



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

## **Buzón Ciudadano**

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Enrique Carrillo Cármenes

**Tutor:** César Ferri Ramírez

2014-15



*If war is the violent resolution of conflict,  
then peace is not the absence of conflict,  
but rather,  
the ability to resolve conflict  
without violence.*

*C.T. Butler*





# Resumen

El objetivo del proyecto es implementar un buzón de sugerencias de participación ciudadana que permita a los miembros o usuarios de una determinada empresa, institución u organización expresar públicamente los asuntos que les preocupen. Un asunto puede ser tanto una queja como una sugerencia y otros usuarios podrán apoyarla, aumentando de esta forma su visibilidad en el buzón y la presión social para que la autoridad competente tome medidas al respecto. La aplicación será única por empresa, institución, organización o municipio (en el caso de ayuntamientos) y solo los miembros de éstas podrán participar. Un punto importante es dar la posibilidad a las autoridades competentes de publicar comentarios, iniciando así un ciclo de conversación entre ambas partes. La finalidad es dar la posibilidad a las autoridades competentes de ser más transparentes con sus vecinos, usuarios, clientes... y a éstos un canal para comunicarse con dichos responsables.

**Palabras clave:** Buzón de sugerencias, Aplicación web, Ruby on Rails, Open Source, Open Data.

# Abstract

The main target of this project is to implement a suggestion box for citizen participation which allows to the members or users of a particular company, institution or organization to express openly the issues they care about. These issues could be suggestions, complaints or congratulations and other users could support it, increasing its visibility on the suggestion box and also social pressure, so the competent authority could take action on it. The application will be unique on each company, institution, organization or town hall and only their members could participate on it. A important point of the application is to allow the competent authority to post comments, which allows to start a conversation cycle with both users and the competent authority. The aim is giving the possibility to the competent authority to be more transparent with their neighbors, users, customers... and give a communication channel with these responsible persons to the last ones.

**Keywords:** Suggestion box, Web application, Ruby on Rails, Open Source, Open Data.





# Tabla de contenidos

Capítulo 1. Introducción.....	13
1.1 Contexto.....	13
1.2 Motivación.....	14
1.3 Objetivo.....	14
Capítulo 2. Buzones de sugerencias tradicionales.....	15
Capítulo 3. Características.....	18
3.1 Anónimo.....	18
3.2 Social.....	18
3.3 Accesible y fácil de utilizar.....	18
3.4 Interfaz de usuario minimalista.....	19
3.5 Internacionalizado.....	19
3.6 Sistema de autenticación sencillo.....	19
3.7 Sistema de lista blanca.....	19
3.8 Autogestionado.....	19
Capítulo 4. Metodología de Trabajo.....	21
4.1 Personas.....	21
4.2 Historias de usuario.....	23
4.3 Kanban.....	24
4.4 Reuniones semanales.....	25
Capítulo 5. Tecnologías utilizadas.....	26
5.1 Ruby.....	26



5.2 Ruby on Rails.....	26
5.3 RSPEC.....	26
5.4 HTML.....	27
5.5 CSS.....	27
5.6 Twitter Bootstrap.....	27
5.7 JQuery.....	27
5.8 Ajax.....	27
5.9 SQLite.....	28
5.10 PostgreSQL.....	28
5.11 Git.....	28
5.12 Heroku.....	28
Capítulo 6. Arquitectura MVC.....	29
6.1 Modelo.....	30
6.2 Vista.....	30
6.3 Controlador.....	30
6.4 Módulos de Ruby On Rails.....	31
6.4.1 Action Pack.....	31
6.4.1.1 Action Dispatch.....	31
6.4.1.2 Action Controller.....	31
6.4.1.3 Action View.....	31
6.4.2 Action Mailer.....	32
6.4.3 Active Model.....	32
6.4.4 Active Record.....	32
6.4.5 Active Resource.....	32
6.4.6 Active Support.....	32
6.4.7 Railties.....	32
Capítulo 7. Trabajo realizado.....	33
7.1 Historias de usuario implementadas.....	33
7.1.1 Ver sugerencias.....	33

7.1.2 Crear sugerencias.....	34
7.1.3 Ficha de sugerencia.....	34
7.1.4 Estadísticas de sugerencia.....	35
7.1.5 Comentar sugerencias.....	35
7.1.6 Ficha de sugerencia Antonia.....	36
7.1.7 Validación nueva sugerencia.....	36
7.1.8 Validación de comentario.....	37
7.1.9 Ficha de sugerencia Juan.....	37
7.1.10 Paginar las sugerencias de la página principal.....	38
7.1.11 Rediseñar formulario de comentarios.....	38
7.1.12 Feature tests.....	39
7.1.13 Internacionalización del buzón.....	39
7.1.14 Firmar sugerencias.....	40
7.1.15 Actividad para los firmantes.....	40
7.1.16 Test unitarios.....	41
7.1.17 Rol ayuntamiento destacado.....	41
7.1.18 Punto de vista sobre la sugerencia.....	42
7.1.19 Filtrar sugerencias.....	42
7.1.20 Mostrar etiqueta con la dirección de la sugerencia.....	43
7.1.21 Añadir tipo de sugerencia.....	43
7.1.22 Modificar sugerencia.....	44
7.1.23 Eliminar comentario.....	44
7.1.24 Eliminar sugerencia.....	45
7.1.25 Inactividad de sugerencia.....	45
7.1.26 Concluir sugerencia.....	46
7.1.27 Diseño general de la web.....	46
7.1.28 Añadir Ajax al filtro de búsqueda y al paginador.....	47
7.1.29 Reportar sugerencia.....	47

7.1.30 Reportar comentario.....	48
7.2 Historias de usuario sin implementar.....	48
7.2.1 Reportar vecino.....	48
7.2.2 Archivar sugerencias.....	49
7.2.3 Compartir sugerencia.....	49
7.2.4 Sugerir títulos.....	49
7.2.5 Firmantes pueden subir imágenes.....	49
7.2.6 Mapa con todas las sugerencias.....	50
7.2.7 Feed de actividad.....	50
7.2.8 Limitar numero de intentos al editar o eliminar.....	50
Capítulo 8. Tipos de usuario.....	51
Capítulo 9. Interfaz de usuario.....	52
9.1 Elementos comunes.....	52
9.1.1 Cabecera.....	52
9.1.2 Pie de página.....	53
9.2 Página Principal.....	53
9.3 Alta y edición de sugerencias.....	54
9.4 Ver sugerencia.....	57
9.5 Publicar comentario.....	59
Capítulo 10. Sistema de autenticación.....	60
10.1 Acción de consultar.....	61
10.2 Acción de publicar.....	61
10.3 Acción de editar o eliminar.....	62
Capítulo 11. Autogestión del Buzón.....	64
11.1 Conclusión de las sugerencias.....	64
Capítulo 12. Pruebas automáticas.....	66
12.1 Pruebas unitarias.....	66
12.2 Integración continua.....	66
12.2.1 Travis CI.....	66

Capítulo 13. Instancia concreta: Buzón UPV.....	68
13.1 Funcionamiento del Buzón UPV.....	68
13.1.1 Viendo las sugerencias publicadas.....	68
13.1.2 Comentando/firmando sugerencias como alumno.....	72
13.1.3 Comentando sugerencias como personal de la UPV.....	73
13.1.4 Eliminando comentarios.....	75
13.1.5 Publicando una sugerencia.....	77
13.1.6 Editando una sugerencia.....	81
13.1.7 Eliminando sugerencias.....	84
13.1.8 Reportando sugerencias y comentarios.....	86
Capítulo 14. Conclusiones.....	87
Capítulo 15. Trabajo futuro.....	88
Bibliografía.....	90
Anexo A. Modelo de datos.....	91
Anexo B. Estructura del proyecto.....	92
Anexo C. Buzón Ciudadano en producción.....	94
Anexo D. Configuración y puesta en marcha.....	96
D.1 Consideraciones iniciales.....	96
D.2 Preparando el entorno de desarrollo.....	96
D.2.1 Instalar RVM y Ruby.....	96
D.2.2 Instalar Ruby on Rails 4.2.....	97
D.3 Despliegue de la aplicación.....	98
D.3.1 Descargar el Buzón Ciudadano.....	98
D.3.2 Ejecución y puesta en marcha.....	98
D.3.3 Recibir emails en local.....	98
D.3.4 Direcciones de correo electrónico.....	99





# Capítulo 1. Introducción

El objetivo del presente trabajo de fin de grado es el desarrollo de un buzón de sugerencias de participación ciudadana destinado a instituciones pública, principalmente ayuntamientos, aunque puede ser adoptado por cualquier tipo de empresa, institución u organización. Durante el desarrollo de la aplicación y la redacción de la memoria se ha utilizado el ejemplo del ayuntamiento por ser la temática del TFG y el ejemplo más interesante de buzón de sugerencias de participación ciudadana. Sin embargo, en el capítulo 13 veremos un ejemplo concreto de un supuesto *Buzón Ciudadano* adaptado a la Universidad Politécnica de Valencia y puesto a disposición de la comunidad universitaria.

El *Buzón Ciudadano* pretende ser una evolución hacia un modelo más social y transparente del típico buzón de sugerencias que todos conocemos y hemos utilizado en alguna ocasión, renovando de esta forma el concepto de buzón de sugerencias tradicional que tienen las personas. Queremos que la gente no vean los buzones de sugerencias como simples cajas negras dónde introducen sus sugerencias y ya está, sino que queremos que la gente vea a los buzones de sugerencias como lugares dónde además de publicar sugerencias (quejas y felicitaciones) también puedan comentarlas ofreciendo su punto de vista y crear de esta manera un debate que ayude a las autoridades responsables a tomar decisiones en base a la opinión de la gente. Además, damos la oportunidad a los órganos de gobierno de poder participar el debate iniciando así un ciclo de conversación con el que se pretende, en el mejor de los casos, intentar llegar a un consenso entre ambas partes.

La diferencia principal de este buzón de sugerencias con los ya existentes es que añade una componente social de forma que cualquier persona puede ver las sugerencias del resto de usuarios y comentar en ellas como si de un foro o de una red social se tratase. Con esto conseguimos que los usuarios se involucren con los temas que les importan y se abra un ciclo de conversación con las autoridades responsables de la institución que hay detrás del buzón.

El *Buzón Ciudadano* se basa en el framework Ruby on Rails y para su desarrollo solo se han utilizado tecnologías Open Source para que su utilización no suponga un gasto para la empresa, institución u organización que quiera adoptarlo.

## 1.1 Contexto

Una de las mayores preocupaciones de los ciudadanos es la gestión que llevan a cabo sus ayuntamientos. Todos los ciudadanos pagamos unos impuestos para que las administraciones públicas dispongan de recursos para financiar las necesidades públicas de los ciudadanos. ¿Pero hasta qué punto están al tanto los ayuntamientos de las necesidades y de las preocupaciones de sus ciudadanos?



Si bien es cierto que los ayuntamientos disponen de mecanismos para que los ciudadanos puedan ponerse en contacto con ellos, como oficinas de atención al público, buzones de sugerencias, direcciones de correo electrónico... Su gestión interna no permite a los ciudadanos saber en qué estado se encuentran sus peticiones, si han sido tenidas en cuenta o en qué plazos piensa el ayuntamiento actuar. De esta falta de transparencia nace el *Buzón Ciudadano*, que pretende ser un canal público de comunicación entre los vecinos y los órganos de gobierno.

La idea principal es que en el *Buzón Ciudadano* solo puedan participar los vecinos de una determinada localidad, municipio o pueblo, de forma que la extensión geográfica del mismo permanezca acotada y los responsables gubernamentales puedan saber qué es lo que ocurre en su municipio a través de sus propios vecinos. Evidentemente la adopción del *Buzón Ciudadano* como mecanismo intermediario entre los vecinos y el ayuntamiento no implica que el ayuntamiento deba solucionar o atender todas las peticiones de los vecinos, es simplemente un mecanismo para que el ayuntamiento pueda conocer de primera mano las preocupaciones de sus vecinos y pueda tomar las medidas oportunas cuando lo crea necesario.

## 1.2 Motivación

El motivo principal por el que elegí este TFG fue el hecho de desarrollar una aplicación que desde mi punto de vista es muy interesante y novedosa, además de por su carácter público y social. Siempre he pensado que el software debe ayudar a las personas haciendo su día a día más cómodo, y ¿qué mejor forma de ayudar a las personas que proporcionarles un canal totalmente transparente de comunicación con los gobernantes de su ciudad?

Otro de los motivos fue el hecho de desarrollar una aplicación web con tecnologías totalmente desconocidas para mí y que me han supuesto un gran reto. Uno de los trabajos más demandados en el mundo laboral es el de desarrollador web, disciplina en la que considero que no he recibido suficiente formación en el Grado en Ingeniería Informática de la UPV y motivo por el cual pensé que era una buena oportunidad para formarme como desarrollador web y aprender nuevas tecnologías y metodologías de trabajo que espero utilizar al finalizar la carrera.

## 1.3 Objetivo

El objetivo del presente trabajo de fin de grado es desarrollar un buzón de sugerencias que permita a los órganos de gobierno ser más transparentes con los vecinos y a los vecinos un canal para comunicarse con las autoridades de forma pública.

Uno de los principales problemas de las administraciones públicas es la falta de dinero a causa de las deudas que han ido contrayendo a lo largo de los años, por lo que en el desarrollo del *Buzón Ciudadano* solo se han utilizado tecnologías *Open Source* para que su implantación no suponga ningún coste económico.

Además se va a intentar que el *Buzón Ciudadano* sea lo más autónomo posible, de forma que necesite el mínimo mantenimiento posible y además no sea necesario contar con un administrador que esté pendiente de moderar o administrar el buzón.

## Capítulo 2. Buzones de sugerencias tradicionales

Un buzón de sugerencias es un mecanismo que se pone a disposición de los usuarios para que puedan realizar sugerencias, quejas o felicitaciones sobre una determinada empresa, institución o cualquier tipo de organización. La finalidad de este mecanismo es que el interesado en conocer la opinión de sus usuarios pueda tener *feedback* directo sobre las preocupaciones de éstos y poder tomar medidas en consecuencia.

La mayoría de los buzones que se utilizan actualmente son, o bien simples webs de información con una dirección de correo electrónico de contacto. Véase la página web de la Escuela Técnica Superior de Arquitectura de la UPV (Figura 1).



Figura 1: Buzón de sugerencia de la ETSA

O bien se tratan de formularios algo más elaborados que permiten recoger la información de una forma más estructurada. Véase el buzón de sugerencias del ayuntamiento de Valencia (Figura 2).

10/06/2015 ☀ Máx.: 23.8°C Min.: 22.3°C | @ Correo ciudadano | Valenciano | Sugerecias | Mapa web | Home

AYUNTAMIENTO | TRÁMITES Y GESTIONES | WEBS MUNICIPALES | LA CIUDAD | NOTICIAS

**Valencia**

ATENCION CIUDADANA  
INSERCIÓN 010 MUNICIPAL

**Inicio**  
Teléfono 010  
Atención presencial  
Página web  
Quioscos de trámites y servicios  
Aplicaciones móviles  
Quejas y reclamaciones  
**Consultas y Sugerecias**  
L'Ajuntament Informa  
Servicios RSS

**AJUNTAMENT DE VALÈNCIA**  
Pl. de l'Ajuntament, 1  
46002 València  
Tel.: 96 3525478

**Sugerecias**  
Si desea realizar una sugerencia sobre cualquier actividad municipal o sobre la ciudad de Valencia, envíenos su mensaje. No olvide indicar su e-mail si desea recibir nuestra contestación.

Nombre:

Apellidos:

Dirección:

Población:

Provincia:

Código Postal:

País:

Teléfono:

Correo electrónico:

Repetir correo electrónico:

Tema de la sugerencia:

Sugerencia:\*

**enviar**

AYUNTAMIENTO | TRÁMITES Y GESTIONES | WEBS MUNICIPALES | LA CIUDAD | INFOCIUDAD | NOTICIAS | AGENDA | MAPA WEB | AVISO LEGAL |

W3C WAI-AA WCAG 1.0

Figura 2: Buzón de sugerencias del ayuntamiento de Valencia

En ambos casos las sugerencias van a parar a una “caja negra” donde esperan a que el encargado del buzón las vea y decida qué hacer con ellas o transmita dicha información a las personas con poder para tomar decisiones. Lo normal en los buzones de sugerencias que implementan la mayoría de empresas, instituciones u organizaciones es que el usuario que ha enviado la sugerencia pierda su pista una vez ésta ha sido enviada y no reciba nunca ningún tipo de información acerca de la misma, ni para saber si ha sido leída ni para saber si se va a tomar algún tipo de medida.

Quizá el buzón de sugerencias más completo que hemos encontrado sea el buzón ciudadano de la Generalitat Valenciana (<http://www.buzonciudadano.gva.es>). Este buzón de sugerencias está dividido en municipios y dispone de un listado con las propuestas realizadas por los vecinos para que cualquiera pueda acceder a ellas y verlas, siendo mucho más transparente que los buzones de sugerencias más tradicionales que hemos comentado antes.

The screenshot shows the 'Buzón Ciudadano' interface. At the top, there are navigation links for 'Inicio', 'Objetivos', 'Reglamento', and 'Buscador'. Below this, a breadcrumb trail reads 'Está en : Inicio > Valencia Carcaixent Propuestas'. The main content area is titled 'Propuestas' and contains a table of suggestions. To the left, there are several sidebar widgets: 'Municipios' (listing Castellón, Alicante, Valencia), 'Menú del usuario' (with links for Cuenta and Participa), 'Acceso Participante' (with a 'Salir' button), 'Acceso Ayuntamientos', and '¿Quién está en línea?' (showing 26 invited and 1 connected member). To the right, there are widgets for 'Promociones' (SOM Online Magazin), '¿CÓMO PARTICIPAR?', 'Índice Carcaixent' (with links for Propuestas, Consultas, Agradecimientos, Informes), and a 'Calendario Carcaixent' for July 2015.

Fecha	Título	Estado
Domingo, 15 de Enero de 2012	<a href="#">control del trafico y estacionamientos</a>	Contestada
Martes, 03 de Enero de 2012	<a href="#">velocidad en la via</a>	Contestada
Jueves, 15 de Diciembre de 2011	<a href="#">Calle Na Violant d'Honorria</a>	Contestada
Jueves, 15 de Diciembre de 2011	<a href="#">Pasos peatonales en confluencia</a>	Contestada
Jueves, 01 de Diciembre de 2011	<a href="#">abocadors ilegals</a>	Contestada
Miércoles, 30 de Noviembre de 2011	<a href="#">Noleja als carrers</a>	Contestada
Miércoles, 08 de Junio de 2011	<a href="#">Concierto</a>	En estudio

Figura 3: Buzón ciudadano Generalitat Valenciana

En este buzón de sugerencias las propuestas realizadas por los vecinos son cerradas al igual que en los buzones de sugerencia tradicionales y nadie salvo el ayuntamiento puede contestar al interesado.

