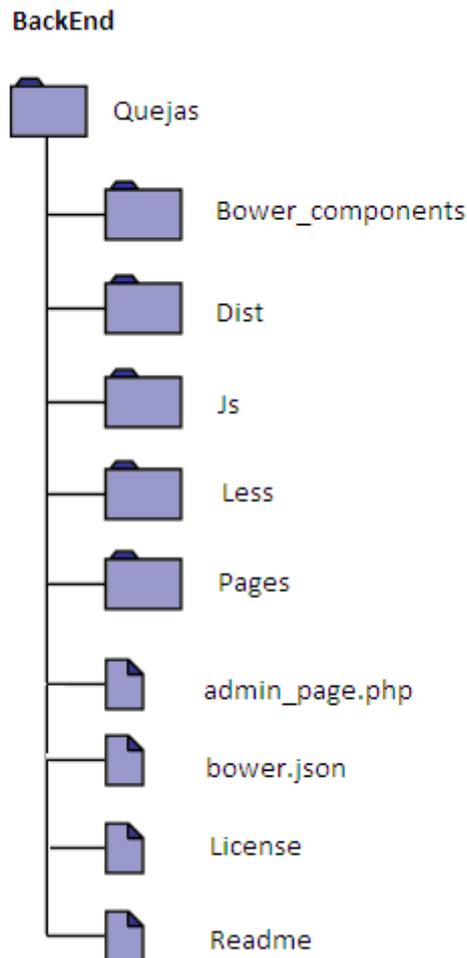


Ilustración 46: Estructura del BackEnd

Las carpetas Bower_components, dist, js, less y los archivos bower.json, license y readme, venían por defecto con la plantilla de Bootstrap

Los archivos que se han creado y son necesarios para la gestión del BackEnd se encuentran en la carpeta Pages, que entre otros contiene los ficheros:

Conexión.php: En este archivo se configura la conexión a la base de datos.

Index.php: Contiene el panel de administrador como tal, para la versión móvil cuenta con una barra de navegación con los contenidos a gestionar, que son los

usuarios, las quejas y las categorías.

También cuenta con una serie de indicadores que muestran el número total de comentarios, las quejas dadas de alta, los usuarios registrados y la lista de las categorías

Categorías.php: En este archivo se encuentra una tabla con el contenido de las categorías, éste contenido está directamente vinculado a la base de datos. En este archivo se pueden realizar tres acciones bien diferenciadas: dar de alta una nueva categoría, modificarla o bien eliminarla.

Quejas.php: Este archivo permite realizar las mismas acciones que “categorías.php” pero para la tabla de quejas.

Usuarios.php: Este archivo permite realizar las mismas acciones que los dos archivos anteriores pero para la tabla de usuarios.

Para cada sección del panel de Administración (Categorías, Quejas y Usuarios) se han creado **siete** tipos de ficheros. A continuación se explicará **un fichero de cada tipo**, para hacer la explicación se utilizará como ejemplo la tabla de los usuarios, hay que repetir éste proceso para la tabla categorías y la tabla quejas.

Nuevo_usuario.php: Contiene el formulario para dar de alta a un nuevo usuario en el sistema.

Guarda_usuario.php: Este archivo es el POST del formulario del archivo anterior, es decir, cuando se envía el formulario, nuevo_usuario.php llama al fichero guarda_usuario.php, en el cual se comprobará si el usuario o el mail ya están siendo utilizados, de ser así, el sistema redirige al usuario a “error_nuevo_usuario.php”, en caso contrario, el usuario será re-dirigido a “ok_nuevo_usuario.php”.

Ok_nuevo_usuario.php: Este archivo muestra la tabla de usuarios notificando que la creación de un nuevo usuario se ha realizado con éxito.

Error_nuevo_usuario.php: Este archivo muestra el formulario de creación del usuario, notificando que la creación del usuario no ha podido realizarse debido a que el usuario o el mail ya existen en el sistema.

Modificar_usuario.php: Contiene el formulario para modificar los datos del usuario que se haya seleccionado. Los campos se rellenan automáticamente para facilitar la modificación al usuario.

Mod_usuario.php: Este archivo es el POST del formulario del archivo anterior, es decir, cuando se envía el formulario, “modificar_usuario.php” llama al fichero “mod_usuario.php”, en el cual se comprobará si el usuario o el mail ya están siendo utilizados, de ser así, el sistema redirige al usuario a “error_mod_usuario.php”, en caso contrario, el usuario será re-dirigido a “ok_mod_usuario.php”.

Eliminar_usuario.php: Este archivo después de confirmación previa del administrador, elimina al usuario que haya seleccionado.

6. Planificación

La planificación del proyecto se ha representado con un diagrama de Gantt.

La planificación se ha hecho pensando en un horario de media jornada.

La duración total estimada del proyecto ha sido de 22 semanas, unos 110 días.

En la ilustración 47 se muestra la planificación inicial que se hizo del proyecto



Ilustración 47: Diagrama de Gantt inicial

Ésta planificación inicial no se pudo cumplir debido a que se había estimado unas franjas muy cortas entre las diferentes tareas a realizar, es decir, si te quedabas estancado en una de ellas, la otra te quedaba pendiente.

Así que finalmente, el tiempo real empleado en el proyecto ha sido el que se describe en la ilustración 48.