A pesar de que las propuestas de los vecinos son públicas, el hecho de no permitir al resto de vecinos aportar su punto de vista en las propuestas de otros vecinos le resta interés al buzón ciudadano propuesto por la Generalitat Valenciana.

This screenshot shows a detailed view of a proposal. The breadcrumb trail is 'Está en : Inicio > Valencia Carcaixent Propuestas control del trafico y estacionamientos'. The proposal is dated 'Domingo, 05 de Febrero de 2012 01:38'. The text of the proposal discusses traffic control at Ave Maria and requests a 'CORTADA EL TRAFICO A LAS HORAS DE ENTRADA Y SALIDA DE LOS ALUMNOS'. Below the proposal, there is a response from the system: 'En primer lugar queremos agradecerle su sugerencia y contribución a la mejora de la ciudad a través de este medio de participación ciudadana. Respecto a su comentario, le informamos que el corte de la circulación en los horarios de entrada y salida no es un procedimiento habitual en el control del tráfico en los centros educativos de Carcaixent. No obstante, pasamos su comentario tanto al área de Interior como a la Policía Local, con la finalidad que se adopten las medidas que se consideren oportunas para mejorar la seguridad de los escolares en esa zona. Gracias por su participación y sugerencia.' At the bottom, there are 'Me gusta' and 'No me gusta' buttons, and the status is 'Estado de la propuesta: Contestada'.

Figura 4: Propuesta de un vecino



## Capítulo 3. Características

El *Buzón Ciudadano* se componen de varias características que lo hacen único dentro de los buzones de sugerencias tradicionales, como su carácter social, su autogestión o su enfoque de lista blanca. También hemos querido fortalecer otras características en las que otros buzones de sugerencia no prestan demasiada atención, como son la sencillez de uso o su interfaz minimalista y adaptada a cualquier tipo de usuario.

### 3.1 Anónimo

Al tratarse de un buzón de sugerencias de acceso público hemos querido que ante todo prevaleciese el anonimato de las personas que lo usan, evitando así que se pueda asociar una sugerencia o comentario con su autor. El único dato de carácter personal que se le solicita al usuario para poder utilizar el *Buzón Ciudadano* es una dirección de correo electrónico válida para poder identificarlo ante el sistema y llevar a cabo las validaciones oportunas, dejando a decisión del usuario la utilización o no de su nombre real cuando va a publicar nuevos comentarios y/o sugerencias.

### 3.2 Social

Quizá una de las diferencias principales del *Buzón Ciudadano* con respecto al resto de buzones de sugerencias más tradicionales es su carácter social. Está pensado para que todos los asuntos que se traten en él sean públicos y además fomenta la participación ciudadana al permitir que cualquier persona pueda compartir su opinión en las sugerencias publicadas por otros vecinos.

### 3.3 Accesible y fácil de utilizar

Una de las características principales del *Buzón Ciudadano* es que se ha construido pensando en cuales serán nuestros futuros usuarios, intentando de esta forma adaptarlo a cualquier tipo de persona. Hoy en día prácticamente todo el mundo tiene al menos un dispositivo con acceso a Internet en su casa y sabe desenvolverse por la red de redes. Sin embargo, hay muchas personas mayores que apenas saben utilizar esta tecnología, por lo que no queríamos que esto fuese un impedimento a la hora de usar el *Buzón Ciudadano* y se ha construido pensando en la accesibilidad y la facilidad de uso.

### 3.4 Interfaz de usuario minimalista

El *buzón Ciudadano* está pensado para que cualquier persona pueda utilizarlo desde el primer momento sin necesidad de un periodo de aprendizaje previo, por eso era imprescindible que la interfaz de usuario fuera lo más minimalista y sencilla posible. Cada ventana tiene los elementos justos para que el usuario pueda interactuar con la misma sin ningún tipo de distracción. Además, se ha optado por que todos los elementos de la interfaz de usuario fuesen lo suficientemente grandes y visibles para que se viesen a simple vista.

### 3.5 Internacionalizado

Hoy en día la internacionalización de los productos software es algo que hay que tener muy en cuenta, sobre todo si queremos que nuestras aplicaciones sean utilizadas por personas de cualquier parte del mundo. El *Buzón Ciudadano* está pensado para ser multi idioma desde su concepción, lo cual ha hecho que desde sus inicios se haya pensado en la internacionalización del mismo y su abstracción de cualquier idioma. Con esto se ha conseguido que traducir el *Buzón Ciudadano* a cualquier idioma sea una tarea trivial y solo sea necesario traducir un simple fichero de texto que contiene todos los textos de la aplicación.

Para el presente TFG se ha optado por traducir la aplicación al castellano y al inglés, pero añadir una traducción a cualquier otro idioma solo le llevaría unos pocos minutos a cualquier traductor y no tendría que tocar ni una sola línea del código fuente de la aplicación.

### 3.6 Sistema de autenticación sencillo

Debido a que el *Buzón Ciudadano* puede ser utilizado por personas de todas las edades y con cualquier conocimiento informático, se descartó en seguida la idea de utilizar el sistema de inicio de sesión clásico como método de acceso. El sistema de autenticación por el que se ha optado se basa en el enfoque *One Less Password*, que permite a los usuarios acceder al sistema utilizando únicamente su dirección de correo electrónico. No hay usuario ni contraseña que recordar. En el capítulo 10 se explica en detalle el enfoque concreto que se ha seguido para el *Buzón Ciudadano*.

### 3.7 Sistema de lista blanca

El enfoque *One Less Password* que se ha seguido ofrece un acceso sencillo al *Buzón Ciudadano*, sin embargo puede resultar un poco molesto para los usuarios el tener que validar o confirmar cada acción que quieran realizar a través de los mensajes de correo electrónico que el sistema les envía a su cuenta. Para evitarlo se ha implementado un sistema de lista blanca, el cual almacena todas las direcciones de correo electrónico que ya han sido validadas anteriormente para evitar volver a enviar mensajes de validación a los usuarios cuando quieren volver a publicar una sugerencia o un comentario. De esta forma se consigue un uso más fluido del buzón para los usuarios que lo utilizan habitualmente. Para la edición o el borrado si que se requiere la confirmación del usuario para evitar la edición o el borrado fraudulento por una tercera persona.

### 3.8 Autogestionado

Debido a que el *Buzón Ciudadano* está dirigido a que los ayuntamientos lo usen como fuente de información para saber que es lo que le preocupa a los vecinos, no queríamos que una “mano” malintencionada pudiese eliminar o editar sugerencias o comentarios a su antojo para distorsionar la realidad. Por este motivo el *Buzón Ciudadano* se autogestiona de forma que no



necesita a una persona con el rol de administrador con poder sobre él. Una vez el *Buzón Ciudadano* es puesto en funcionamiento no requiere de la intervención de personas para su funcionamiento salvo para casos muy concretos y que se explicarán más adelante.

# Capítulo 4. Metodología de Trabajo

Para la realización del presente trabajo de fin de grado se ha optado por utilizar las metodologías ágiles frente a las metodologías más tradicionales, dejando de lado las largas jornadas de análisis, prototipado, definición de casos de uso, diagramas... Para centrarnos en las necesidades reales y cambiantes del desarrollo de un proyecto software.

Durante la realización del proyecto se ha utilizado un desarrollo iterativo e incremental en el que en cada iteración obtenemos un incremento de funcionalidad sobre el producto a desarrollar. La idea era dividir el desarrollo del proyecto en pequeñas tareas que se irán agrupando en bloques que representan nuevas funcionalidades a desarrollar. Para llevar un control del trabajo a realizar se han utilizado las siguientes herramientas.

## 4.1 Personas

El primer paso antes de empezar a definir los requisitos software es pensar en todos los posibles usuarios que va a tener nuestra aplicación para poder ponernos en la piel de éstos y adaptar los requisitos software a nuestros usuarios. Como el *Buzón Ciudadano* va a estar dirigido a personas de todas las edades se han definido cuatro roles muy distintos entre si. Por una parte tenemos a Antonia que representa a las personas de la tercera edad, Juan, que representa a los adolescentes, Ana que representa al personal del ayuntamiento y a las personas de media edad y por último tenemos a Victoria, la alcaldesa.

Para poder ponernos en la piel de estos futuros usuarios de nuestra aplicación necesitamos ponerles cara y describirlos como si de personas reales se trataran, ya que la idea es poder ponernos en la piel de ellos para poder detectar funcionalidades especiales o adaptar la aplicación a determinados tipos de personas, que sin ponernos en la piel de las mismas seguramente pasaríamos por alto.

En las siguientes tablas podemos ver las personas definidas así como una pequeña descripción sobre las mismas.



Antonia	Juan
	
<p>Antonia es una ex profesora de instituto ya jubilada y a la que a sus 70 años le encanta dar largos paseos por el parque y echar de comer a las palomas que andan revoloteando por allí.</p> <p>Entre sus aficiones se encuentra el baile de salón y viajar al extranjero, cosas que hace siempre que sus dos mascotas, Timmy un chihuahua y Rodolfo su gato le permiten.</p> <p>Antonia nunca ha usado un PC, pero desde que le tocó en la rifa del mercado un smartphone de última generación se pasa horas cotilleando la vida de sus vecinas en Facebook. Tiene pensado comprarse un ordenador portátil y ya se ha apuntado a un curso de informática que dan en el ayuntamiento de su pueblo.</p>	<p>Juan es un adolescente de 17 años a punto de acabar el instituto al que le encanta salir con sus amigos y la música electrónica. Su gran pasión son los ordenadores y tiene pensado estudiar Ingeniería Informática cuando termine el instituto.</p> <p>Juan se pasa horas jugando juegos online con sus amigos, especialmente MMORPG y FPS y en sus ratos libres administra un foro de robótica del que es copropietario junto con dos amigos suyos.</p>

Tabla 1: Personas

Ana	Victoria
	
<p>Ana es una chica de 28 años alegre y risueña que sueña con viajar por todo el mundo para conocer nuevas culturas. Actualmente trabaja como administrativa en el ayuntamiento de su pueblo y la han encomendado la supervisión del <i>Buzón Ciudadano</i> para transmitir las preocupaciones de los vecinos a los responsables del ayuntamiento.</p> <p>Ana es voluntaria de varias asociaciones en su pueblo y odia las injusticias, por lo que aceptó el cargo de supervisora del <i>Buzón Ciudadano</i> sin pensarlo. Ana haría cualquier cosa por ayudar a sus vecinos.</p>	<p>Victoria tiene 45 años, es licenciada en derecho por la Universidad de Valencia y desde hace 2 años es la alcaldesa de su pueblo. Es aficionada al tiro con arco y a los dardos.</p> <p>Aunque Victoria no es muy partidaria de las nuevas tecnologías, se ha propuesto actualizar su ayuntamiento para hacerlo más social y que los vecinos de su pueblo puedan estar al día de lo que pasa en su localidad y puedan compartir con el ayuntamiento sus opiniones.</p>

Tabla 2: Personas

Aunque pueda parecer una tontería el hecho de definir roles de nuestros futuros usuarios, lo cierto es que es un proceso que nos ayuda mucho a la hora de pensar y definir los requisitos de software de nuestra aplicación, y que nos obliga a pensar en el uso que le van a dar nuestros usuarios a la aplicación y qué tipo de problemas se van a poder encontrar. De esta forma podemos descubrir nuevas funcionalidades o simplemente adaptarlas de forma que cualquier tipo de usuario pueda utilizarlas.

## 4.2 Historias de usuario

Las historias de usuario representan requisitos de software que son descritos en una o dos frases utilizando el lenguaje natural. Son una forma rápida de administrar los requisitos software sin la necesidad de elaborar documentos formales ni de administrarlos con complejas herramientas como ocurre en las metodologías tradicionales.

Las Historias de usuario deben ser independientes unas de otras, negociables, estimables, pequeñas y verificables. Con esto se consigue que los requisitos software se puedan desarrollar rápidamente, no necesitan mantenimiento, permite dividir los proyectos en pequeñas entregas, permite estimar fácilmente su esfuerzo y son ideales para proyectos con requisitos en constante evolución.

Para la descripción de cada una de las historias de usuario se utiliza la siguiente plantilla:

Como <Persona/Rol>  
Quiero <Funcionalidad>  
Para <Beneficio>

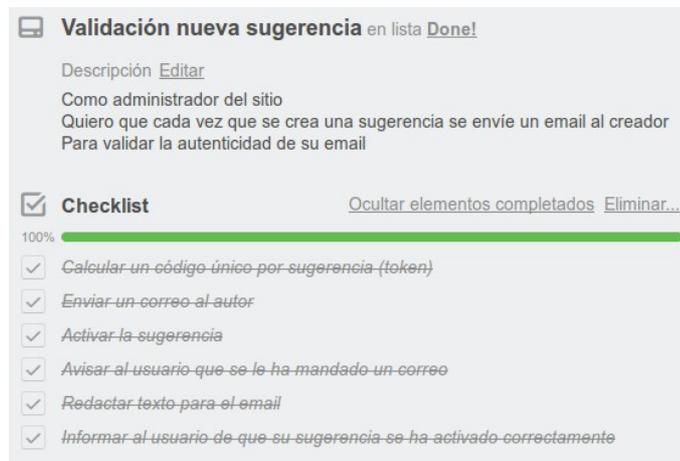


Figura 5: Historia de usuario

Con esto conseguimos resumir en unas pocas palabras el requisito software a implementar y además es entendible por cualquier persona, lo cual es muy importante en el caso de que trabajemos directamente con el cliente o con personas que no se dediquen al desarrollo de software y no estén familiarizados con otras herramientas más técnicas, como puede ser los diagramas UML (diagrama de clases, de casos de uso, de interacción...).

## 4.3 Kanban

El Kanban es un sistema de información gráfica que nos permite controlar la fabricación o desarrollo de un producto software de forma que sabemos en cada momento en qué estado se encuentra cada tarea, quién la tiene asignada y qué tenemos acabado.

El Kanban es un tablero que se encuentra dividido en columnas que representan cada una de las actividades por las que irán pasando todos nuestros artefactos software, que estarán representados por tarjetas (cada tarjeta es una historia de usuario). El número de columnas va a depender de lo que queramos especializar cada una de las actividades a realizar. En mi caso he preferido utilizar actividades más generales para simplificar el proceso:

- **Backlog:** Es la columna inicial y es dónde se van a ir almacenando todas las tarjetas con las funcionalidades a desarrollar para nuestra aplicación. No es necesario detallarlas en este punto, con un simple título descriptivo de la tarea a realizar es suficiente. Es importante mantener las tarjetas ordenada por prioridad para evitar desarrollar tareas poco importantes al principio o para llevar un control de hasta dónde vamos a desarrollar para la entrega de la siguiente versión del producto.
- **Prepared Stories:** Para cada nuevo incremento del producto que estamos desarrollando se cogen varias tarjetas del backlog, se terminan de completar añadiendo la descripción de la misma y las tareas a realizar para su implementación y se pasan a esta columna a la espera de ser cogidas por alguno de los desarrolladores. En esta columna tendríamos las tarjetas ya preparadas (de ahí el nombre de *prepared stories*) y listas para empezar su desarrollo.
- **Doing:** En esta columna se encuentran las tareas que actualmente están en desarrollo, ya sea en la fase de programación o testeado. Mirando esta columna podemos ver que tareas se están desarrollando actualmente y las personas de nuestro equipo que están actualmente en la fase de desarrollo.
- **Done:** Es el punto final de nuestro desarrollo. Aquí se irán dejando las tarjetas de las tareas que hayan sido finalizadas y es el lugar al que acudiremos para ver el trabajo acabado.

En la siguiente imagen se puede ver el estado del Kanban utilizado en un momento determinado del desarrollo de la aplicación:

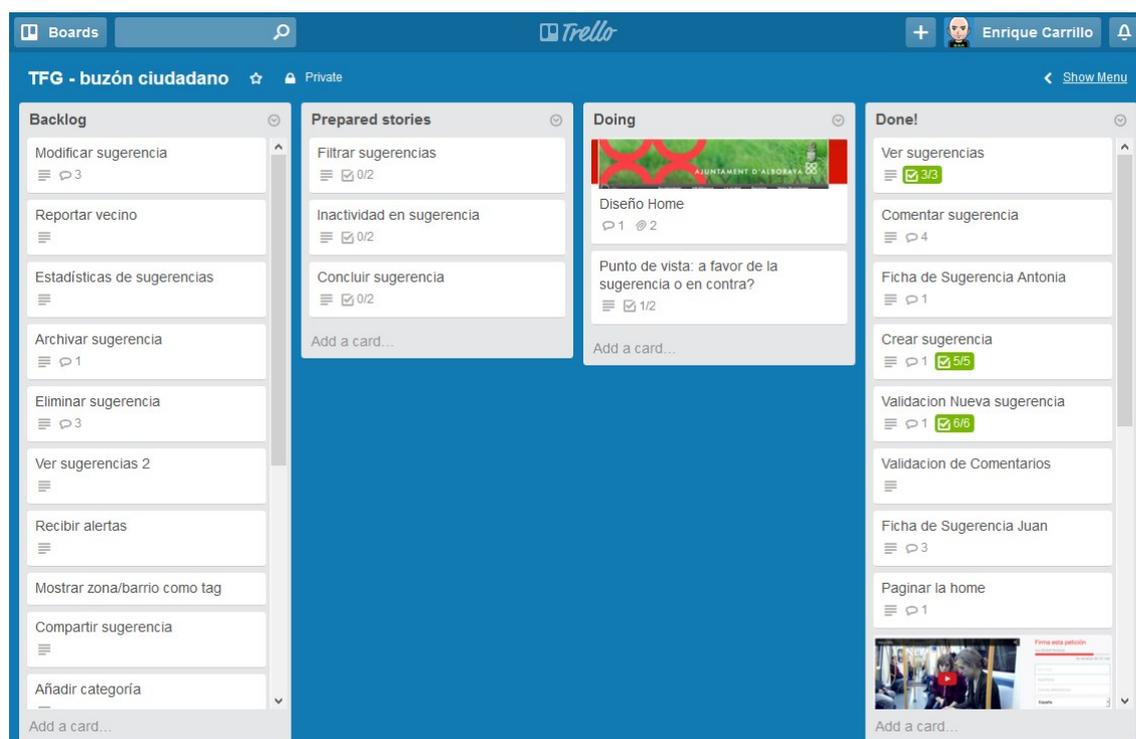


Figura 6: Tablero Kanban

## 4.4 Reuniones semanales

Una de las tareas habituales de cualquier desarrollo ágil son las reuniones diarias, como las *daily meetings* de Scrum, dónde el equipo de desarrollo se reúne para que cada uno de los miembros del equipo informe al resto sobre que ha hecho desde la última reunión, que va a hacer a partir de ese momento y que impedimentos o riesgo ha tenido o va a tener para cumplir con las tareas asignadas. La idea es que todo el equipo esté informado de lo que ocurre dentro del proyecto y del estado del mismo a través de sus propios compañeros.

En mi caso las reuniones han sido semanales debido a la intermitencia durante la semana a la hora llevar a cabo las tareas asignadas. Aun así la idea es la misma, cada semana se ha llevado a cabo un seguimiento por parte de mi tutor dónde se explicaba el trabajo realizado desde la última reunión, se preparaban las tarjetas pendientes del Kanban para realizar hasta la siguiente reunión y se hablaba de los problemas que habían surgido durante el desarrollo, cómo se han enfrentado y cuál ha sido el resultado final.



## Capítulo 5. Tecnologías utilizadas

Como ya se ha comentado anteriormente, todas las tecnologías utilizadas para el desarrollo del *Buzón Ciudadano* son tecnologías de código abierto por las que no hay que pagar ningún tipo de licencia ni para uso personal ni comercial. Además somos totalmente libres de leer, modificar y distribuir su código fuente sin permiso de sus propietarios, lo cual fomenta su desarrollo, evolución y corrección de errores.

Aunque el framework principal que se ha utilizado para el *Buzón Ciudadano* ha sido Ruby on Rails en su versión 4.2, también se han utilizado otras tecnologías adicionales para suplir o complementar a este último y dotar a la aplicación de mayor funcionalidad.

A continuación se va a describir brevemente todas las tecnologías utilizadas en el desarrollo del *Buzón Ciudadano*.

### 5.1 Ruby

Ruby es el lenguaje principal que se ha utilizado para el desarrollo del *Buzón Ciudadano* y es el lenguaje de programación en el que se basa el framework Ruby on Rails.

Ruby es un lenguaje de programación orientado a objetos desarrollado por el programador japonés Yukihiro “Matz” Matsumoto y que fue lanzado en el año 1995. Una de las características principales de Ruby es su sintaxis, que está más próxima al lenguaje natural (en inglés) que al resto de lenguajes de programación. Lo cual hace que sea fácil de escribir y de leer además de fácil de aprender.

### 5.2 Ruby on Rails

Ruby on Rails, abreviado Rails o RoR, es un framework para el desarrollo de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby. Es multiplataforma y sigue el paradigma de arquitectura Modelo Vista Controlador (MVC). Ruby on Rails trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones escribiendo menos código que con otros frameworks y con un mínimo de configuración. Además, Ruby on Rails se enfoca en la productividad del programador y en el desarrollo web ágil.

### 5.3 RSPEC

RSPEC es un framework de Ruby orientado al desarrollo dirigido por comportamientos (*behavior-driven development* o también abreviado *BDD*). RSPEC nos permite definir con una sintaxis clara el comportamiento que esperamos de ciertas partes de nuestro código antes de

empezar a programarlas y nos va a guiar durante el proceso de programación para conseguir que las partes de nuestro código hagan lo que en realidad deben hacer y no lo que se supone que deberían de hacer. Para ello se usan de una forma exhaustiva las pruebas de código (*Testing* en inglés).

## 5.4 HTML

HTML (*HyperText Markup Language*) es un lenguaje de marcado para la elaboración de páginas web. Se trata de un estándar de la W3C (*World Wide Web Consortium*) y actualmente es el lenguaje utilizado por defecto para estructurar el contenido de las páginas web.

## 5.5 CSS

CSS (*Cascading Style Sheets*) es un lenguaje utilizado para definir y crear estilos para formatear nuestros documentos HTML y dotarlos de mayor vistosidad. Con CSS podemos alterar las fuentes, colores, márgenes, posición... de cualquiera de los elementos HTML de nuestra web.

Como el diseño de la interfaz de una página web es un tema complejo y requiere de mucha experiencia, he optado por utilizar el framework *Twitter Bootstrap* que facilita mucho el trabajo al tener una gran variedad de clases CSS con estilos ya predefinidos.

## 5.6 Twitter Bootstrap

Es un framework de software libre para diseños de sitios y aplicaciones web. Este framework nos permite crear interfaces web con CSS y JavaScript que adaptan su interfaz al tamaño del dispositivo en el que se visualizan de forma automática y sin apenas configuración. Esto es lo que se conoce como diseño adaptativo o *Responsive Design*.

Como el diseño web puede llegar a ser muy complejo si no tienes las aptitudes necesarias, he optado por utilizar este framework que me ha facilitado mucho las cosas. El hecho de tener una gran variedad de clases CSS predefinidas y ser totalmente adaptativo ha hecho que me pudiese centrar en la programación y dejar un poco de lado el diseño web sin que por ello se viera perjudicado el mismo.

## 5.7 JQuery

JQuery es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones... Con JQuery se ha dotado al *Buzón Ciudadano* de ciertos efectos, animaciones y ciertas funcionalidades que hacen más agradable el uso de la aplicación.

## 5.8 Ajax

AJAX (*Asynchronous JavaScript And XML*), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.



## 5.9 SQLite

SQLite es una librería software que implementa un motor transaccional de base de datos SQL totalmente autónomo, que no necesita de un servidor para funcionar y que no precisa de ningún tipo de configuración para su utilización. SQLite es uno de los motores de base de datos más utilizados debido a su facilidad de uso y portabilidad, ya que la base de datos física no es más que un fichero de texto.

SQLite es el sistema de base de datos que incluye y utiliza por defecto Ruby on Rails para el entorno de desarrollo local.

## 5.10 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y de código libre. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. Una de sus características principales es que no tiene a una empresa detrás de él, sino que es mantenido por una amplia comunidad de desarrolladores que trabajan de forma desinteresada, altruista y libre.

PostgreSQL es el sistema gestor de base de datos que utiliza Heroku por defecto y es el que se encarga de la persistencia de nuestros datos en la versión del *Buzón Ciudadano* que tenemos en producción.

## 5.11 Git

*Git* es un software de control de versiones diseñado por Linus Torvalds (creador del sistema operativo Linux) que permite tener un control absoluto sobre todos los cambios que vamos haciendo durante el desarrollo de nuestro software. Es una herramienta imprescindible, ya que en caso de necesidad nos permite volver a versiones previas de nuestro código o incluso crear ramas para seguir desarrollos paralelos sin afectar al código “maestro”.

**Repositorio del Buzón Ciudadano:** [https://github.com/ecarrillo88/Suggestion\\_Box](https://github.com/ecarrillo88/Suggestion_Box)

## 5.12 Heroku

*Heroku* es una plataforma como servicio (abstracción de un ambiente de desarrollo) de computación en la nube que permite desplegar en su infraestructura aplicaciones escritas en varios lenguajes, como *Ruby*, *Java*, *Python*... Con *Heroku* he podido desplegar el *Buzón Ciudadano* en su sistema y simular su puesta en producción como si de un producto final se tratase. De esta forma he podido comprobar como el desarrollar y ejecutar una aplicación en un entorno de desarrollo local es muy diferente a publicarla y ejecutarla en un servidor “real”.

**URL Buzón Ciudadano:** <http://tfg-suggestion-box.herokuapp.com/>

## Capítulo 6. Arquitectura MVC

La arquitectura utilizada en el desarrollo del *Buzón Ciudadano* es la arquitectura Modelo-Vista-Controlador, también conocida como MVC, y es en la arquitectura en la que se basa Ruby on Rails para el desarrollo de aplicaciones web. Esta arquitectura presenta varias ventajas, como la separación de la lógica de negocio de la interfaz de usuario, la facilidad para mantener el código limpio y el principio de desarrollo *DRY (D'ont Repeat Yourself)*, que añade el framework de Ruby on Rails y promueve la reducción de código duplicado.

En la siguiente figura se muestra cómo interaccionan los diferentes componentes de la arquitectura MVC junto con su flujo de ejecución.

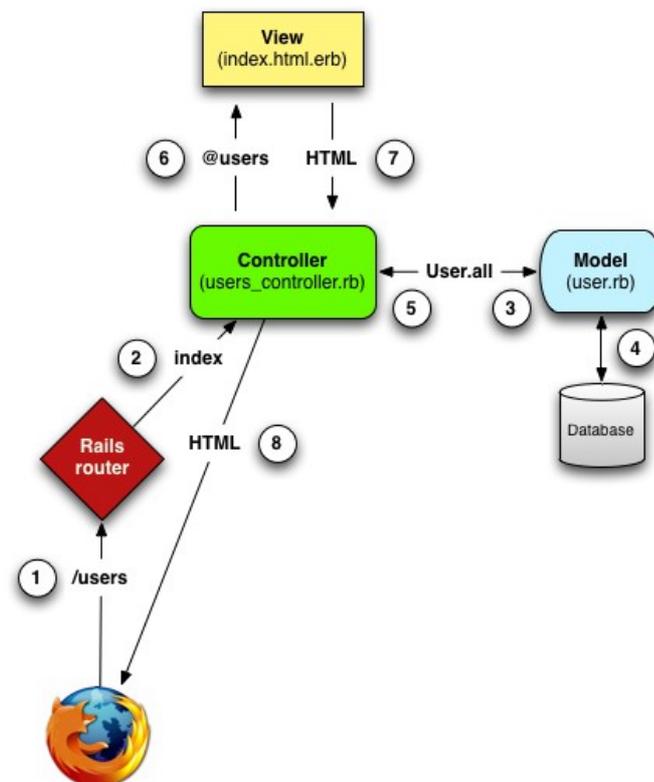


Figura 7: Arquitectura MVC

Su funcionamiento es el siguiente:

1. Un usuario a través de su navegador web realiza una petición al servidor para visualizar una determinada página. En este caso el usuario está accediendo a la página principal del sitio web, también conocida como *Index* (*/users* en el ejemplo).
2. El enrutador de Rails recoge la petición del cliente y la redirecciona a la acción adecuada (*index*) del controlador adecuado (*users\_controller.rb*) para que ésta sea procesada.
3. El controlador ejecuta la acción *index* y en este caso en concreto se encarga de recuperar de la base de datos una lista con todos los usuarios del sistema a través de la instrucción *User.all*.
4. Internamente Rails realiza una consulta a la base de datos recuperando todos los usuarios y los mapea a su representación en objetos del tipo *User*. Este proceso es transparente para el desarrollador.
5. Estos objetos que representan a cada uno de los usuarios recuperados de la base de datos son retornados al controlador dentro de una estructura de datos, típicamente una lista.
6. El controlador busca la vista asociada a la acción *index* del controlador (*index.html.erb*) y le pasa el objeto *@users* que contiene la lista de usuarios, ejecutando el código Ruby embebido en la vista para formar el documento HTML final que será retornado al cliente.
7. La vista ya formateada a HTML puro es retornada al controlador.
8. Por último el controlador envía el documento HTML al cliente para que éste pueda mostrárselo al usuario.

## 6.1 Modelo

El modelo representa la información (los datos) de la aplicación y las reglas para manipular dicha información. En el caso de Rails, los modelos son utilizados principalmente para gestionar la interacción con las tablas de la base de datos. En la mayoría de los casos, cada tabla de la base de datos se corresponde con un modelo de la aplicación. La mayor parte de la lógica de negocio de la aplicación se concentrará en los modelos.

## 6.2 Vista

Las vistas representan las interfaces de usuario de nuestra aplicación. En Rails las vistas son principalmente documentos HTML con código Ruby embebido que realiza las tareas relacionadas únicamente con la presentación de los datos. Las vistas se encargan de recoger los datos introducidos por los usuarios y de realizar las peticiones al servidor.

## 6.3 Controlador

Los controladores son el “pegamento” entre los modelos y las vistas. En Rails, los controladores son los responsables de procesar las peticiones entrantes procedentes del navegador del usuario, la consulta de los modelos que representan los datos y de pasar el resultado de los datos a las vistas para su representación gráfica. Por último proporciona la vista al navegador que hizo la petición para mostrar los resultados al usuario.

## 6.4 Módulos de Ruby On Rails

Ruby on Rail se compone de varios módulos y cada uno de ellos se encarga de una parte del funcionamiento del framework. En la siguiente figura se pueden ver los distintos módulos que componen Ruby on Rails y la interacción de unos con otros.

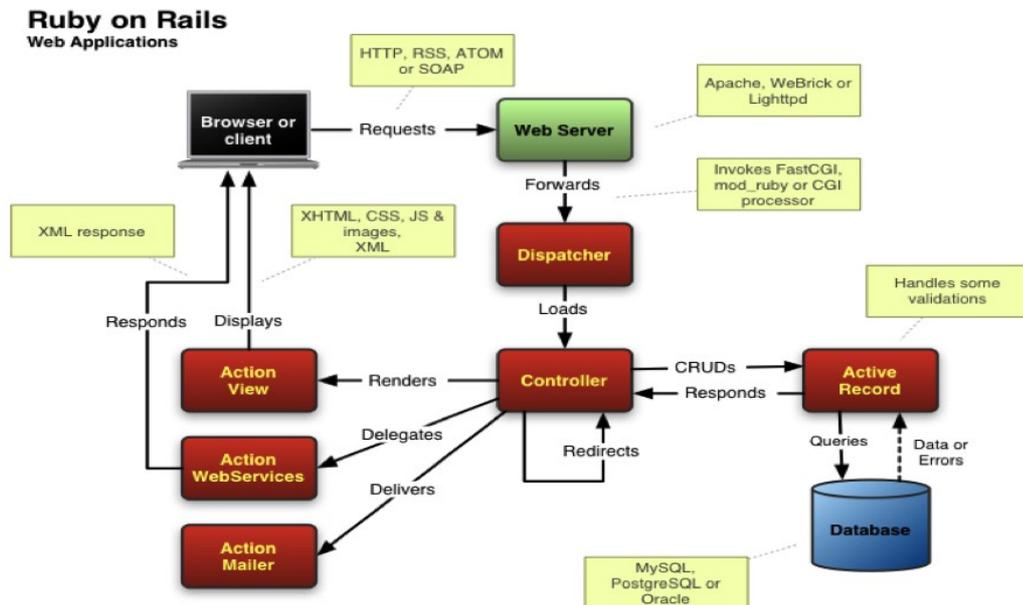


Figura 8: Módulos Ruby on Rails

### 6.4.1 Action Pack

El módulo Action Pack proporciona las capas controlador y vista del patrón MVC. Estos módulos capturan las peticiones de los usuarios realizadas a través del navegador web y las mapea con las acciones correspondientes. Estas acciones son definidas en la capa controlador y posteriormente las acciones renderizan las vistas que son visualizadas en el navegador. Action pack se divide en tres submódulos: Action Dispatch, Action Controller y Action View.

#### 6.4.1.1 Action Dispatch

Se encarga del enrutamiento de las solicitudes del navegador web, rediriéndolas al controlador adecuado.

#### 6.4.1.2 Action Controller

Es el componente que se encarga de gestionar los controladores en las aplicaciones Rails. El Action Controller procesa las solicitudes entrantes, extrae los parámetros y los envía a la acción correspondiente. Los servicios proporcionados por el Action Controller incluyen la gestión de sesiones, la representación de las vistas y la gestión de las redirecciones.

#### 6.4.1.3 Action View

El Action View se encarga de la gestión de las vistas de nuestra aplicación y es llamado por el Action Controller. Se encarga de la renderización de las vistas y la composición de las mismas a partir de vistas parciales.

## 6.4.2 Action Mailer

Este módulo es el encargado de proveer los servicios de correo electrónico. Se encarga de procesar los emails entrantes y de la creación de emails nuevos. Este módulo puede manejar emails de texto plano o de formato enriquecido con HTML.

## 6.4.3 Active Model

Define la interfaz entre los módulos Action Pack y Active Record.

## 6.4.4 Active Record

El módulo de Active Record es un patrón arquitectónico usado para gestionar los datos almacenados en las bases de datos relacionales. Además se encarga del mapeo objeto-relacional que transforma las tablas de nuestra base de datos en objetos. Active Record provee la independencia de la base de datos utilizada y provee las herramientas necesarias para implementar la funcionalidad CRUD (*Create, Read, Update, Delete*) sin ningún tipo de configuración previa. CRUD nos permite crear, leer, modificar y eliminar registros de la base de datos a través de objetos Ruby sin utilizar lenguaje SQL.

Este módulo es utilizado para la creación de las clases de los modelos, que contienen la lógica de negocio, gestionan las validaciones y las relaciones entre objetos y encapsulan el acceso a los datos.

## 6.4.5 Active Resource

Active Resource es el módulo encargado de la gestión de las conexiones entre los servicios web RESTful y los objetos de negocio. Active Resource se encarga del mapeo de las clases de nuestros modelos en recursos REST remotos, de la misma manera en que el Active Record se encarga del mapeo que transforma los modelos en tablas de la base de datos.

## 6.4.6 Active Support

Active Support es una colección de clases y librerías que se utilizan en el desarrollo de Rails. Incluye la internacionalización, la gestión de zonas horarias, el testing...

## 6.4.7 Railties

Railties es el núcleo de Ruby on Rails que se encarga de la creación de nuevas aplicaciones Rails. Es el encargado de hacer que todos los módulos que forman el framework trabajen en conjunto. Además se encarga de la generación automática de código y de la interfaz de línea de comandos. Rake es una de las interfaces de línea de comandos que se encarga de las tareas relacionadas con la base de datos, despliegue de la aplicación, documentación y testing.

# Capítulo 7. Trabajo realizado

Como se ha comentado en capítulos anteriores, para el diseño y desarrollo del *Buzón Ciudadano* se han empleado metodologías ágiles, lo que implica que la documentación referente al proyecto sea bastante escasa. No se ha desarrollado ningún tipo de diagrama de clases, de base de datos o de casos de uso de las funcionalidades previstas, así como de ningún tipo de documentación previa salvo las historias de usuario.