7.2.2 Evitar inyección SQL

Los ataques de inyección SQL [15] consisten en introducir textos que modifican una sentencia SQL en los campos de los formularios de la aplicación. Por ejemplo:

```
$query = "SELECT * FROM usuarios WHERE usuario='{$_POST['usuario']}' AND passwd='{$_POST['passwd']}'";  
mysql_query($query);
```

Si no se revisa la cadena de texto que viene por POST, nos podemos encontrar:

```
$_POST['usuario'] = 'sergi';  
$_POST['passwd'] = "' OR ''='";
```

Con esto, un usuario inicia sesión con una contraseña inválida, ya que modificaría la sentencia resultante y quedaría así:

```
SELECT * FROM usuarios WHERE usuario='sergi' AND passwd='' OR ''=''
```

aplicando la función de mysql

```
string mysql_real_escape_string ( string cadena [, resource id_enlace] )
```

a los campos POST del formulario se puede evitar éste problema,

```
$_POST['usuario'] = mysql_real_escape_string($_POST['usuario']);  
$_POST['passwd'] = mysql_real_escape_string($_POST['passwd']);
```

7.2.3 Minimizar el contenido CSS/HTML

Minimizar y unificar el contenido tanto de las CSS como del HTML, para así lograr disminuir el número de “llamadas” a objetos, el tiempo de carga de la aplicación web y, en consecuencia, aumentar la velocidad con la que un usuario navega.

Para lograrlo se utilizará Minify [16], una aplicación PHP5 que combina, minimiza y cachea varios archivos JavaScript y CSS bajo petición para aumentar la velocidad de descarga de las webs.

7.2.4 Realizar las peticiones a la base de datos con Ajax

Con Ajax [17], es posible realizar peticiones a la base de datos y obtener respuesta de esta en segundo plano y usar esos datos para, a través de JavaScript, modificar los contenidos de la página creando efectos dinámicos y rápidos.

7.2.5 Gráficos FrontEnd y BackEnd

Para el FrontEnd, cambiar los ratios que hay actualmente y mostrarlos en forma de gráficos para visualmente mejorar el aspecto de la página inicial de la web.

Para el BackEnd, añadir en el panel de Administración un grafico que muestre el número de comentarios, los usuarios creados, el número de quejas y el número de categorías.

Para la realización de estos gráficos se utilizará la librería de Google “Google Chart Tools” [18]

7.2.6 Realizar pruebas de usuario

Para mejorar la experiencia del usuario y poder ofrecerle una herramienta cómoda y fácil de utilizar se realizarán pruebas de usuarios.

Son pruebas de usabilidad que se basan en la observación y análisis de cómo un grupo de usuarios reales utiliza la aplicación web, anotando los problemas con los que se encuentran para poder solucionarlos posteriormente.

7.2.7 Accesibilidad

Se mejorará la aplicación, haciéndola más accesible para personas de edad avanzada. Para ello se habilitará un acceso especial para ellos mejorando:

- El uso de botones y controles estándares informando en todo momento al usuario de la tarea que está realizando, así no le exige realizar un esfuerzo de memoria.
- Mejorar el tiempo de respuesta.
- Cualquier operación no realizada correctamente, informa al usuario del error que ha cometido.
- Tipo de letra, Arial 14, no se utilizan colores como amarillo, azul y verde juntos.
- No se usan tecnicismos, se emplea un lenguaje claro y familiar para el usuario.
- Posibilidad de seleccionar diferentes idiomas, hay ancianos que entienden mejor el catalán que el castellano.
- En la aplicación web no existen elementos auditivos por lo tanto cualquier usuario con estos problemas, podrá navegar perfectamente.
- Botones, controles y menús grandes

7.2.8 Iniciar sesión con perfiles sociales

Posibilidad de registrarse e iniciar sesión en la aplicación web con una cuenta de Facebook.

7.2.9 Publicación de quejas con Drag & Drop

Permitir al usuario publicar ellos mismos las quejas, utilizando la tecnología Drag & Drop [10] para la subida de la imagen asociada a esa queja. Estas quejas serán posteriormente aceptadas para su publicación en la web

7.2.10 Posicionamiento SEO

Para dar a conocer la aplicación web y mejorar su posicionamiento en los navegadores web utilizaremos la guía sobre optimización para motores de búsqueda. Entre otras, se recomienda:

- **Elegir las palabras clave:** Hay que tener en cuenta el numero de búsquedas de cada termino para poder determinar tus mejores palabras clave. Para ello usaremos Google Trends [19].
- **Comprobar la competencia:** Con Google Adwords [20] se comprobará que palabras son las más asequibles según la popularidad de tu web.
- **Densidad de las palabras:** Medir que palabras se están utilizando mas de nuestra web e intentar repetir más veces las palabras que quieras posicionar bien. Para ello se utilizará Internet Ninja [21]
- **Usar las palabras clave:** Utilizar las palabras clave en los textos, etiquetas y categorías.
- **Palabras clave en negrita y en títulos:** Hay que utilizarlas para crear títulos y a poder ser hacerlas destacar en los artículos para que Google pueda catalogarlo.
- **Medir y analizar el posicionamiento natural para las diferentes palabras clave:** Utilizar CuteBank [22] para consultar la clasificación de las palabras claves de tu aplicación web en los buscadores.

7.2.11 Dinamismo en la sección de categorías

Para evitar realizar el paso de la ilustración 73, se modificará el código PHP para agregarle dinamismo al fichero, evitando así crear un nuevo archivo cada vez que se dé de alta una nueva categoría en el sistema.