## 7.1 Historias de usuario implementadas

Las Historias de usuario implementadas son las que se pueden ver en la versión final del *Buzón Ciudadano* y son las siguientes. Están ordenadas por orden de prioridad y finalización de las mismas durante el desarrollo de la aplicación.

### 7.1.1 Ver sugerencias

Esta historia de usuario consiste en implementar la funcionalidad básica para ver en la página principal un listado de sugerencias. Entre sus tareas se encuentra la creación del modelo de datos para poder manipular sugerencias en el futuro y el diseño de una primera versión de la página principal del buzón. Además se introducen algunas sugerencias directamente en la base de datos para poder utilizarlas temporalmente y darlas formato en la vista para poder tener una representación gráfica de como se verán más adelante.

Ver Sugerencias
<b>Descripción</b>
Como Antonia Quiero ver una lista de las sugerencias Para enterarme de lo que pasa en mi localidad
<b>Tareas</b>
<ul style="list-style-type: none"><li>• Crear modelo para sugerencias</li><li>• Mostrar lista de sugerencias</li><li>• Estilo de la página principal</li></ul>



### 7.1.2 Crear sugerencias

Una funcionalidad básica del buzón es la de crear sugerencias, por lo que se crea un formulario con los campos correspondientes para poder dar de alta nuevas sugerencias, añadiendo las funcionalidades para subir imágenes y geolocalizar la sugerencia mediante los mapas de *Google Maps*. Además se crea el modelo de datos para poder manipular sugerencias y se actualiza la página principal para que se muestren las sugerencias que se van creando.

Crear sugerencia
<b>Descripción</b>
Como Antonia Quiero crear una sugerencia Para que el ayuntamiento este informado de lo que me pasa
<b>Tareas</b>
<ul style="list-style-type: none"> <li>• Formulario para crear la sugerencias</li> <li>• Solicitar título, comentario y autor para la sugerencia</li> <li>• Subir imágenes (Cloudinary, plugin jQuery)</li> <li>• Añadir validación de tamaño para las imágenes</li> <li>• Geolocalización con Google Maps</li> <li>• Cuando la sugerencia sea creada debe mostrarse en la lista de sugerencias publicadas</li> </ul>

### 7.1.3 Ficha de sugerencia

Una vez tenemos implementada la funcionalidad para crear nuevas sugerencias hay que implementar la funcionalidad para poder ver dichas sugerencias una vez han sido publicadas. Para ello se diseña una vista genérica para mostrar la ficha de las sugerencias a los usuarios mostrando su título, autor, texto de la sugerencia, imágenes subidas (opcional), y un mapa con la dirección postal a la que afecta la sugerencia (opcional).

Ficha de sugerencia
<b>Descripción</b>
Como Ana Quiero poder ver las sugerencias creadas por mis vecinos Para estar informada de lo que ocurre en mi localidad
<b>Tareas</b>
<ul style="list-style-type: none"> <li>• Diseñar interfaz para mostrar la ficha de cada sugerencia</li> </ul>

### 7.1.4 Estadísticas de sugerencia

Se incluye unas estadísticas básicas en cada sugerencia para que los usuarios puedan ver el estado de las mismas. La información que se muestra es el número de comentarios, el número de firmantes, el porcentaje de votos, la fecha de publicación, el tipo de la sugerencia y el nombre del autor. Esta información será visible tanto desde la lista de sugerencias de la página principal como desde la propia ficha de cada una de las sugerencias.

Estadísticas de sugerencia
<b>Descripción</b>
Como vecino Quiero ver las estadísticas de cada sugerencia (número de votos, número de comentarios, número de firmas, número de veces vista por otros vecinos...) Para saber cómo de popular es una sugerencia en mi localidad
<b>Tareas</b>
<ul style="list-style-type: none"><li>• Diseñar una cabecera para las estadísticas</li><li>• Mostrar tipo de sugerencia, nombre del autor, fecha de publicación, número de comentarios, número de firmas y porcentaje de votos a favor y en contra.</li></ul>

### 7.1.5 Comentar sugerencias

Se crea un formulario dentro de la ficha de cada sugerencia para poder publicar nuevos comentarios en las mismas. Para ello se crea el modelo de datos para los comentarios y se implementa la funcionalidad para poder publicarlos. Los comentarios de cada sugerencia se muestran al final de la misma ordenados por fecha de antigüedad.

Comentar sugerencia
<b>Descripción</b>
Como Ana Quiero realizar comentarios en las sugerencias de las que tenga más información Para transmitir esa información a los vecinos
<b>Tareas</b>
<ul style="list-style-type: none"><li>• Formulario para dar de alta comentarios</li><li>• Crear modelo para comentarios</li><li>• Diseñar estilo de los comentarios</li><li>• Los comentarios se mostrarán debajo de la sugerencia ordenados por antigüedad</li><li>• Mostrar nombre, comentario y fecha de publicación</li><li>• Actualizar contador de comentarios en la sugerencia</li></ul>



### 7.1.6 Ficha de sugerencia Antonia

Después de tener una versión totalmente funcional de las fichas de las sugerencias vimos que Antonia iba a tener dificultades para poder leer algunos textos y para poder identificar algunas funcionalidades debido a su tamaño. En esta tarea se implementó una funcionalidad para poder aumentar y disminuir el texto de la sugerencia al antojo del usuario y se mejoró la usabilidad general de la ficha de la sugerencia cambiando algunos tipos de fuente, colores...

Ficha de sugerencia Antonia
<b>Descripción</b>
Como Antonia Quiero ver todos los datos de una sugerencia Para ver si la sugerencia se refiere al camino de su casa al mercado
<b>Tareas</b>
<ul style="list-style-type: none"> <li>• Funcionalidad para aumentar y disminuir el tamaño del texto de la sugerencia</li> <li>• Añadir icono de lupa debajo del texto de la sugerencia</li> <li>• Mejorar la usabilidad general de la vista de sugerencias</li> </ul>

### 7.1.7 Validación nueva sugerencia

Hasta ahora cualquier persona podía crear sugerencias sin ningún tipo de control ni validación. En esta tarea se implementó la validación de las sugerencias enviando un mensaje de correo electrónico al creador antes de publicar su sugerencia. De esta forma evitamos que se publiquen sugerencias falsas o con direcciones de correo electrónico de otras personas obligando a que el usuario utilice una cuenta de correo válida y de la que posee la contraseña para poder acceder a validar su sugerencia.

Validación nueva sugerencia
<b>Descripción</b>
Como administrador del sitio Quiero que cada vez que se crea una sugerencia Se envíe un email al creador para validar la autenticidad de su email
<b>Tareas</b>
<ul style="list-style-type: none"> <li>• Crear código único por sugerencia (token)</li> <li>• Enviar un correo al autor</li> <li>• Activar sugerencia</li> <li>• Avisar al usuario de que se le ha mandado un correo</li> <li>• Redactar texto para el email</li> <li>• Informar al usuario de que su sugerencia se ha activado correctamente</li> </ul>

### 7.1.8 Validación de comentario

Al igual que con las sugerencias, se implementó la validación de los comentarios para evitar comentarios fraudulentos. El procedimiento es el mismo, se envía un mensaje de correo electrónico a la cuenta del autor del comentario antes de su publicación. Si el usuario valida el comentario pulsando el enlace de validación del correo que se le ha enviado entonces el comentario se publica. En caso contrario el comentario no será publicado.

Validación de comentario
<b>Descripción</b>
Como administrador del sistema Quiero que cuando un usuario cree un comentario por primera vez se le envíe un correo de validación Para validar su cuenta de correo y evitar así el <i>spam</i> y los comentarios fraudulentos
<b>Tareas</b>
<ul style="list-style-type: none"><li>• Crear código único por comentario (token)</li><li>• Enviar un correo al autor</li><li>• Activar comentario</li><li>• Avisar al usuario de que se le ha mandado un correo</li><li>• Redactar texto para el email</li><li>• Informar al usuario de que su comentario se ha validado</li></ul>

### 7.1.9 Ficha de sugerencia Juan

En este punto del desarrollo la ficha para poder consultar las sugerencias ya estaba terminada y con la usabilidad revisada para que Antonia no tenga problemas al leer las sugerencias. Se nos ocurrió que a Juan le gustaría algo más moderno y se implementó un *slideshow* para ver las imágenes en grande y poder desplazarse entre ellas con efectos de transición. Además se mejoró el diseño general para que sea “responsive” y se adapte mejor a dispositivos móviles.

Ficha de sugerencia Juan
<b>Descripción</b>
Como Juan Quiero ver todos los datos de una sugerencia Para ver si me interesa y poder respaldarla mas adelante
<b>Tareas</b>
<ul style="list-style-type: none"><li>• Recortar imagen al subir</li><li>• Redimensionar imagen en función de la vista</li><li>• Responsive design</li><li>• Slideshow para las imágenes</li></ul>



### 7.1.10 Pagar las sugerencias de la página principal

Se introduce un paginador en la página principal del buzón para evitar tener demasiadas sugerencias en pantalla y que el usuario tenga que desplazarse hacia abajo. Con esto conseguimos tener la vista general de las sugerencias más compacta y ahorramos en consumo de red al traer menos elementos al cliente. Se ha configurado para mostrar diez sugerencias por página.

Pagar las sugerencias de la página principal
<b>Descripción</b>
Como Juan Quiero que la página principal cargue rápido Para poder enterarme cuanto antes de lo que ocurre en mi localidad
<b>Tareas</b>
<ul style="list-style-type: none"> <li>• Paginador de 10 resultados por página</li> </ul>

### 7.1.11 Rediseñar formulario de comentarios

El formulario que había hasta ahora era muy simple y se decidió rediseñarlo para que fuese más moderno y se adaptara mejor al diseño general de la página. Para ello se modificaron los estilos del mismo y se añadió mediante jQuery la funcionalidad para que se desplace con el *scroll* de la página y el usuario lo tenga siempre visible (por ahora solo funciona en Firefox)

Rediseñar formulario de comentarios
<b>Descripción</b>
Como Victoria Quiero que el formulario para publicar nuevos comentarios tenga un diseño más acorde con el diseño general de la aplicación Para que se integre mejor con el resto de elementos
<b>Tareas</b>
<ul style="list-style-type: none"> <li>• Rediseñar formulario de comentarios</li> </ul>

### 7.1.12 Feature tests

Con el desarrollo ya bastante avanzado y con parte de las funcionalidades importantes de la aplicación ya implementadas empecé a desarrollar algunos casos de prueba y a probar el funcionamiento general de la aplicación. Hasta este momento no era muy importante debido a las pocas funcionalidades de la aplicación, pero según iba creciendo me di cuenta de que a la larga iba a ser imposible probar toda la aplicación para comprobar si se han ido introduciendo errores al implementar nuevas funcionalidades. Los *Feature tests* se encargan de probar las interfaces gráficas y el funcionamiento de sus elementos.

Feature tests
<b>Descripción</b>
Como desarrollador Quiero poder probar la aplicación en cualquier momento Para estar seguro de que se mantiene libre de errores y poder corregir los que van surgiendo
<b>Tareas</b>
<ul style="list-style-type: none"><li>• Probar la funcionalidad de las interfaces de creación de sugerencias y comentarios</li><li>• Probar el funcionamiento de los botones y enlaces que llevan al formulario de creación de sugerencias</li><li>• Comprobar que al crear una sugerencia el email del autor se añade a la lista blanca de emails válidos</li><li>• Probar que las sugerencias publicadas aparecen en el listado de la página principal</li></ul>

### 7.1.13 Internacionalización del buzón

Para poder adaptar el *Buzón Ciudadano* al idioma de cualquier región o país se ha implementado la internacionalización del mismo. Con esta internacionalización se consigue que traducir una aplicación a otro idioma sea una tarea trivial dónde solo hay que traducir un simple fichero de configuración sin necesidad de entrar al código fuente y traducir todos los textos.

Para esta primera versión del *Buzón Ciudadano* se ha optado por traducirlo al castellano y al inglés por ser los dos idiomas más hablados en el mundo.

Internacionalización del buzón
<b>Descripción</b>
Como Victoria Quiero que mi Buzón Ciudadano pueda ser traducido a cualquier idioma Para adaptarlo al idioma de mis vecinos
<b>Tareas</b>
<ul style="list-style-type: none"><li>• Añadir I18n</li><li>• Traducir al castellano y al inglés</li></ul>



### 7.1.14 Firmar sugerencias

Además de poder publicar comentarios, también queremos que los vecinos puedan implicarse de alguna forma con las sugerencias que les interesan. El concepto es el mismo que el de publicar un comentario, con la diferencia de que ahora se podrá ver además el número de firmas en la cabecera de la sugerencia para ver a simple vista si la sugerencia interesa o no a los vecinos.

<b>Firmar sugerencias</b>
<b>Descripción</b>
Como Vecino Quiero respaldar las sugerencias de otros vecinos aportando más información Para que la sugerencia tenga más visibilidad
<b>Tareas</b>
<ul style="list-style-type: none"> <li>• Cuando se respalda una sugerencia, se envía un correo al que respalda con un enlace único. Cuando pulsa el enlace entonces es cuando tiene efecto la acción</li> <li>• Actualizar contador de firmas en las estadísticas de la sugerencia</li> <li>• tests</li> </ul>

### 7.1.15 Actividad para los firmantes

Para dar un valor añadido a las personas que han firmado las sugerencias a partir de ahora serán notificadas cuando se publiquen nuevos comentarios para que puedan estar al día de la evolución de la sugerencia sin tener que estar entrando la web a ver su estado.

<b>Actividad para los firmantes</b>
<b>Descripción</b>
Como Juan, que firmo una sugerencia Quiero recibir actualizaciones Para estar al día de la sugerencia
<b>Tareas</b>
<ul style="list-style-type: none"> <li>• Cuando hay un nuevo comentario se manda un correo.</li> <li>• Poner el número de firmantes en el mensaje de correo</li> </ul>

### 7.1.16 Test unitarios

Además de los tests para probar las funcionalidades básicas de la aplicación, también se tuvo la necesidad de incluir tests unitarios para probar fragmentos de código con el fin de evitar introducir errores al añadir nuevas funcionalidades.

Tests unitarios
<b>Descripción</b>
Como desarrollador Quiero poder probar el código de mi aplicación Para estar seguro de que se mantiene libre de errores y poder corregir los que van surgiendo
<b>Tareas</b>
<ul style="list-style-type: none"><li>• Probar el subir imágenes</li><li>• Probar la publicación de sugerencias</li><li>• Probar la whitelist de emails validados<ul style="list-style-type: none"><li>◦ Si el usuario no está en la whitelist añadirlo</li></ul></li><li>• Probar actualizar sugerencia</li><li>• Probar el envío de emails</li><li>• Probar mensajes de retorno al usuario</li></ul>

### 7.1.17 Rol ayuntamiento destacado

Para diferenciar los comentarios de los vecinos de los del personal del ayuntamiento, que pueden comentar como un vecino más, se decidió modificar su estilo visual para que fuesen reconocidos a simple vista. Además al personal del ayuntamiento se le envía un mensaje de correo electrónico informativo dónde se le explica que los comentarios del personal del ayuntamiento tienen mayor impacto y responsabilidad que los de los vecinos al venir directamente de una institución pública.

Rol ayuntamiento destacado
<b>Descripción</b>
Como responsable del ayuntamiento Quiero aparecer destacado en los comentarios de las sugerencias Para que los vecinos puedan distinguirme del resto de vecinos
<b>Tareas</b>
<ul style="list-style-type: none"><li>• El dominio del email del comentario se comprueba contra una lista de dominios configurable</li><li>• El correo tiene que ser distinto al de los comentarios normales, informando de lo que implica ser del ayuntamiento</li><li>• Destacar el comentario en la interfaz</li><li>• Tests</li></ul>



### 7.1.18 Punto de vista sobre la sugerencia

Una idea interesante que surgió durante el desarrollo de la aplicación fue el poder dar el punto de vista al comentar en una sugerencia para poder diferenciar los comentarios a favor, de los neutrales, de los de encostra. Para ello se incluyó un desplegable con estas tres opciones en el formulario de publicación de comentarios y se añadió un icono al lado de los comentarios de los vecinos para ver a simple vista el punto de vista del mismo sin tener que leerlo. Además se incluyó una barra con el porcentaje de votos a favor y en contra en cada sugerencia parra poder ver a simple vista que opina la gente.

Punto de vista sobre la sugerencia
<b>Descripción</b>
Como Victoria Quiero que en cada comentario se indique si la intención del comentario es positiva, negativa o neutra con respecto a la sugerencia Para poder ver a simple vista que opina un vecino de la sugerencia
<b>Tareas</b>
<ul style="list-style-type: none"> <li>• En cada comentario es obligado marcar una de las 3 opciones               <ul style="list-style-type: none"> <li>◦ A favor</li> <li>◦ Neutral</li> <li>◦ En contra</li> </ul> </li> <li>• Indicar el punto de vista del vecino junto a su nombre en el comentario mediante un icono</li> <li>• Añadir una barra de orcentaje de votos a favor y en contra en la sugerencia</li> <li>• Tests</li> </ul>

### 7.1.19 Filtrar sugerencias

Se añade un filtro de búsquedas en la página principal para que los usuarios puedan filtrar las sugerencias por algún criterio concreto, como el título de la misma, dirección postal o distancia de un punto en concreto.

Filtrar sugerencias
<b>Descripción</b>
Como Juan Quiero filtrar el listado de sugerencias Para consultarlas por algún criterio
<b>Tareas</b>
<ul style="list-style-type: none"> <li>• Filtrar por título</li> <li>• Filtrar por dirección postal</li> <li>• Filtrar por proximidad (500m)</li> <li>• Tests</li> </ul>

## 7.1.20 Mostrar etiqueta con la dirección de la sugerencia

Además del título, del nombre del autor, numero de comentarios... Se han añadido unas etiquetas con la dirección postal de las sugerencias que han sido geolocalizadas para que sin necesidad de acceder a su ficha los usuarios puedan ver si están o no geolocalizadas y a que dirección postal afectan.

<b>Mostrar etiqueta con la dirección de la sugerencia</b>
<b>Descripción</b>
Como Juan Quiero ver en la lista de sugerencias la dirección de las sugerencias que han sido geolocalizadas Para ver a simple vista a qué zona afecta la sugerencia
<b>Tareas</b>
<ul style="list-style-type: none"><li>• Diseño de las etiquetas para la página principales</li><li>• Formatear dirección postal para mostrar la calle, número, localidad y provincia</li></ul>

## 7.1.21 Añadir tipo de sugerencia

Hasta ahora las sugerencias no tenían ningún tipo y no había manera de poder clasificarlas o filtrarlas, por lo que se decidió añadir varios tipos (sugerencia, queja, felicitación e incidencia) para que los vecinos puedan clasificar el tipo de sus sugerencias al crearlas o filtrarlas por tipo en el filtro de búsqueda de la página principal.