8. Bibliografía

- [1] Gardner, Lyza Danger & Grisby, Jason (2012). *Head First Mobile Web*. O'Reilly
- [2] Magno, Alexandre (2013). *Mobile First BootStrap. Develop advanced websites optimized for mobile devices using Mobile First feature of Bootstrap*. Packt publishing
- [3] Bootstrap <<http://getbootstrap.com/>> [Consulta: 02/2015].
- [4] Foundation <<http://foundation.zurb.com/>> [Consulta: 02/2015].
- [5] Dive Into HTML5 <<http://diveintohtml5.info/>> . [Consulta: 03/2015].
- [6] HTML5 Doctor <<http://html5doctor.com/>> [Consulta: 04/2015].
- [7] The expressive web [beta]. Demostración de las nuevas propiedades de HTML5 y CSS3. <<http://beta.theexpressiveweb.com/>> [Consulta: 04/2015].
- [8] HTML5 & CSS3 Readiness <<http://html5readiness.com/>> [Consulta: 05/2015].
- [9] Bootstrap 3, el manual oficial <https://librosweb.es/libro/bootstrap_3/> [Consulta: 03/2015].
- [10] Aprende Web, Crea y diseña tus sitios en la web. Drag & Drop <http://aprende-web.net/NT/html5/html5_7.php> [Consulta: 05/2015].
- [11] Principios heurísticos de Molich y Nielsen. Evaluación Heurística, Universidad de Lleida <<http://aipo.es/libro/pdf/15-Evaluacion-Heuristica.pdf>> [Consulta: 05/2015].
- [12] Comparativa SASS VS LESS <<http://www.wizache.com/blog/sass-vs-less-que-pre-procesador-de-css-es-mejor.html>> [Consulta: 05/2015].
- [13] CSS3 <<https://rolandocaldas.com/html5>> [Consulta: 04/2015].
- [14] Encriptación Sha512 <<http://php.net/manual/es/function.crypt.php>> [Consulta: 06/2015].
- [15] Inyección SQL <<http://php.net/manual/es/security.database.sql-injection.php>> [Consulta: 06/2015].
- [16] Minify <<https://code.google.com/p/minify/>> [Consulta: 06/2015].
- [17] Ajax <<http://api.jquery.com/jquery.ajax/>> [Consulta: 06/2015].
- [18] Google Chart Tools <<https://developers.google.com/chart/>> [Consulta: 06/2015].
- [19] Google Trends <<http://www.google.com/trends/>> [Consulta: 06/2015].
- [20] Google Adwords <<https://adwords.google.com/KeywordPlanner>> [Consulta: 06/2015].
- [21] Internet Ninja <<http://www.internetmarketingninjas.com/seo-tools/keyword-density/>> [Consulta: 06/2015].
- [22] Cute Bank <<http://www.iblogzone.com/go/cuterank-2/>> [Consulta: 06/2015].
- [23] XAMPP <<https://www.apachefriends.org/es/index.html>> [Consulta: 03/2015].
- [24] Start Up <<http://www.todostartups.com/>> [Consulta: 05/2015].
- [25] Insatisfacción del consumidor <<http://www.escueladeservicioalcliente.es/indice-de-insatisfaccion-del-consumidor/>> [Consulta: 02/2015].
- [26] PHP <<http://php.net/>> [Consulta: 03/2015].

[27] MySQL <<https://www.mysql.com/>> [Consulta: 03/2015].

[28] Comparativa entre Twitter Bootstrap y Zurb Foundation <<http://librosweb.es/eventos/web-php-conference-2013/comparativa-entre-twitter-bootstrap-y-zurb-foundation/>> [Consulta: 04/2015].

[29] Soporte con navegadores <<http://responsive.vermilion.com/compare.php>> [Consulta: 04/2015].

[30] Apache <<http://httpd.apache.org/>> [Consulta: 03/2015].

[31] PHPMailer <<https://github.com/PHPMailer/PHPMailer>> [Consulta: 05/2015].

[32] Mobile Detect <<http://mobiledetect.net/>> [Consulta: 05/2015].

[33] JavaScript <<http://librosweb.es/libro/javascript/>> [Consulta: 04/2015].

9. Anexo

9.1 Manual de usuario administrador (BackEnd)

A continuación se incluye un manual para el administrador del BackEnd, donde se detallarán las funciones principales que pueden realizar.

Éste manual está centrado en poder realizar las diferentes funciones desde un dispositivo móvil o tablet.

Acceder al BackEnd: El acceso se realizará desde el formulario login de la página web, utilizando un usuario con privilegios de administrador.

El único usuario que tiene éstos privilegios es “admin” con contraseña “12345”.

Por lo tanto haremos lo siguiente:

Ingresaremos en la página web: <http://localhost/quejas>



Ilustración 49: Ilustración parcial de la página inicial del proyecto

Pulsar en el icono de la derecha para que se abra el menú desplegable.



Ilustración 50: Menú desplegable de la página inicial.

Para poder abrir el formulario de login, pulsar en “Login”. Escribir las credenciales del usuario administrador y enviar el formulario.



Ilustración 51: Formulario de login

Una vez enviado el formulario, el usuario será redirigido al panel de Administración o BackEnd.



Ilustración 52: Página de bienvenida al Panel de Administración

Una vez en el BackEnd, al pulsar en el desplegable, se podrá ver un menú de navegación, donde el administrador puede realizar las siguientes acciones:

Gestión de Usuarios

Al seleccionar “usuarios” del desplegable se abrirá el siguiente menú:



Ilustración 53: Tabla de usuarios

Donde el administrador podrá realizar las siguientes tareas:

Nuevo: Dar de alta un nuevo usuario.



Ilustración 54: Proceso para crear un usuario

A modo de ejemplo, se va a crear el usuario con nombre “manual”, contraseña “manual”, Mail “manual@gmail.com” y pulsar el botón “Nuevo”.

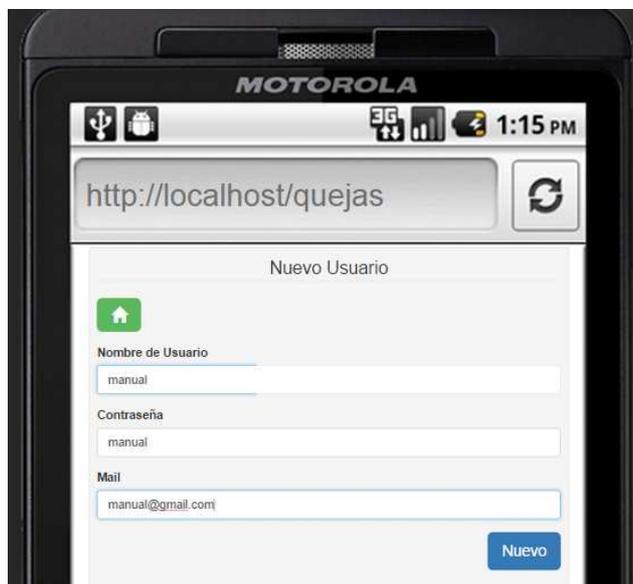


Ilustración 55: Formulario para dar de alta un usuario

Si el usuario no está dado de alta en el sistema, se mostrará el mensaje conforme ha sido creado con éxito. Se muestra en la imagen siguiente.



Ilustración 56: Mensaje de creación satisfactoria.

Modificar: Modificar un usuario existente.



Ilustración 57: Proceso para modificar un usuario.

Los datos se obtienen automáticamente de la base de datos, sin la necesidad de que el administrador tenga que rellenarlos de nuevo. Se puede apreciar que hay

varios campos que no salían en el formulario de la ilustración 55, como “Nombre” donde se escribirá el nombre del usuario, y “Contraseña Administrador”, éste campo habrá que rellenarlo en el caso de querer convertir a éste usuario en administrador. Modificar los campos que quiera y pulsar el botón “Guardar”.



Ilustración 58: Formulario para modificar un usuario

Eliminar: Eliminar un usuario existente.



Ilustración 59: Proceso para eliminar un usuario

Al querer eliminar el registro, el sistema por seguridad, notificará al administrador que va a eliminar ése usuario. Al confirmar el mensaje, se eliminará el registro.

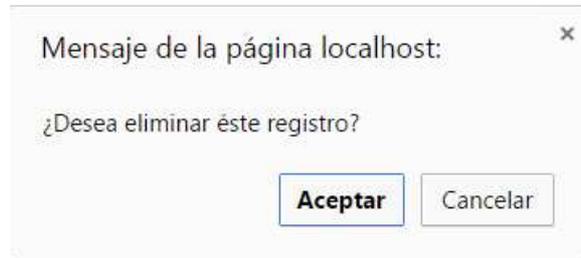


Ilustración 60: Mensaje de confirmación para eliminar el usuario

Gestión de Quejas

El proceso es igual que para un usuario, la diferencia está en el formulario de modificación, para publicar la queja se tendrán que realizar los siguientes pasos:

En la ilustración 61 se muestra la visualización de la tabla "quejas".



Ilustración 61: Tabla quejas.



Ilustración 62: Proceso para modificar una queja.

Los datos se obtienen automáticamente de la base de datos, sin la necesidad de que el administrador tenga que rellenarlos de nuevo. En la ilustración 63 se puede apreciar cómo hay que rellenar los diferentes campos, para poder modificar una queja.

Ilustración 63: Formulario para modificar la queja

Título: Título de la queja

Intro: Texto que servirá como introducción de la queja, se mostrará en la página “quejas”.

Contenido: Texto que servirá para explicar la queja completa, se mostrará en la página de cada queja.

Categoría Actual: Categoría que tiene actualmente asignada la queja

Cambiar por: Categoría a la que queremos asignar la queja.

Ocultar: En el caso de querer ocultar la queja, para que no aparezca en la página web y no puedan verla el resto de usuarios, hay que escribir en este campo la letra “s”. Si el administrador quiere volver a dejar la queja visible para el resto de usuarios, tiene que dejar éste campo en blanco.

Imagen: Se escribe el nombre del fichero de la imagen de la queja.

Si nos dirigimos a la página web y buscamos la queja creada anteriormente, se puede apreciar que no tiene ninguna imagen asignada. Para poder asignarle una imagen a la queja, se tendrá que realizar el paso siguiente:

La imagen tendrá que guardarse en la carpeta del proyecto “quejas/img” con el mismo nombre que se le asigne en el campo que se ha rellenado del formulario anteriormente, llamado “imagen” y con la extensión de imagen “.png”. Tal y como puede apreciarse en la ilustración 64.