<b>Añadir tipo de sugerencia</b>
<b>Descripción</b>
Como Juan Quiero ver el tipo de sugerencia al lado del título de la misma Para saber de que tipo es sin necesidad de entrar a leerla
<b>Tareas</b>
<ul style="list-style-type: none"><li>• Añadir la opción de seleccionar el tipo de la sugerencia en el formulario de creación (sugerencia, queja, felicitación, problema...)</li><li>• Añadir el tipo de la sugerencia debajo del título de esta.</li><li>• Filtrar por tipo de sugerencia en el filtro de búsquedas de la página principal</li></ul>

### 7.1.22 Modificar sugerencia

Funcionalidad para permitir a los vecinos modificar sus sugerencias para corregir errores o aportar más información. Para permitir la edición de una sugerencia se envía un mensaje de correo electrónico al dueño de la misma con un enlace de confirmación que le dará acceso a la edición de la sugerencia.

<b>Modificar sugerencia</b>
<b>Descripción</b>
Como Antonia Quiero poder modificar las sugerencias que yo he creado Para poder añadir información extra o corregir errores
<b>Tareas</b>
<ul style="list-style-type: none"> <li>• Generar código único y mandarlo al email del dueño de la sugerencia</li> <li>• Validar código y si es correcto redirigir a la edición de la sugerencia</li> <li>• Validar código y si no es correcto mostrar error</li> <li>• Tests</li> </ul>

### 7.1.23 Eliminar comentario

Para permitir que los usuarios puedan eliminar sus comentarios se ha añadido la funcionalidad de eliminar comentarios. Para ello se envía un mensaje de correo electrónico al dueño del comentario con un enlace de confirmación que se encarga de eliminar el comentario una vez que el usuario lo ha pulsado.

<b>Eliminar comentario</b>
<b>Descripción</b>
Como Juan Quiero poder eliminar mis comentarios Para que desaparezcan de la sugerencia
<b>Tareas</b>
<ul style="list-style-type: none"> <li>• Generar código único y mandarlo al email del dueño del comentario</li> <li>• Validar código y si es correcto eliminar comentario y redirigir a la sugerencia</li> <li>• Validar código y si no es correcto redirigir a la sugerencia y mostrar error</li> <li>• Tests</li> </ul>

### 7.1.24 Eliminar sugerencia

Su funcionamiento es exactamente el mismo que el de eliminar comentarios. Para permitir que los usuarios puedan eliminar sus sugerencias se envía un mensaje de correo electrónico al dueño de la sugerencia con un enlace de confirmación que se encarga de eliminar la sugerencia una vez que el usuario lo ha pulsado.

Eliminar sugerencia
<b>Descripción</b>
Como Antonia Quiero eliminar mis sugerencias Para que desaparezcan del sistema y no se tengan en cuenta
<b>Tareas</b>
<ul style="list-style-type: none"><li>• Generar código único y mandarlo al email del dueño de la sugerencia</li><li>• Validar código y si es correcto eliminar la sugerencia e informar al usuario</li><li>• Validar código y si no es correcto mostrar error</li><li>• Tests</li></ul>

### 7.1.25 Inactividad de sugerencia

Para evitar tener sugerencias abiertas de forma indefinida se ha implementado una funcionalidad para que pasados 30 días desde el último comentario publicado en una sugerencia se avise vía email a todos los usuarios que han comentado en ella para que sepan que si en 7 días no se publican nuevos comentarios la sugerencia se cerrará. Con esto conseguimos aumentar la vida de la sugerencia y que los usuarios no se olviden de ella.

Inactividad en sugerencia
<b>Descripción</b>
Como Antonia, que he firmado la sugerencia o he comentado Quiero recibir un correo que me avise cuando alguna de las sugerencias en las que he comentado esté inactiva durante un mes Para poder comentar con la intención de aportar más información o resolver los comentarios negativos de otros vecinos
<b>Tareas</b>
<ul style="list-style-type: none"><li>• Mandar correo después de 30 días desde el último comentario</li><li>• Mostrar las estadísticas de la sugerencia en el email</li><li>• tests</li></ul>



### 7.1.26 Concluir sugerencia

Si después de siete días tras el aviso de cierre la sugerencia ésta no ha recibido nuevos comentarios entonces se cerrará. Cerrar una sugerencia implica que a partir de ese momento solo se permite comentar al personal del ayuntamiento y además se enviará un mensaje de correo electrónico al/los responsable/es del ayuntamiento encargados de tomar decisiones para que sepan que la sugerencia se ha cerrado y que ahora está en sus manos el solucionar los problemas o dejarlos pasar.

Concluir sugerencia
<b>Descripción</b>
Como Victoria Quiero que se cierren las sugerencias que tras el primer aviso no han vuelto a recibir comentarios pasados 7 días Para evitar que haya sugerencias muertas sin resolver en el sistema
<b>Tareas</b>
<ul style="list-style-type: none"> <li>• Enviar email informando de que la sugerencia se ha cerrado</li> <li>• Se manda un correo a un responsable del ayuntamiento</li> <li>• Solo se permite comentar al personal del ayuntamiento</li> <li>• Tests</li> </ul>

### 7.1.27 Diseño general de la web

Esta tarea ha sido constante durante todo el desarrollo del *Buzón Ciudadano* debido a que cada vez que se añadía alguna funcionalidad ha habido que retocar el diseño para adaptarlo a las nuevas necesidades.

Diseño general de la web
<b>Descripción</b>
Como diseñador Quiero que la aplicación sea bonita y usable Para que agrade a mis usuarios.
<b>Tareas</b>
<ul style="list-style-type: none"> <li>• Cabecera grande</li> <li>• Formato de tarjetas</li> <li>• Responsive design</li> <li>• Diseño minimalista</li> <li>• Elementos grandes y vistosos</li> </ul>

### 7.1.28 Añadir Ajax al filtro de búsqueda y al paginador

Se ha utilizado la tecnología Ajax en el filtro de búsquedas de la página principal y en el paginador de sugerencias para evitar que la página entera se recargue cada vez que el usuario filtra las sugerencias o cambia de página. Con esto conseguimos que el usuario no se entere de que ha habido una comunicación cliente-servidor y mejoramos la interactividad y la usabilidad del filtro de búsquedas al no tener que esperar que la página cargue cada vez que se hace una nueva petición al servidor.

<b>Añadir Ajax al filtro de búsqueda y al paginador</b>
<b>Descripción</b>
Como desarrollador Quiero que al utilizar el filtro de búsqueda no se recargue la página entera Para mejorar la interactividad y usabilidad al filtrar las sugerencias
<b>Tareas</b>
<ul style="list-style-type: none"><li>• Añadir Ajax al filtro de búsquedas y al paginador</li><li>• Enmascarar listado de sugerencias mientras la comunicación asíncrona no haya terminado</li><li>• Añadir cartel de “Cargando...” mientras se procesa la petición al servidor</li></ul>

### 7.1.29 Reportar sugerencia

Para evitar que los usuarios puedan crear sugerencias con contenido inapropiado (contenido publicitario, racista, fraudulento...) todas las sugerencias publicadas en el sistema disponen de un botón para reportarlas e informar al responsable del buzón para que tome medidas en caso de que sea necesario.

<b>Reportar sugerencia</b>
<b>Descripción</b>
Como Ana Quiero poder reportar las sugerencias con contenido inapropiado Para que el responsable del Buzón Ciudadano pueda tomar medidas
<b>Tareas</b>
<ul style="list-style-type: none"><li>• Añadir botón en la vista de cada sugerencia para reportarlas</li><li>• Enviar un mensaje de correo electrónico al responsable del buzón para informarle</li><li>• Informar al usuario mediante un mensaje en pantalla de que la sugerencia ha sido reportada</li></ul>



### 7.1.30 Reportar comentario

Para evitar que los usuarios puedan publicar comentarios con contenido inapropiado (contenido publicitario, racista, fraudulento...) todos los comentarios publicados en el sistema disponen de un botón para reportarlos e informar al responsable del buzón para que tome medidas en caso de que sea necesario.

Reportar comentario
<b>Descripción</b>
Como Ana Quiero poder reportar los comentarios con contenido inapropiado Para que el responsable del Buzón Ciudadano pueda tomar medidas
<b>Tareas</b>
<ul style="list-style-type: none"> <li>• Añadir botón en cada comentario para reportarlo</li> <li>• Enviar un mensaje de correo electrónico al responsable del buzón para informarle</li> <li>• Informar al usuario mediante un mensaje en pantalla de que el comentario ha sido reportado</li> </ul>

## 7.2 Historias de usuario sin implementar

Para el desarrollo del TFG no se propuso ningún límite ni de tiempo ni de funcionalidades. El objetivo era conseguir una versión del *Buzón Ciudadano* básica y totalmente funcional y a partir de ahí ir implementando funcionalidades secundarias para completar lo máximo posible el buzón.

En las primeras reuniones de concepción del proyecto surgieron infinidad de funcionalidades que no se han implementado por falta de tiempo y porque entonces el desarrollo del *Buzón Ciudadano* hubiese durado meses. Al tratarse de un proyecto en constante evolución las nuevas funcionalidades que van surgiendo durante las reuniones semanales son infinitas.

Aquí están las historias de usuario menos importantes que no se han implementado y que se dejan como trabajo futuro. Estas historias de usuario están definidas a nivel muy básico, por lo que aún no se había pensado en las tareas a desarrollar para llevarlas a cabo.

### 7.2.1 Reportar vecino

La idea es que los vecinos pudiesen reportar a los usuarios con un comportamiento inadecuado dentro del *Buzón Ciudadano* para poder prohibirles el acceso en el futuro

Reportar vecino
<b>Descripción</b>
Como vecino Quiero reportar a los vecinos que utilicen de forma fraudulenta o inadecuada el buzón de sugerencias Para que el responsable del ayuntamiento tome medidas

## 7.2.2 Archivar sugerencias

Además de cerrar las sugerencias cuando estás entras en un período largo de actividad, también se había pensado en archivarlas cuando estas fuese resueltas por el ayuntamiento con el fin de que los usuarios estuviesen al día de las sugerencias que se resuelven.

Archivar sugerencias
<b>Descripción</b>
Como responsable del ayuntamiento Quiero archivar las sugerencias abiertas por los vecinos Para darlas como resueltas

## 7.2.3 Compartir sugerencia

Funcionalidad para compartir en las redes sociales o mediante correo electrónico las sugerencias que a los vecinos les parezcan interesantes.

compartir sugerencia
<b>Descripción</b>
Como vecino Quiero compartir las sugerencias que me parezcan interesantes Para que otros vecinos se enteren de lo que ocurre en nuestra localidad

## 7.2.4 Sugerir títulos

Para evitar que los vecinos creen sugerencias repetidas o con el mismo nombre se había pensado en implementar una funcionalidad para que dependiendo del título de la sugerencia elegido por el usuario se muestren títulos similares ya existentes. De esta forma se evitarían títulos repetidos o muy similares o sugerencias repetidas sobre el mismo tema.

Sugerir títulos
<b>Descripción</b>
Como administrador del sistema Quiero que cuando el usuario vaya a poner el título de su sugerencia se le muestren títulos similares ya existentes Para evitar títulos o sugerencias repetidas

## 7.2.5 Firmantes pueden subir imágenes

Se pensó en dar la posibilidad a los firmantes de poder añadir imágenes a la sugerencias para completarlas aun más.

Firmantes pueden subir imágenes
<b>Descripción</b>
Como firmante de la sugerencia Quiero poder subir imágenes a las sugerencias en las que he dado mi apoyo firmando Para aportar más información o mejorar la existente



### 7.2.6 Mapa con todas las sugerencias

Se había pensado añadir en la página principal un mapa donde se muestren todas las sugerencias geolocalizadas para que los vecinos pudiesen ver a simple vista dónde se encuentran las sugerencias y ver si hay alguna cerca de su casa para enterarse de lo que pasa.

Mapa con todas las sugerencias
<b>Descripción</b>
Como Juan Quiero ver todas las sugerencias geolocalizadas en un mismo mapa Para ver si hay sugerencias cerca de mi barrio

### 7.2.7 Feed de actividad

La idea era añadir un feed de actividad en la página principal del buzón para que los vecinos pudiesen ver en tiempo real la actividad dentro del buzón. En este feed se mostrarían las nuevas sugerencias y comentarios publicados, los que han sido editados o eliminados, en que sugerencia está comentando actualmente el personal del ayuntamiento...

Feed de actividad
<b>Descripción</b>
Como Juan Quiero ver un feed de la actividad del buzón en tiempo real Para no tener que entrar en cada sugerencia para ver si hay nuevos comentarios o si han sido modificadas

### 7.2.8 Limitar número de intentos al editar o eliminar

Para evitar que algún usuario malintencionado pueda pulsar repetidas veces el botón para editar o eliminar sugerencias y comentarios ajenos y que le lleguen infinidad de mensajes de correo al dueño de la sugerencia o del comentario, se había pensado en limitar el número de veces que una misma persona pulsa dichos botones.

Limitar número de intentos
<b>Descripción</b>
Como administrador del sitio Quiero limitar el número de veces que una misma persona malintencionada pulsa el botón de eliminar o editar una sugerencia o comentario Para evitar que el dueño de la sugerencia/comentario reciba infinidad de mensajes de confirmación que él no ha solicitado

## Capítulo 8. Tipos de usuario

El *Buzón Ciudadano* cuenta con tres tipos de roles claramente diferenciados. Por una parte tenemos al usuario estándar (vecino en el ejemplo del ayuntamiento), que es el usuario principal del buzón y la persona que publica sugerencias y comenta en ellas dando su opinión para ejercer más presión en las mismas de cara a la institución detrás del buzón. Por otra parte tenemos al personal de la empresa o institución (personal ayuntamiento en nuestro ejemplo) que se encarga de responder en nombre de la institución a la que pertenece y de dar su opinión como parte interesada. Por último tendríamos al responsable del buzón, que se encarga de velar porque se cumplan las normas del mismo y es el encargado de avisar a los responsables indicados una vez las sugerencias se han cerrado para que éstos tomen medidas.

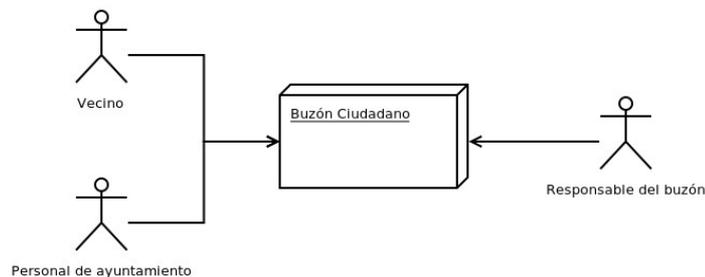


Figura 9: Tipos de usuario

La idea inicial era que el rol de responsable del buzón lo tuviese alguien que fuese ajeno a las dos partes interesadas en el *Buzón Ciudadano* para evitar la manipulación del mismo. Este rol lo podría tener, por ejemplo, la empresa encargada de instalar y poner en funcionamiento el *Buzón Ciudadano*. Por no complicar demasiado las cosas, hemos asumido que el rol de responsable del buzón lo va a tener alguna persona de la institución que lo implante, en nuestro caso sería un responsable del ayuntamiento.

## Capítulo 9. Interfaz de usuario

La interfaz de usuario del *Buzón Ciudadano* está pensada para ser accesible y fácil de utilizar, por este motivo se ha optado por un diseño de elementos grandes y vistosos con colores neutros. Para el diseño de la interfaz de usuario me he basado en el diseño minimalista de las aplicaciones de Google y en su formato de tarjetas para mostrar al usuario el contenido. Además, con interfaces simples y poco cargadas se consigue que éstas carguen más deprisa en el navegador del usuario y el tiempo de espera sea menor.

Las capturas de pantalla que se muestran a partir de ahora se corresponde con la instancia concreta del *Buzón Ciudadano* de la Universidad Politécnica de Valencia, llamado ahora *Buzón UPV*. En el capítulo 13 se explica con más detalle esta instancia y su funcionamiento.

### 9.1 Elementos comunes

La aplicación se compone de varios elementos comunes que están presentes en todas las pantallas de la aplicación. Como veremos más adelante estos elementos nos muestran información o nos permiten realizar determinadas acciones que es interesante mantener siempre a la vista del usuario para evitar que tenga que desplazarse por la página.

#### 9.1.1 Cabecera

La cabecera de la web se divide en dos secciones. Por una parte tenemos la cabecera superior de color negro en la que tenemos un acceso directo a la página principal (logo y texto de Buzón UPV) y un botón de acceso rápido a la publicación de nuevas sugerencias. Esta cabecera permanece pegada a la parte superior del navegador y se desplaza hacia abajo con nosotros para tener estas funcionalidades siempre disponibles.

La segunda cabecera, y uno de elementos más característico de la aplicación, es la cabecera gigante o *jumbotron*, la cual se encuentra presente en todas las secciones de la web y nos muestra información acerca de la sección en la que nos encontramos (página principal, alta, edición...) y una descripción sobre la misma.



Figura 10: Cabecera de la página principal

### 9.1.2 Pie de página

El pie de la web, que también es de un tamaño considerable, nos proporciona información sobre la empresa o institución que lo tenga implantado en este momento, como información de contacto, información sobre el buzón, condiciones y privacidad y enlaces a las distintas redes sociales en las que se encuentre la empresa o institución detrás del buzón. Además desde aquí podemos cambiar el idioma de la interfaz de toda la web. Actualmente la web se encuentra traducida al castellano y al inglés.