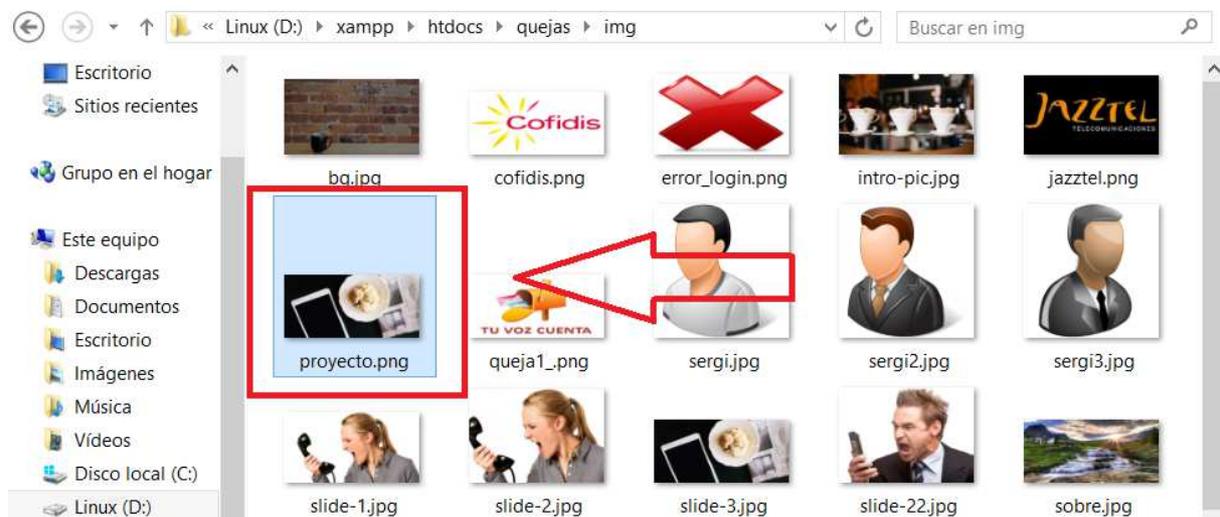


Ilustración 64: Como guardar la imagen correctamente.

En la ilustración 65 se puede ver que la queja se ha creado correctamente.

En la ilustración 66 se puede ver la queja publicada en la web, se aplica el filtro por categorías seleccionando “Agencias de viajes” tal y como se ha modificado en la ilustración 63.



Ilustración 65: Filtro aplicado para la visualización de la queja modificada



Ilustración 66: Visualización de la queja modificada en la página web.

Gestión de Categorías

Nuevo: Dar de alta una nueva categoría

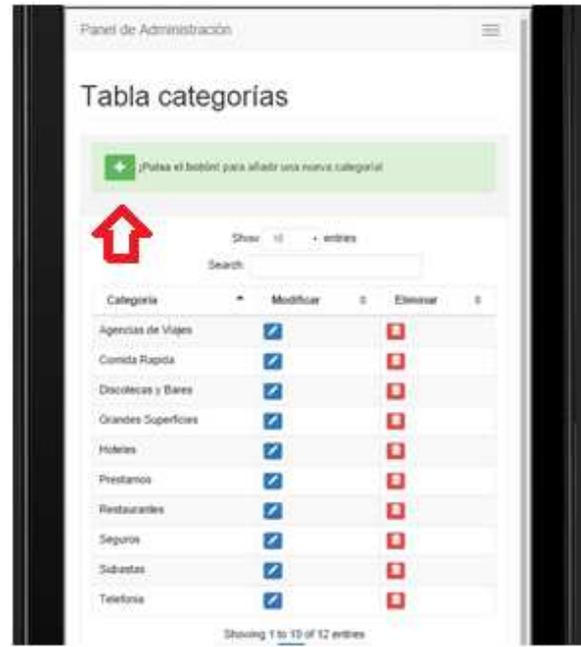


Ilustración 67: Proceso para crear una categoría



Ilustración 68: Creación de la categoría "proyecto"



Ilustración 69: Creación realizada con éxito

Modificar: Modificar una categoría existente.

Los datos se obtienen automáticamente de la base de datos, sin la necesidad de que el administrador tenga que rellenarlos de nuevo.

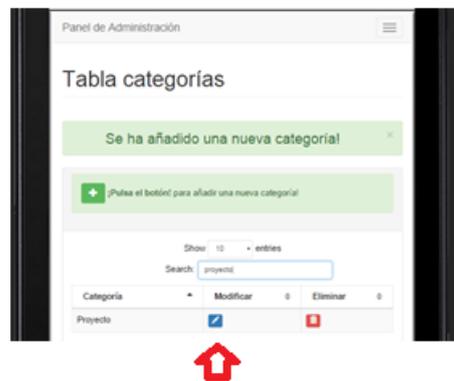


Ilustración 70: Proceso para modificar una categoría



Ilustración 71: Modificación de la categoría “proyecto” por “proyectoModificado”

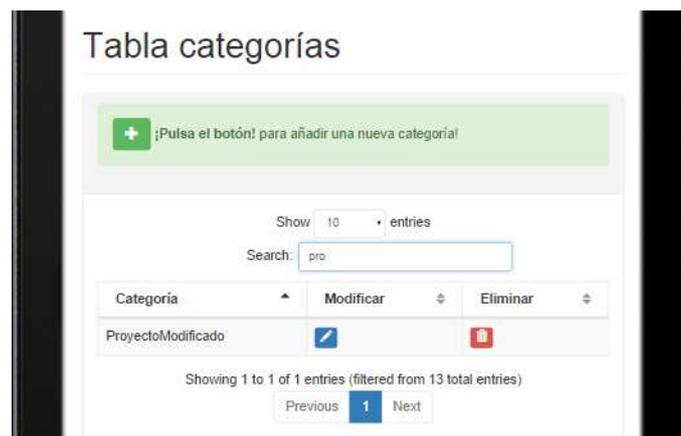


Ilustración 72: Modificación realizada con éxito

Una vez creada la categoría, el siguiente paso es copiar el archivo “Agencias de Viajes” y pegarlo con el nombre de la categoría creada.Éste paso se podrá evitar una vez se haya realizado el trabajo futuro explicado en el punto 7.2.11

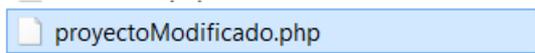


Ilustración 73: Nuevo archivo creado

Volvemos a la tabla “quejas” y cambiamos la categoría de la queja llamada “proyecto” por la nueva categoría “proyectoModificado”

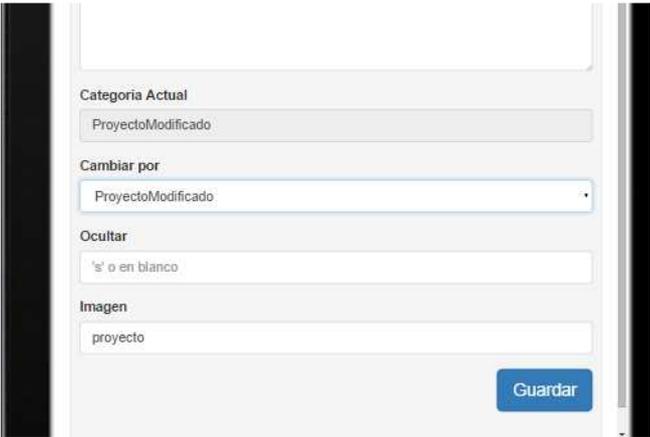


Ilustración 74: Cambio de categoría por “proyectoModificado”

Como se puede apreciar en la ilustración 75, si se filtra por la categoría “ProyectoModificado” aparece la nueva queja.



Ilustración 75: Visualización de la queja con la nueva categoría

Eliminar: Eliminar una categoría existente



Ilustración 76: Proceso para eliminar una categoría

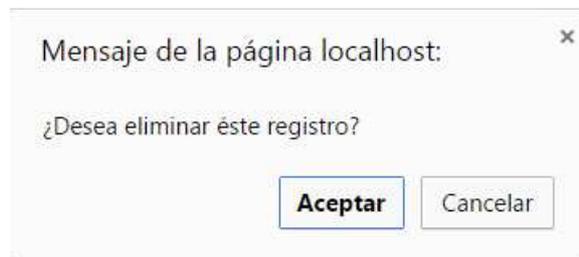


Ilustración 77: Mensaje de confirmación para eliminar el usuario



Ilustración 78: Visualización en la página web de la categoría eliminada

9.2 Prototipo Inicial

Para el diseño de la página web se realizó un prototipo inicial de baja definición que permitió revisar los diferentes casos de uso.