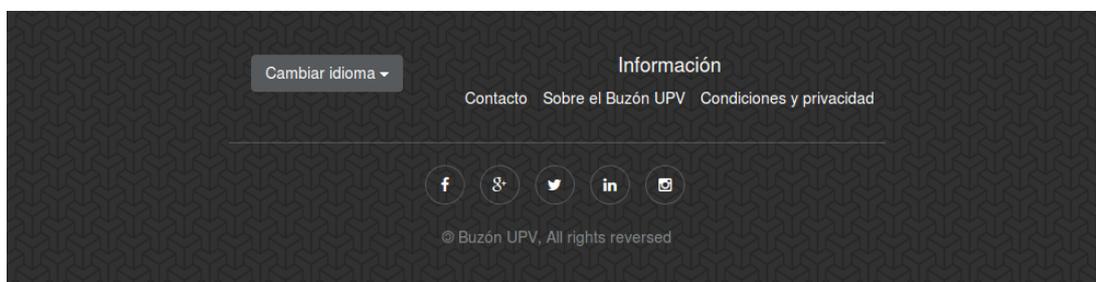


Figura 11: Pie de la web

## 9.2 Página Principal

La página principal está formada por dos secciones. Por un lado tenemos el filtro de búsquedas, desde el cual podemos filtrar las sugerencias por título, tipo, estado, dirección postal y/o distancia, y por otra parte tenemos la lista ordenada por fecha de publicación en orden ascendente de todas las sugerencias publicadas en el sistema. En esta lista se muestra un pequeño resumen de cada sugerencia, con el título de la misma, el tipo, el nombre o apodo del creador (Anonymous por defecto), la fecha de publicación, el número de comentarios, el número de firmas y la dirección postal en el caso de que esté geolocalizada. En el caso de que la sugerencia se encuentre cerrada se mostrará el icono de un candado cerrado al lado del título.

Pulsando el título de una de las sugerencias de la lista seremos redirigidos a la ficha completa de la sugerencia.

UNIVERSITAT POLITÈCNICA DE VALÈNCIA Buzón UPV Publica una sugerencia

VINIVERSIDAD POLITÈCNICA VALENCIA

# Buzón UPV

Deja tus sugerencias, comentarios, peticiones y quejas

Publica una sugerencia

Filtro de búsqueda

**Título**

**Dirección**

**Tipo**  
Todos

**Estado**  
Todos

**Distancia**  
Seleccionone una distancia

Limpiar **Buscar**

**Biblioteca abierta 24h en época de exámenes**

📍 Sugerencia 👤 Sergio92 📅 15 de Junio, 2015 💬 0 Comentarios 📄 Camí Vera, 14 46022 Valencia  
👤 0 Firmas

**Precio de la matrícula demasiado alto**

📍 Queja 👤 Manolo García 📅 30 de Mayo, 2015 💬 3 Comentarios 👤 0 Firmas

Figura 12: Página principal

## 9.3 Alta y edición de sugerencias

La ventana de alta y edición de sugerencias es exactamente la misma, la única diferencia es que si se trata de una nueva sugerencia el formulario y todas las funciones se encuentran en blanco o sin valor y si se trata de una edición de una sugerencia existente los datos se recuperan del servidor para que el dueño de la sugerencia pueda modificarlos.

La ventana de alta/edición de sugerencias se divide en tres secciones. Por una parte tenemos el formulario dónde se describe la sugerencia introduciendo su título, el tipo (sugerencia, queja, felicitaciones o incidencia), el nombre o alias del autor (por defecto el nombre del autor es Anonymous), su dirección de correo electrónico y el comentario que describe la sugerencia.

En la segunda sección tenemos dos *widgets* desde los que el usuario puede adjuntar hasta dos imágenes relacionadas con la sugerencia. Las imágenes pueden ayudar a describir la sugerencia aportando una visión gráfica de la misma.

La tercera y última sección permite al usuario geolocalizar la sugerencia para poder ubicarla en un mapa. Para geolocalizar la sugerencia el usuario puede introducir el nombre de la calle en el campo de texto o utilizar el mapa para seleccionar el punto exacto al que hace referencia la sugerencia. En ambos casos aparecerá un marcador en el mapa para que el usuario pueda ver la posición seleccionada y modificarla a su antojo.



# Nueva Sugerencia

Crea una nueva sugerencia sobre un problema, una solicitud, una queja... o cualquier cosa que creas que es importante

<b>Título</b>	<input type="text" value="Obligatorio"/>	<b>Tipo</b>	<input type="text" value="Selecciona el tipo de la sugerencia"/>
<b>Autor</b>	<input type="text" value="Anonymous por defecto"/>	<b>Email</b>	<input type="text" value="Obligatorio"/>
<b>Comentario</b>			
<input type="button" value="Bold"/> <input type="button" value="Italic"/> <input type="button" value="Underline"/> <input type="button" value="List"/> <input type="button" value="Text"/> <input type="button" value="Link"/>			
<input type="text" value="Obligatorio"/>			

## Adjuntar Imágenes

Si lo deseas puedes adjuntar imágenes que aporten más información a la sugerencia.






## Geolocaliza la sugerencia

Pon una marca haciendo click en el mapa o busca una dirección para geolocalizar la sugerencia. Si no quieres geolocalizar la sugerencia elimina la marca del mapa.

<input type="text" value="Buscar dirección"/>	<input type="button" value="Q"/>	<input type="button" value="Quitar Marca"/>
		
<input type="button" value="Publicar Sugerencia"/>	<input type="button" value="Volver"/>	

Figura 13: Alta/edición de sugerencia

## 9.4 Ver sugerencia

Al seleccionar una sugerencia de la página principal podremos acceder a su ficha, dónde tendremos acceso a la descripción completa de la sugerencia que ha publicado un usuario determinado, las imágenes que haya subido éste usuario para aportar una visión gráfica de su sugerencia y el mapa en el caso de que haya sido geolocalizada. Las imágenes y el mapa son opcionales, por lo que si no aportan nada a la sugerencia y el usuario no ha utilizado estas funcionalidades estos elementos no aparecerán en la ficha de la sugerencia.

Haciendo clic sobre las imágenes éstas se mostrarán en grande y podremos desplazarnos de una a otra con las flechas de transición que aparecen junto a la imagen que se esté mostrando actualmente.

En esta ventana también tenemos el formulario para publicar nuevos comentarios o firmar la sugerencia. Este elemento es clave para permitir a los usuarios poder dar su punto de vista sobre la sugerencia e interactuar con el resto de usuarios.

En la parte inferior tenemos los comentarios de los usuarios ordenados por antigüedad. Los comentarios de los usuarios “normales” tienen el fondo blanco y los comentarios realizados por el personal de la empresa/institución tienen el fondo amarillo para diferenciarlos del resto (Ver figura 16).



UNIVERSITAT POLITÈCNICA DE VALÈNCIA
Buzón UPV
Publica una sugerencia

# Sugerencia

Si piensas que la sugerencia es importante, deja una firma o un comentario

## Biblioteca abierta 24h en época de exámenes

📌 Sugerencia
👤 Sergio92
📅 15 de Junio, 2015
💬 5 Comentarios
  
👍 2 Firmas

👍
👎

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed mauris augue, porttitor ac ullamcorper in, imperdiet non sem. Nullam quis lacus posuere, molestie augue eu, scelerisque orci. Vivamus interdum dapibus venenatis. Fusce eget ante neque. Nulla facilisi. Cras sollicitudin id nunc ac efficitur. Integer in gravida nisi. In sed lacinia nisi, sit amet viverra nulla. Vivamus sed enim sed lacus gravida scelerisque a nec orci.

Suspendisse ultrices lorem commodo interdum finibus. In at risus eu orci sodales rhoncus sit amet et tellus. Sed erat eros, tempor a porttitor id, accumsan eget leo. Vestibulum euismod sollicitudin erat nec mollis. Sed nec vestibulum sem. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque eleifend eget erat eget iaculis. Donec porttitor dolor eu sapien pretium, ut dignissim arcu porttitor. Etiam lacus urna, finibus in porttitor et, semper vitae nisl.

Editar
Eliminar
🔍 Tamaño texto
📄 Reportar

### Deja un comentario

**Vota la sugerencia** 🗳️

### Imágenes




### Geolocalización

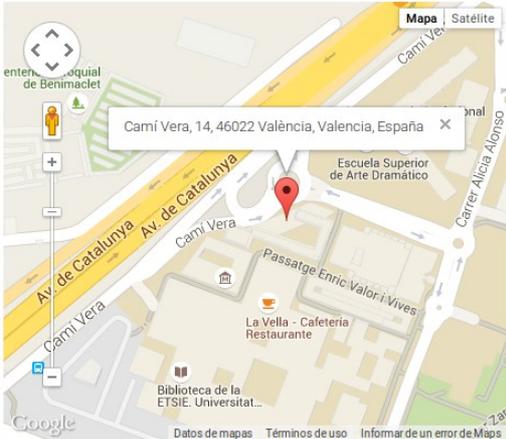


Figura 14: Vista sugerencia

## 9.5 Publicar comentario

Para publicar un comentario, al igual que para publicar una sugerencia, solo es necesario contar con una dirección de correo electrónico válida. El usuario deberá introducir el nombre que se mostrará junto al comentario (no tiene porqué ser real), su dirección de correo electrónico, el comentario que va a realizar, su punto de vista acerca de la sugerencia (a favor, abstención o en contra) y por último decidir si va a realizar un simple comentario o por el contrario va a firmar la sugerencia.

Firmar una sugerencia, además de comentar en la misma, implica estar a favor con ella y dotarla de mayor peso de cara a los responsables del buzón. La idea de poder firmar una sugerencia es que la gente se involucre de algún modo y la empresa o institución detrás del buzón vea que los usuarios además de comentar, también la respaldan con su apoyo.

Por su parte el personal de la empresa/institución también puede realizar comentarios, pero no podrá ni firmar la sugerencia ni votar la misma indicando su punto de vista. Para comentar como personal el usuario debe utilizar la dirección de correo electrónico que le haya sido asignada para su uso como personal en dicha institución, ya que el sistema conoce el dominio de las direcciones de correo electrónico de la empresa/institución y podrá identificarlos como tal ante el sistema. Los comentarios del personal están destacados para que los usuarios puedan reconocerlos a simple vista y ver que es lo que opinan los responsables del buzón.



Deja un comentario

Nombre

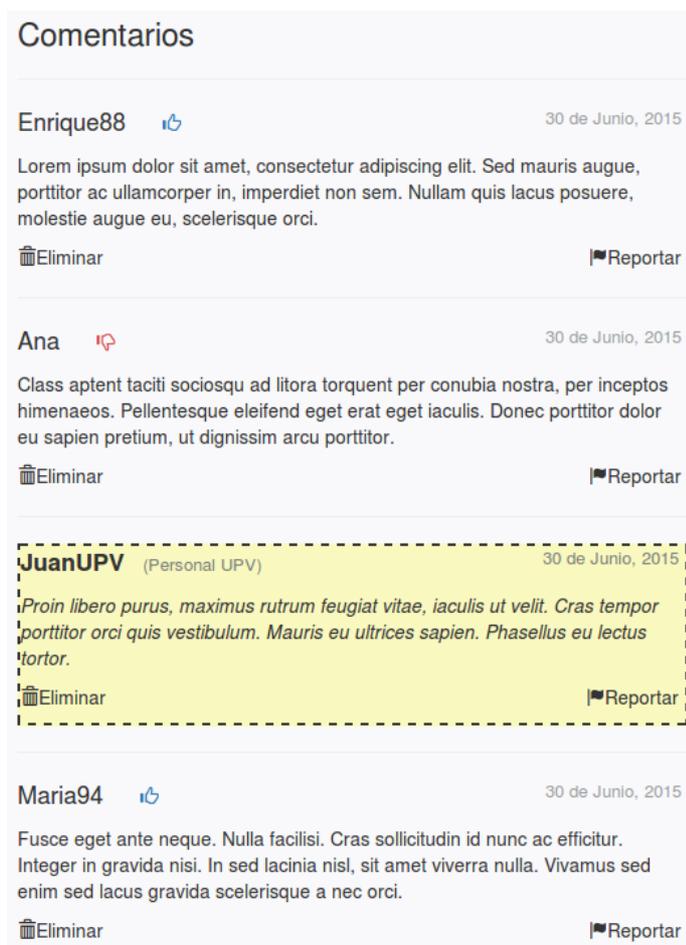
Email

Comentario

Vota la sugerencia

Seleccione una opción

Figura 15: Formulario comentarios



### Comentarios

Enrique88 30 de Junio, 2015

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed mauris augue, porttitor ac ullamcorper in, imperdiet non sem. Nullam quis lacus posuere, molestie augue eu, scelerisque orci.

Eliminar Reportar

---

Ana 30 de Junio, 2015

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque eleifend eget erat eget iaculis. Donec porttitor dolor eu sapien pretium, ut dignissim arcu porttitor.

Eliminar Reportar

---

**JuanUPV** (Personal UPV) 30 de Junio, 2015

*Proin libero purus, maximus rutrum feugiat vitae, iaculis ut velit. Cras tempor porttitor orci quis vestibulum. Mauris eu ultrices sapien. Phasellus eu lectus tortor.*

Eliminar Reportar

---

Maria94 30 de Junio, 2015

Fusce eget ante neque. Nulla facilisi. Cras sollicitudin id nunc ac efficitur. Integer in gravida nisi. In sed lacinia nisl, sit amet viverra nulla. Vivamus sed enim sed lacus gravida scelerisque a nec orci.

Eliminar Reportar

Figura 16: Comentarios

# Capítulo 10. Sistema de autenticación

Como se ha comentado en el capítulo 3, para la autenticación de los usuarios en el sistema se ha seguido el enfoque *One Less Password* [11], el cual se basa en la utilización del correo electrónico de los usuarios para proporcionarles la clave de acceso al sistema sin que éstos tenga que recordar una contraseña. Este enfoque se basa en una autenticación en dos pasos. En primer lugar el usuario se identifica en el sistema utilizando su nombre de usuario o su dirección de correo electrónico. En el segundo paso el sistema le envía al usuario un mensaje de correo electrónico con una clave de acceso que se genera automáticamente cada vez. Este segundo paso será el que le proporcione al usuario el acceso al sistema.

La ventaja principal de esta forma de identificación y acceso es que el usuario no necesita recordar una contraseña (que puede perder, olvidar o incluso le pueden robar), siendo el sistema el encargado de facilitársela cada vez que el usuario solicite acceso al mismo. La seguridad de este método de identificación reside por una parte en la identificación en dos pasos y *por otra parte* en la propia seguridad de las cuentas de correo electrónico, ya que la única forma de poder ver la contraseña que el sistema le ha enviado al usuario es tener acceso a su cuenta de correo.

Para identificar a los usuarios en el *Buzón Ciudadano* se ha seguido un enfoque muy parecido al *One Less Password* dónde la idea principal es que los usuarios no tengan que recordar un nombre de usuario y una contraseña y puedan acceder al sistema utilizando únicamente su dirección de correo electrónico.

En la siguiente figura se puede observar un pequeño resumen de los pasos que se realizan para autenticar al usuario en el sistema y proporcionarle o denegarle el acceso a determinadas acciones en el mismo.

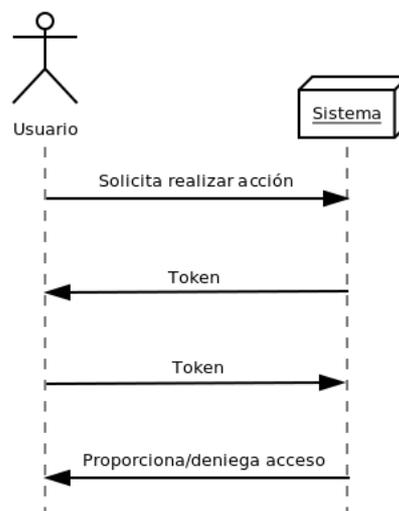


Figura 17: Autenticación de usuarios

El *Buzón Ciudadano* no dispone del registro e inicio de sesión clásico porque se ha buscado que fuese lo más sencillo de utilizar posible, y el echo de tener que crear una cuenta en el sistema para después tener que acceder al mismo cada vez que un usuario quiere utilizarlo puede echar para atrás a muchos usuarios. Por este motivo hemos utilizado una variante propia del enfoque *One Less Password*, ya que es el enfoque que mejor se adaptaba a nuestras necesidades.

Para suplir la carencia del inicio de sesión en nuestro sistema, y por ello no poder controlar los privilegios que pueda tener cada usuario dentro del mismo, nos hemos basado en acciones que deben de ser validadas o confirmadas por los propios usuarios para permitir o denegar el acceso. Todos los usuarios puede publicar, editar y eliminar comentarios y/o sugerencias, pero solamente el dueño de las mismas podrá realizar con éxito dichas acciones.

Cada vez que un usuario quiera publicar, editar o eliminar una sugerencia o un comentario deberá introducir su dirección de correo electrónico. Automáticamente el sistema le enviará un mensaje a su cuenta de correo electrónico con un enlace de validación que le permitirá identificarse ante el sistema de una forma totalmente transparente e interactuar con el mismo. Este enlace de validación incluye un token de seguridad que es generado de forma aleatoria cada vez para evitar que pueda ser utilizado más de una vez. Si al pulsar sobre el enlace de validación el token que tiene asignado coincide con el token asignado a la sugerencia/comentario, entonces se realizará la acción solicitada. En el caso de que el token de validación no coincida o ya haya sido utilizado antes, el sistema denegará la acción solicitada al usuario y se lo notificará con un mensaje de error.

Aunque a simple vista puede parecer un sistema poco seguro en comparación con el sistema de sesiones clásico (registro e inicio de sesión) al permitir que cualquier persona pueda solicitar una acción sobre cualquier sugerencia/comentario del sistema, en realidad es incluso más seguro, ya que nos apoyamos en la propia seguridad de las cuentas de correo electrónico para proporcionar o denegar el acceso a determinadas acciones en el sistema. Un usuario cualquiera puede solicitar el borrado de una sugerencia, pero sin acceso a la cuenta de correo electrónico del propietario de la sugerencia no va a poder eliminarla.

Para poder realizar alguna acción sobre una determinada sugerencia o comentario se deben de cumplir varias condiciones. La primera es que el usuario sea dueño de la sugerencia o comentario, ya que el mensaje de validación será enviado a la cuenta de correo electrónico que éste utilizó al publicar la sugerencia o el comentario. La segunda condición es que el usuario tenga acceso a su cuenta de correo electrónico para poder acceder al mensaje de validación. La tercera y última condición que se debe de cumplir es que el token de validación sea válido y no haya sido utilizado antes o manipulado por algún usuario. Apoyándonos en estas tres condiciones podemos tener la total seguridad de que ningún usuario va a poder editar o modificar sugerencias y/o comentarios que no le pertenezcan.

## 10.1 Acción de consultar

La consulta de sugerencias no tiene ningún tipo de autenticación y cualquier persona que entre al *Buzón Ciudadano* como visitante puede leer tanto las sugerencias como los comentarios de las mismas.