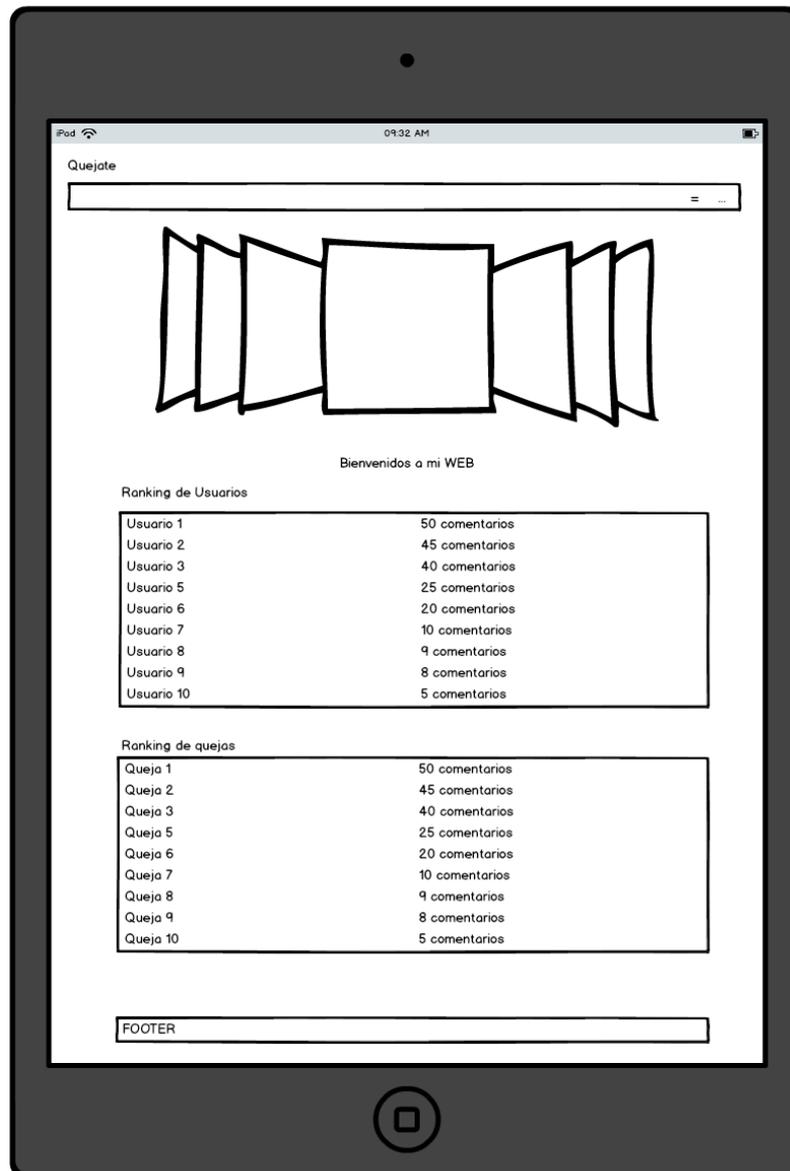


Ilustración 79: Prototipo página de inicio

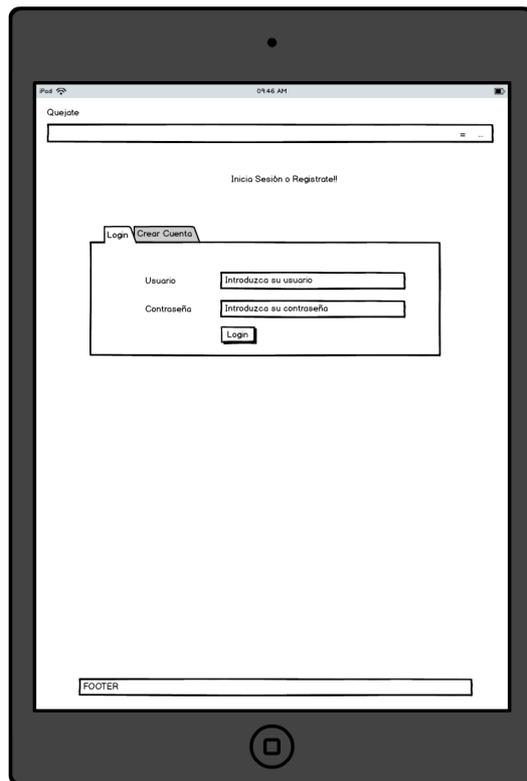


Ilustración 80: Prototipo página de login

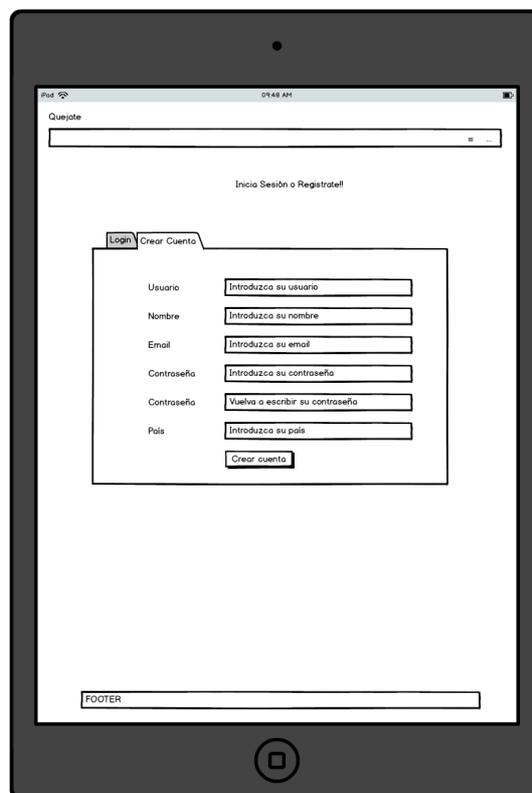


Ilustración 81: Prototipo página "registro"

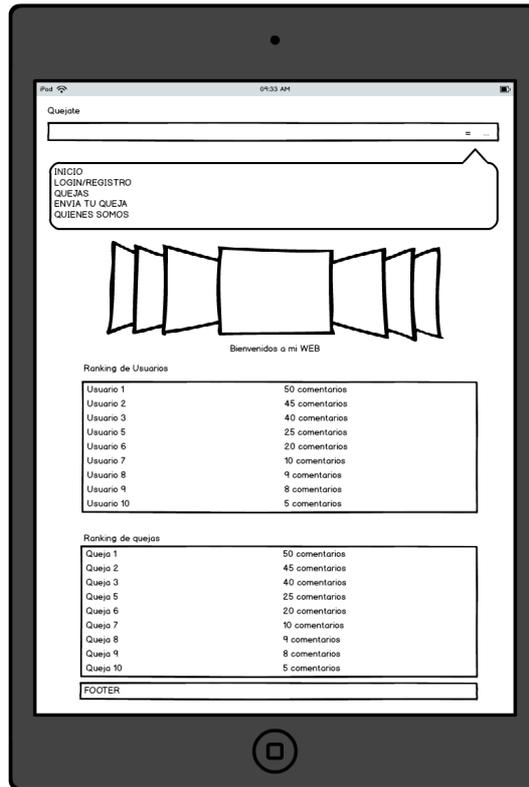


Ilustración 82: Prototipo del menú desplegable

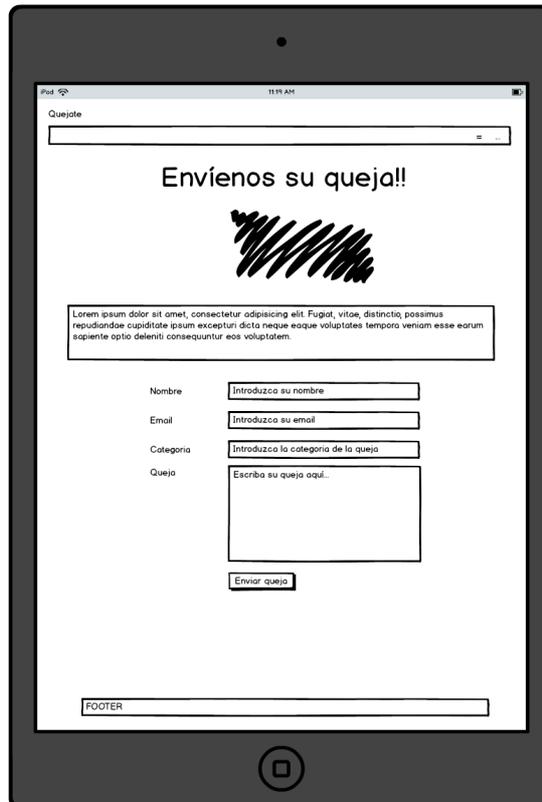


Ilustración 83: Prototipo formulario “nueva queja”

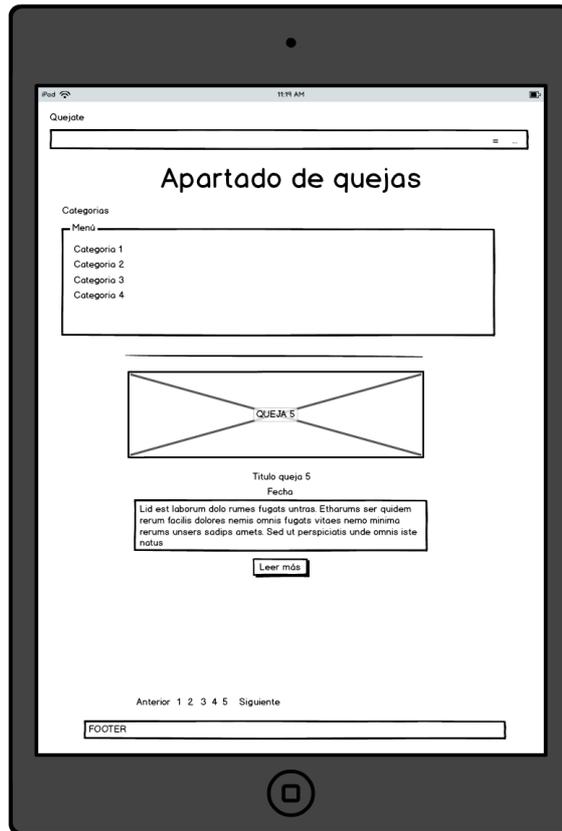


Ilustración 84: Prototipo para el filtrado de las quejas por categorías

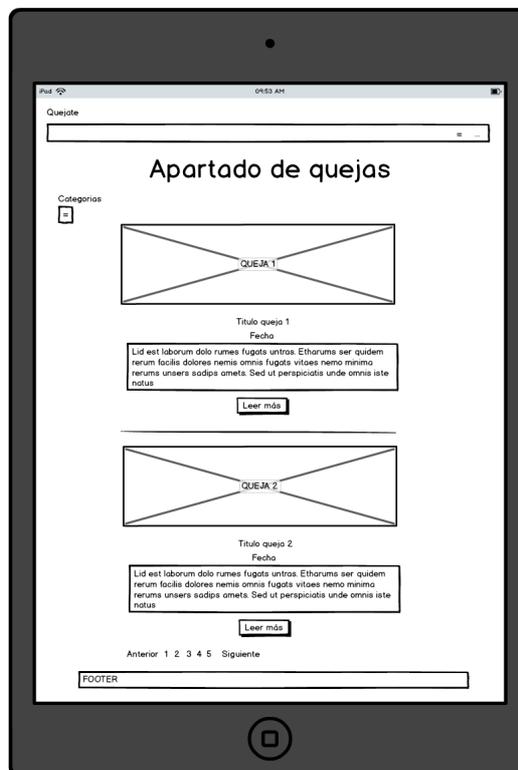


Ilustración 85: Prototipo página de las quejas

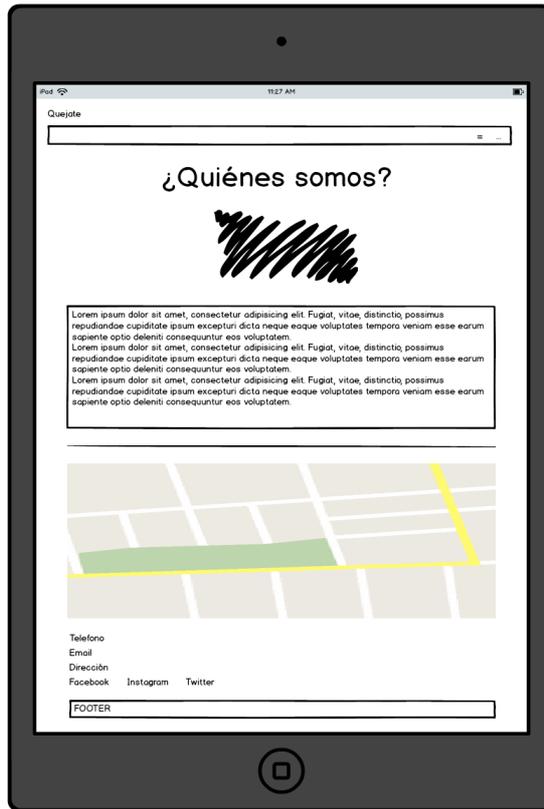


Ilustración 86: Prototipo página “quiénes somos”