## 10.2 Acción de publicar

Al publicar una nueva sugerencia o un nuevo comentario el sistema enviará un mensaje de validación a la cuenta de correo electrónico del usuario y guardará la dirección de correo del usuario una vez éste valide la publicación de la sugerencia o del comentario a través del enlace que ha recibido. A partir de este momento el usuario podrá volver a publicar nuevas sugerencia o comentarios sin necesidad de validación previa, simplemente introducirá su dirección de correo



electrónico en el campo correspondiente y la sugerencia o el comentario se publicarán automáticamente.

Este mecanismo de lista blanca puede parecer inseguro a simple vista, sobre todo porque un vecino malintencionado que conozca la dirección de correo electrónico de otro vecino podría publicar sugerencias como si fuese otra persona. Ciertamente podría hacerlo, pero las posibilidades de que un usuario pueda conocer la dirección de correo electrónico que está usando otro vecino para publicar sugerencias es muy bajo. Además, sería imposible utilizar esta cuenta de correo electrónico para suplantar a otro usuario, ya que ningún dato mostrado en la propia sugerencia puede vincularla con ningún usuario en concreto.

Al personal del ayuntamiento siempre se le enviará un mensaje de confirmación al publicar nuevas sugerencias o comentarios, ya que el hecho de que un comentario aparezca identificado como comentario del ayuntamiento conlleva una gran responsabilidad, y no podemos permitir que cualquier persona comente en nombre del ayuntamiento sin estar seguros de que esa persona es quien dice ser.

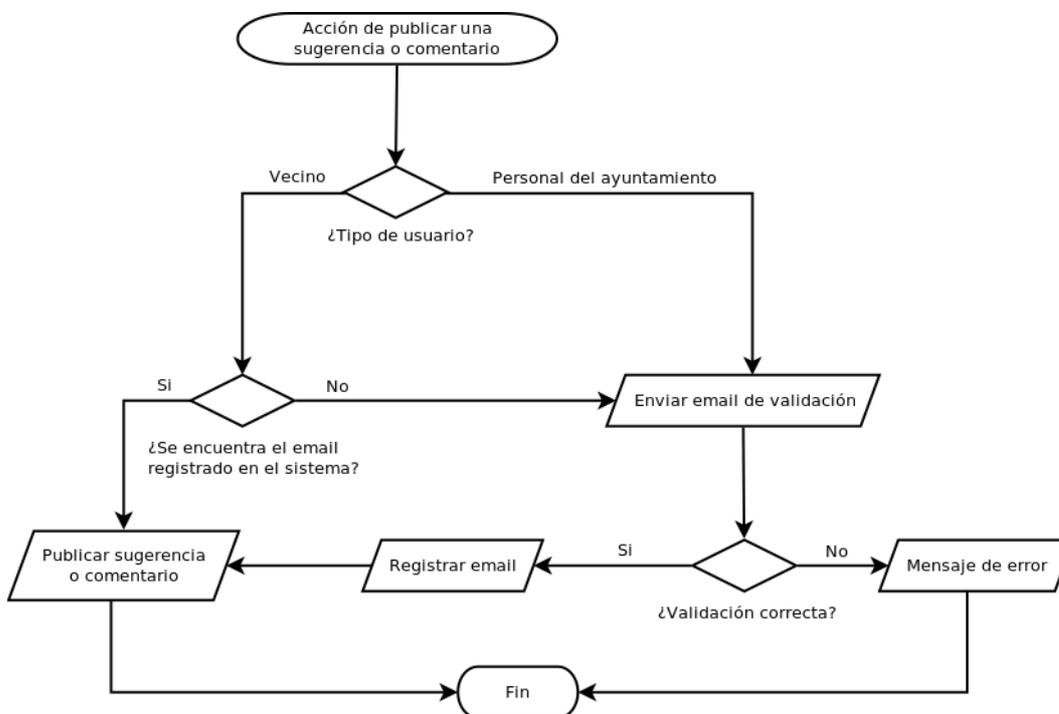
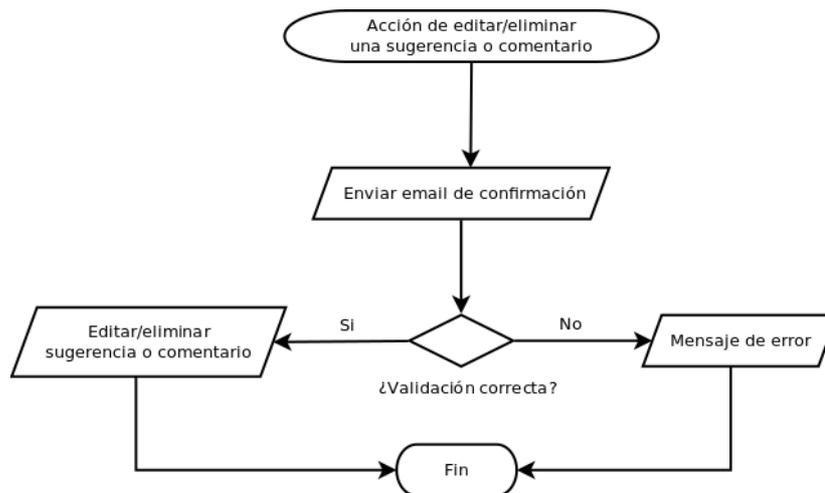


Figura 18: Acción de publicar

### 10.3 Acción de editar o eliminar

Para la edición y el borrado de sugerencias y comentarios siempre se le enviará al propietario de la sugerencia o del comentario un mensaje de confirmación. Con esto evitamos que un usuario malintencionado pueda editar o eliminar sugerencias y comentarios solo con conocer la dirección de correo electrónico del dueño de estos.



*Figura 19: Acción de editar/eliminar*



# Capítulo 11. Autogestión del Buzón

Como ya se ha comentado antes, la idea principal del *Buzón Ciudadano* es que sea autogestionado (dentro de lo posible) para evitar el uso fraudulento del mismo por parte de los ayuntamientos (o la empresa, institución u organización que lo implemente) cuando no quieran, que por ejemplo, se saque a la luz su mala gestión. La idea es que el sistema necesite de la menos intervención humana posible y siempre limitando su poder de acción con respecto al *Buzón Ciudadano*.

## 11.1 Conclusión de las sugerencias

Una de las cosas que se han tenido claras desde el principio es que las sugerencias no podían ser cerradas por el ayuntamiento para evitar el cierre de sugerencias “incomodas”, así como tampoco se ha querido que las sugerencias al cerrarse desaparecieran del sistema. Una buena o mala gestión a la hora de resolver las incidencias por parte de sus responsables tiene que ser visible siempre.

El sistemas de cierre de sugerencias es automático y se lleva a cabo cuando la sugerencia ha entrado en un periodo de inactividad de 30 días. Cuando una sugerencia lleva un mes sin recibir nuevos comentarios se envía un mensaje de correo electrónico a los usuarios que han comentado en la misma (en el caso de que los haya) para informarlos de que pasados 7 días si no hay comentarios nuevos la sugerencia se cerrará. Si pasados 7 días no hay nuevos comentarios la sugerencia se da por concluida informando al responsable del *Buzón Ciudadano* para que pueda informar al responsable del área al que afecte la sugerencia y pueda tomar medidas al respecto. A partir de este momento solo se permiten comentar al personal del ayuntamiento con el fin de informar a los vecinos tanto de si se va a resolver o no la sugerencia como de informar cuando ésta haya sido resuelta.

La idea de hacerlo de este modo es intentar evitar las discusiones que no llevan a ningún lado entre vecinos y ayuntamiento. Mientras la sugerencia permanece abierta todo el mundo puede comentar, pero una vez se cierra solo se le da voz al ayuntamiento para que pueda informar a los usuarios del estado actual de la sugerencia.

En la siguiente figura se puede ver de forma gráfica el flujo que se sigue para cerrar o mantener abierta una sugerencia.

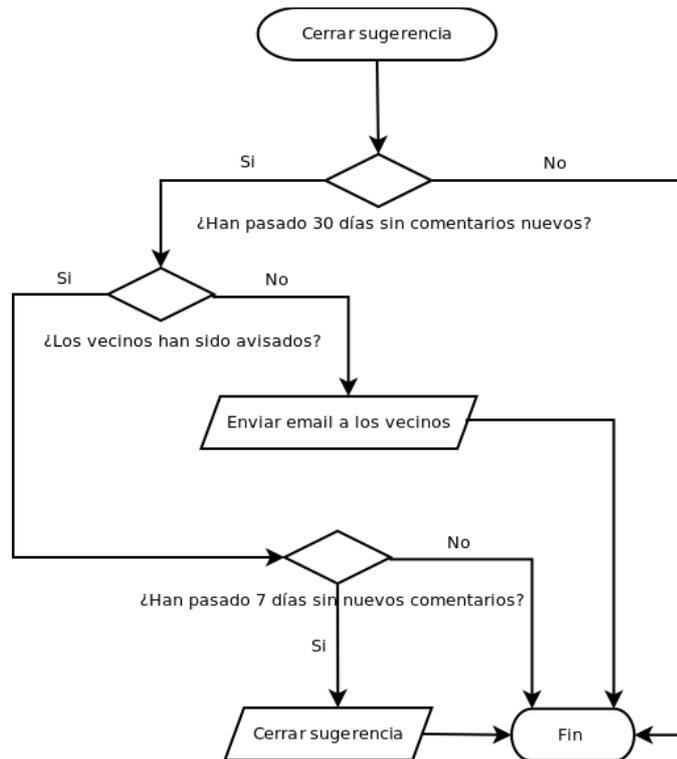


Figura 20: Cerrar sugerencia

Para llevar a cabo el control de las sugerencias que deben cerrarse se ha implementado un script en Ruby que es ejecutado por el servidor de forma periódica todas las noches a las 00:00, que es cuando el servidor tiene menos tráfico y así las sugerencias se cierran de un día para otro.

Cuando una sugerencia se cierra se le añade el icono de un candado cerrado al lado del título para que los usuarios sepan que la sugerencia está cerrada y que no van a poder comentar en ella salvo que sean parte del personal del ayuntamiento.

Las sugerencias cerradas no se eliminan del buzón, sino que permanecen en él a modo de histórico para que los usuarios puedan ver si el ayuntamiento tiene en cuenta o no sus sugerencias, quejas o información sobre incidencias en su localidad.

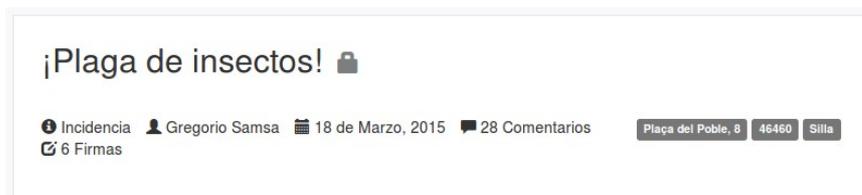


Figura 21: Sugerencia cerrada



# Capítulo 12. Pruebas automáticas

## 12.1 Pruebas unitarias

Una de las partes más importantes de todo desarrollo software y que no debería de faltar nunca son las pruebas (o tests). Las pruebas nos permiten probar las diferentes partes de nuestra aplicación y automatizar este proceso obteniendo información acerca de las mismas. Automatizando las pruebas conseguimos no tener que ejecutar cientos de casos de prueba cada vez que hacemos un cambio en el código para ver si hemos “roto” algo o si por el contrario todo funciona con normalidad, y con un simple clic el sistema pasará las pruebas por nosotros. Por otra parte los tests nos proporcionan una valiosa información, ya que nos permite conocer al instante qué partes de nuestro código han dejado de funcionar debido a alguna modificación.

Para el desarrollo del *Buzón Ciudadano* no he querido centrarme simplemente en los tests unitarios para probar pequeñas partes del código, sino que he creado una batería de pruebas lo suficientemente amplia como para abarcar desde el funcionamiento de la interfaz gráfica (*Feature Tests*), pasando por el funcionamiento de los controladores, hasta llegar a los modelos.

Los tests de la aplicación se encuentran en el directorio *spec* de la raíz del proyecto y para ejecutarlos solamente hay que ejecutar desde un terminal el comando *rspec* estando dentro de la raíz del proyecto.

## 12.2 Integración continua

La integración continua es un modelo propuesto por Martin Fowler que consiste en hacer integraciones automáticas de un proyecto lo más a menudo posible con el fin de poder detectar fallos cuanto antes. La integración consiste en la compilación y ejecución de las pruebas de todo un proyecto.

El proceso de integración continua consiste en descargarse el código fuente del control de versiones, compilarlo, ejecutar las pruebas y generar informes con los resultados. Este proceso se suele hacer cada cierto tiempo (normalmente horas) y está muy ligado a las metodologías ágiles, como *Extreme Programming*.

### 12.2.1 Travis CI

Para llevar a cabo la integración continua durante el desarrollo del *Buzón Ciudadano* se ha utilizado la herramienta Travis CI.

Travis CI es un sistema distribuido de generación e integración continua Open Source que permite conectar nuestro repositorio de GitHub y ejecutar los tests de nuestra aplicación después

de cada push (subida de código al repositorio) que hagamos, generando informes sobre los tests fallidos para que podamos corregir los errores.

Una de las principales ventajas de Travis CI es que su entorno de integración continua está compuesto por múltiples *runtimes* (entornos de ejecución) y sistemas de base de datos que nos permiten probar nuestros programas sobre distintas configuraciones sin tener que tenerlas instaladas localmente. Además no es necesario que estemos pendientes de entrar a la web de Travis CI para ver si nuestro código ha pasado los tests, ya que Travis CI nos envía un correo electrónico con los resultados de las pruebas a los pocos minutos de subir los cambios al repositorio.



## Capítulo 13. Instancia concreta:

### Buzón UPV

Como se ha comentado a lo largo de la memoria, el *Buzón Ciudadano* ha sido pensado para que cualquier tipo de organización, ya sea una empresa, una institución pública o una comunidad de vecinos, pueda conocer las preocupaciones de sus miembros a través de un canal de comunicación público y transparente.

Aunque para el desarrollo del *Buzón Ciudadano* se ha seguido el ejemplo del ayuntamiento por ser el ejemplo más claro y a priori el más interesante por lo que implica el poder hacer públicas tus quejas o sugerencias con respecto a la gestión que lleva a cabo tu ayuntamiento, hemos decidido crear una instancia más concreta del *Buzón Ciudadano* orientada a la Universidad Politécnica de Valencia. La idea sigue siendo la misma que con el ayuntamiento. La UPV como institución interesada en conocer los problemas la comunidad universitaria pone a su disposición el *Buzón UPV* para que puedan expresarse libremente y poder conocer de primera mano que es lo que ocurre en la UPV.

#### 13.1 Funcionamiento del Buzón UPV

Una vez instanciado un caso concreto para el *Buzón Ciudadano* vamos a ver su funcionamiento paso a paso tal y como interactuarían los usuarios con él.

##### 13.1.1 Viendo las sugerencias publicadas

Cuando un usuario entra por primera vez al *Buzón UPV* con lo primero que se encuentra en la página principal es con un listado de todas las sugerencias publicadas por los miembros de la universidad ordenadas por fecha de antigüedad en orden descendente.

La página principal dispone de un filtro de búsquedas para que los usuarios puedan filtrar las sugerencias que quieren ver por algún criterio. Estos criterios son el título de la sugerencia, el tipo (sugerencia, queja, felicitaciones o incidencia), el estado (abiertas, cerradas o ambas), la dirección postal (solo aparecerán sugerencias geolocalizadas) y el perímetro de distancia desde la dirección indicada (por defecto es de 500 metros).

En el listado de sugerencias se muestran los títulos de las sugerencias publicadas, el tipo de la sugerencia, el nombre o alias del autor, la fecha de publicación, el número de comentarios, el número de firmas y la dirección postal de la sugerencia en el caso de que esté geolocalizada.

The screenshot shows the 'Buzón UPV' interface. At the top, there is a navigation bar with the university logo, the text 'UNIVERSITAT POLITÈCNICA DE VALÈNCIA Buzón UPV', and a 'Publica una sugerencia' button. Below this is a large red banner with the university seal and the text 'Buzón UPV' and 'Deja tus sugerencias, comentarios, peticiones y quejas'. A blue 'Publica una sugerencia' button is centered on the banner.

Below the banner is a search filter section titled 'Filtro de búsqueda'. It contains several input fields: 'Título', 'Dirección', 'Tipo' (with a dropdown menu set to 'Todos'), and 'Estado' (with a dropdown menu set to 'Todos'). There is also a 'Distancia' dropdown menu with the text 'Seleccionone una distancia'. At the bottom of the filter section are 'Limpiar' and 'Buscar' buttons.

The main content area displays a list of suggestions. The first suggestion is titled 'Biblioteca abierta 24h en época de exámenes'. It includes metadata: 'Sugerencia' (indicated by an 'i' icon), author 'Sergio92', date '15 de Junio, 2015', '0 Comentarios', and '0 Firmas'. It also shows location tags: 'Camí Vera, 14', '46022', and 'València'. The second suggestion is titled 'Precio de la matrícula demasiado alto'. It includes metadata: 'Queja' (indicated by a 'p' icon), author 'Manolo Garcia', date '30 de Mayo, 2015', '3 Comentarios', and '0 Firmas'.

Figura 22: Lista de sugerencias publicadas

Cuando un usuario hace clic sobre una de estas sugerencias automáticamente es redirigido a la página de la sugerencia donde podrá ver su contenido.

En la cabecera de la sugerencia se muestra la información relativa a la sugerencia, como su título, el tipo de la sugerencia, el nombre del autor (por defecto anonymous), la fecha de publicación, el número de comentarios, el número de firmas y el porcentaje de votos a favor y en contra. El cuerpo de la sugerencia está formado por la descripción de la misma, un apartado dónde se podrán ver las imágenes subidas por el autor (opcional) y un apartado con la dirección a la que afecta la sugerencia dibujada sobre un mapa (opcional). Si hacemos clic sobre las imágenes éstas se mostrarán en grande para su mejor visualización.

En la parte inferior de la sugerencia tenemos el listado con los comentarios de los usuarios junto con su voto (o punto de vista sobre la sugerencia). Los comentarios con el fondo blanco corresponden a comentarios de los alumnos y los comentarios con el fondo amarillo son los comentarios realizados por el personal de la universidad.

# Sugerencia

Si piensas que la sugerencia es importante, deja una firma o un comentario

## Biblioteca abierta 24h en época de exámenes

 Sugerencia
  Sergio92
  15 de Junio, 2015
  5 Comentarios

 2 Firmas



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed mauris augue, porttitor ac ullamcorper in, imperdiet non sem. Nullam quis lacus posuere, molestie augue eu, scelerisque orci. Vivamus interdum dapibus venenatis. Fusce eget ante neque. Nulla facilisi. Cras sollicitudin id nunc ac efficitur. Integer in gravida nisi. In sed lacinia nisl, sit amet viverra nulla. Vivamus sed enim sed lacus gravida scelerisque a nec orci.

Suspendisse ultrices lorem commodo interdum finibus. In at risus eu orci sodales rhoncus sit amet et tellus. Sed erat eros, tempor a porttitor id, accumsan eget leo. Vestibulum euismod sollicitudin erat nec mollis. Sed nec vestibulum sem. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque eleifend eget erat eget iaculis. Donec porttitor dolor eu sapien pretium, ut dignissim arcu porttitor. Etiam lacus urna, finibus in porttitor et, semper vitae nisl.

Editar

Eliminar

  Tamaño texto
  Reportar

## Deja un comentario

Nombre

Email

Comentario

## Vota la sugerencia

Seleccione una opción

Firma y comenta

Comenta

## Imágenes



## Geolocalización

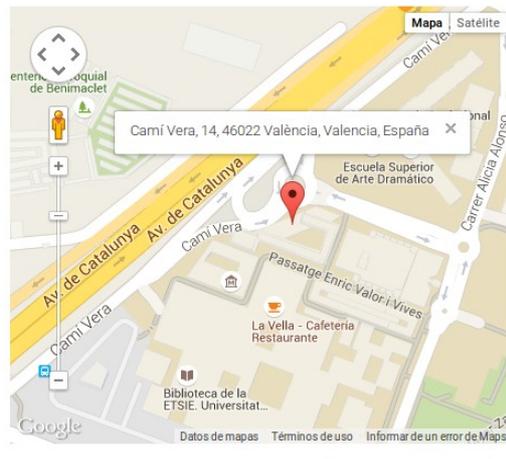


Figura 23: Parte alta de la sugerencia

## Comentarios

Luis\_Miguel 

29 de Junio, 2015

Integer in gravida nisi. In sed lacinia nisl, sit amet viverra nulla. Vivamus sed enim sed lacus gravida scelerisque a nec orci.

 Eliminar

 Reportar

Enrique88 

30 de Junio, 2015

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed mauris augue, porttitor ac ullamcorper in, imperdiet non sem. Nullam quis lacus posuere, molestie augue eu, scelerisque orci.

 Eliminar

 Reportar

Ana 

30 de Junio, 2015

Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque eleifend eget erat eget iaculis. Donec porttitor dolor eu sapien pretium, ut dignissim arcu porttitor.

 Eliminar

 Reportar

JuanUPV (Personal UPV)

30 de Junio, 2015

Proin libero purus, maximus rutrum feugiat vitae, iaculis ut velit. Cras tempor porttitor orci quis vestibulum. Mauris eu ultrices sapien. Phasellus eu lectus tortor.

 Eliminar

 Reportar

Maria94 

30 de Junio, 2015

Fusce eget ante neque. Nulla facilisi. Cras sollicitudin id nunc ac efficitur. Integer in gravida nisi. In sed lacinia nisl, sit amet viverra nulla. Vivamus sed enim sed lacus gravida scelerisque a nec orci.

 Eliminar

 Reportar

Cambiar idioma ▾

Información

[Contacto](#) [Sobre el Buzón UPV](#) [Condiciones y privacidad](#)



© Buzón UPV, All rights reserved

Figura 24: Parte baja de la sugerencia

### 13.1.2 Comentando/firmando sugerencias como alumno

Cuando un alumno encuentra una sugerencia que le parece interesante y quiere aportar su punto de vista puede hacerlo desde el formulario de publicación de comentarios que hay en la ficha de cada sugerencia. En la siguiente figura puede verse un ejemplo de comentario realizado por un alumno cualquiera.

Deja un comentario

Amapola Silvestre  
amapola@silvestre.com

Praesent ac ultrices lectus, nec fringilla justo. Phasellus lobortis risus quis nibh molestie, viverra rhoncus velit tincidunt. Quisque nec elementum felis.

Vota la sugerencia ⓘ

Abstenerse

Firma y comenta

Comenta

Figura 25: Formulario de comentarios

Tanto la acción de comentar como la de firmar crean un comentario nuevo en la sugerencia. La diferencia es que al firmar el usuario firmante será avisado por correo electrónico de toda la actividad de la sugerencia, ya que entendemos que si se ha tomado la molestia de dar su apoyo a la sugerencia es que le interesa estar al tanto de lo que ocurra en la misma. Un usuario puede comentar todas las veces que quiera en una misma sugerencia, pero solo se permite una única firma por usuario en una misma sugerencia.

Si el usuario ya ha publicado sugerencias o comentarios en el pasado su nuevo comentario se publicará automáticamente. Si el usuario no ha utilizado antes el *Buzón UPV* o lo hace con una cuenta de correo electrónico nueva entonces el sistema le avisará que antes de que el comentario se publique deberá validarlo. Este paso de validación es obligatorio y sirve para comprobar que el usuario es una persona real y no un robot que quiere publicar *spam* en nuestro sitio web.

En breves momentos recibirá un email para confirmar su comentario x

Biblioteca abierta 24h en época de exámenes

📍 Sugerencia 👤 Sergio92 📅 15 de Junio, 2015 💬 5 Comentarios  
📝 2 Firmas

👍 🔄

Progress bar: 50% (blue) / 50% (red)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed mauris augue, porttitor ac ullamcorper in, imperdiet non sem. Nullam quis lacus posuere, molestie augue eu, scelerisque orci. Vivamus interdum

Deja un comentario

Nombre

Email

Comentario

Figura 26: Mensaje email enviado

A los pocos segundos de crear el comentario el usuario recibirá un mensaje de correo electrónico como el siguiente en el que se le informa que debe pulsar el enlace de validación para que su comentario sea publicado en la sugerencia y sea visible por el resto de la comunidad universitaria. Con esto nos evitamos que los alumnos utilicen cuentas de correo electrónico falsas o que no les pertenecen.



## Bienvenido al Buzón UPV, Amapola Silvestre

Has creado un nuevo comentario. Para que sea visible debes validarlo pulsando el siguiente enlace:

[Enlace de validacion](#)

Gracias por utilizar el Buzón UPV

Figura 27: Email validación nuevo comentario

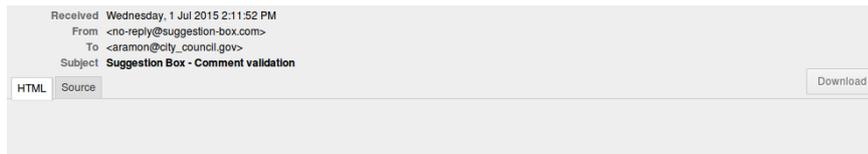
Cuando el alumno haga clic sobre el enlace de validación será redirigido a la sugerencia dónde se le informará que su comentario ha sido publicado y podrá verlo al final de la cola de comentarios existentes.



Figura 28: Mensaje comentario publicado

### 13.1.3 Comentando sugerencias como personal de la UPV

El proceso es exactamente el mismo que el de comentar como alumno, las únicas diferencias son que el personal de la UPV no puede firmar las sugerencias y que siempre se le solicitará validar la sugerencia para evitar que alguien pueda comentar como personal de la UPV sin serlo. El personal de la UPV no está obligado a dar su opinión sobre las sugerencias de los usuarios, sin embargo es de agradecer que se tomen la molestia de comentar como personal de la UPV dando su propia opinión al respecto y por este motivo se les da las gracias en el mensaje de correo electrónico que reciben.



## Buzón UPV

Hola Andrés Ramón,

Gracias por tomarte la molestia de dejar un comentario en la sugerencia "Biblioteca abierta 24h en época de exámenes"

Para validar el comentario y hacerlo visible haz click en el siguiente enlace:

[Enlace de validacion](#)

Gracias por comentar

*Figura 29: Email validación comentario personal UPV*

En la figura 30 se puede ver el comentario de una persona de la UPV. Estos comentarios tienen el fondo amarillo y pone la palabra *Personal UPV* al lado del nombre del usuario para que el resto de usuarios puedan identificarlos a primera vista.

### 13.1.4 Eliminando comentarios

Puede darse el caso de que un usuario o algún trabajador de la UPV desee eliminar un comentario de su autoría debido a diversos motivos. Por ejemplo, la alumna de Ingeniería Informática, Amapola Silvestre, se ha dado cuenta de que se le olvidó comentar algo importante en su comentario y quiere eliminarlo para más tarde publicar uno nuevo más completo. Para eliminarlo solo tiene que pulsar el icono con forma de papelera que hay bajo cada comentario.

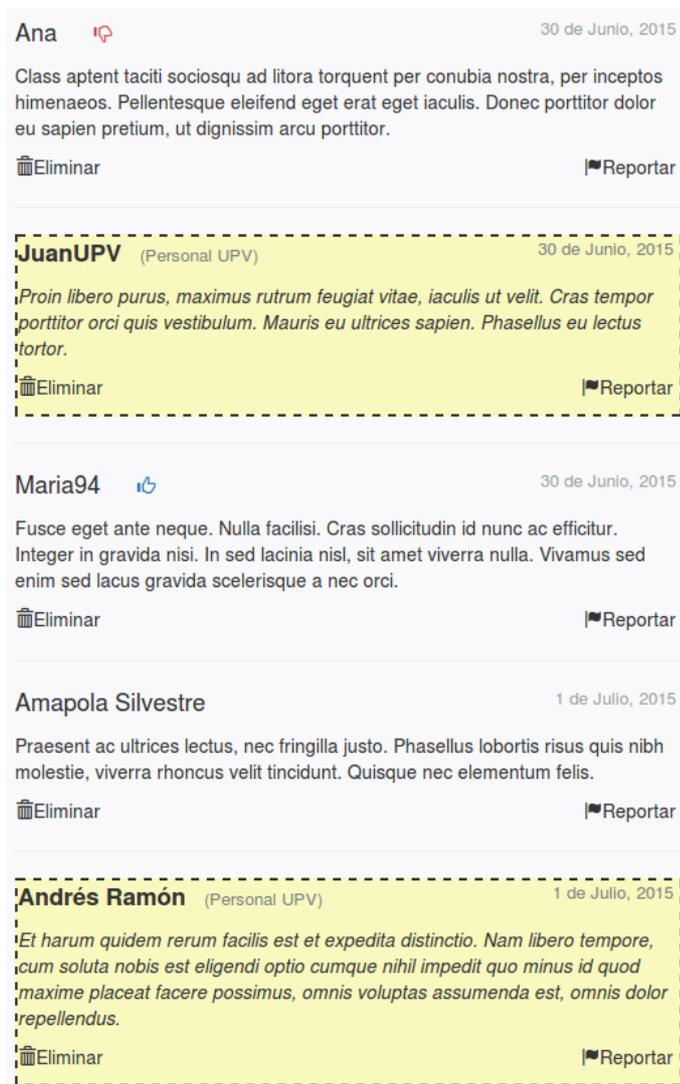


Figura 30: Comentarios antes de eliminar

Automáticamente se le notificará al usuario con un mensaje en pantalla de que se le ha enviado un mensaje a su cuenta de correo electrónico con el enlace de confirmación. La confirmación para eliminar un comentario siempre es obligatoria y de esta forma evitamos que cualquier persona pueda eliminar comentarios a su antojo.

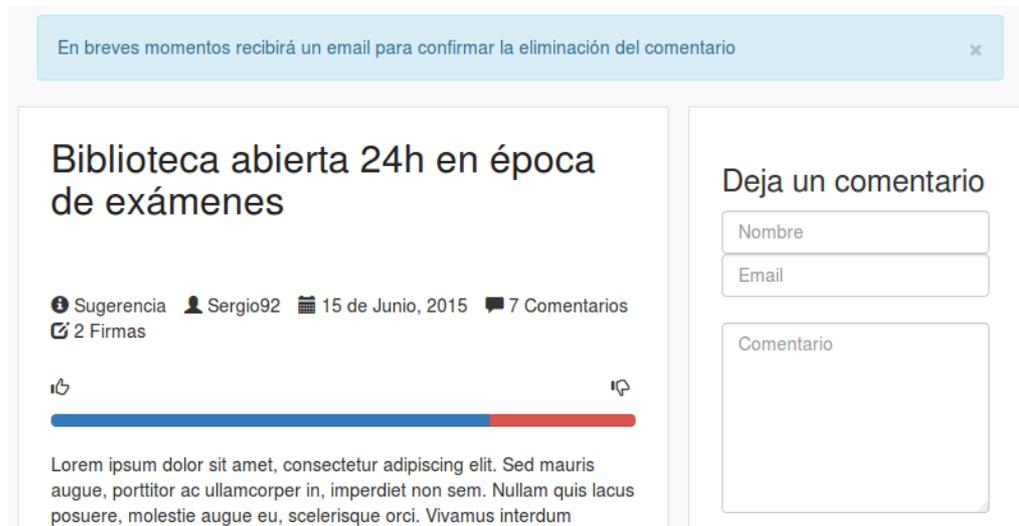
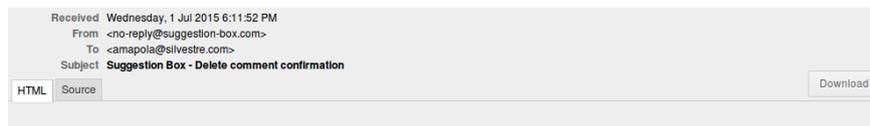


Figura 31: Mensaje eliminar comentario



## Buzón UPV

Hola Amapola Silvestre, hemos recibido una petición para eliminar el siguiente comentario de la sugerencia "Biblioteca abierta 24h en época de exámenes"

**Comentario:** Praesent ac ultrices lectus, nec fringilla justo. Phasellus lobortis risus quis nibh molestie, viverra rhoncus velit tincidunt. Quisque nec elementum felis.

Si deseas eliminar el comentario de forma definitiva haz click en el siguiente enlace. En caso contrario ignora este email.

[Eliminar comentario](#)

Gracias por utilizar el Buzón UPV

Figura 32: Email eliminar comentario

Cuando el usuario pulse sobre el enlace de confirmación que contiene el mensaje de correo electrónico que se le ha mandado a su cuenta el comentario será eliminado del sistema y desaparecerá de la sugerencia.



Figura 33: Mensaje comentario eliminado

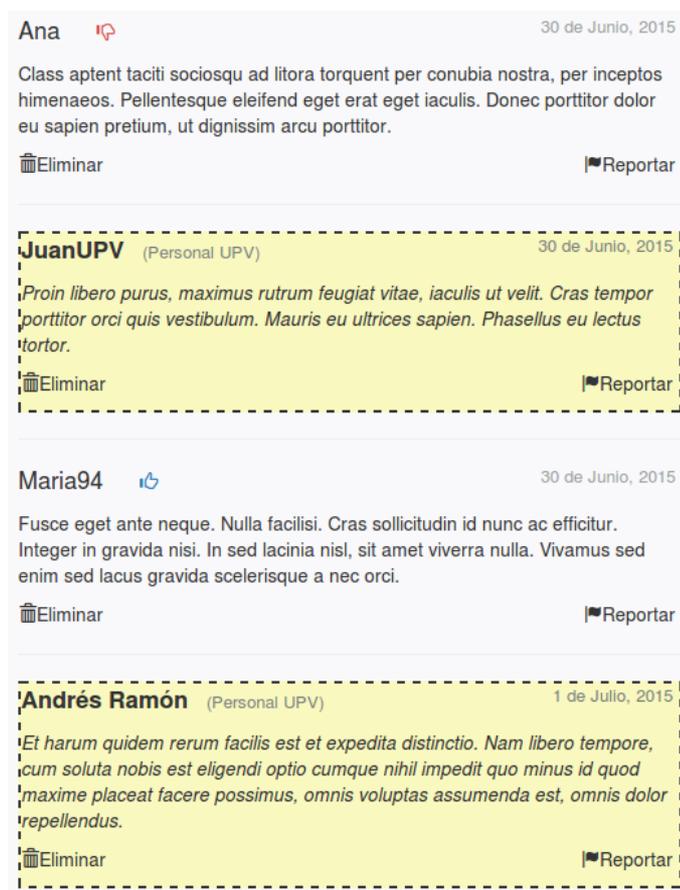


Figura 34: Comentarios después de eliminar

### 13.1.5 Publicando una sugerencia

Tanto los usuarios de la comunidad universitaria como el personal de la UPV pueden publicar sugerencias sin que estas se diferencien unas de otras como ocurre con los comentarios. Esto se debe a que se ha querido preservar ante todo el anonimato de todos los usuarios del buzón para que la utilización del mismo no conlleve ningún tipo de represalia por parte de otro usuario o de la misma institución donde el usuario reside, estudia o trabaja.

El proceso para publicar una sugerencia es muy sencillo y solo consta de un formulario dónde el usuario tiene que introducir de forma obligatoria el título de la sugerencia, el tipo, su dirección de correo electrónico y el comentario de la sugerencia. El campo del nombre del autor y las opciones para subir imágenes y geolocalizar la sugerencia son opcionales y depende del usuario su utilización.

En la siguiente figura se puede comprobar el estado de una sugerencia rellena por un usuario cualquiera de la UPV y lista para ser publicada.

# Nueva Sugerencia

Crea una nueva sugerencia sobre un problema, una solicitud, una queja... o cualquier cosa que creas que es importante

## Título

## Tipo

## Autor

## Email

## Comentario









Li European lingues es membres del sam familie. Lor separat existentie es un myth. Por scientie, musica, sport etc, litot Europa usa li sam vocabular. Li lingues differe solmen in li grammatica, li pronunciation e li plu commun vocabules. Omnics directe al desirabilite de un nov lingua franca: On refusa continuar payar custosi traductores. At solmen va esser necessari far uniform grammatica, pronunciation e plu sommun paroles. Ma quande lingues coalesce, li grammatica del resultant lingue es plu simplic e regulari quam ti del coalescent lingues.

## Adjuntar Imágenes

Si lo deseas puedes adjuntar imágenes que aporten más información a la sugerencia.






## Geolocaliza la sugerencia

Pon una marca haciendo click en el mapa o busca una dirección para geolocalizar la sugerencia. Si no quieres geolocalizar la sugerencia elimina la marca del mapa.



Figura 35: Nueva sugerencia

Si la dirección de correo electrónico del usuario ya se encuentra dada de alta en el sistema la sugerencia se publicará automáticamente sin ningún tipo de confirmación, en caso contrario el usuario será redirigido a la ficha de su sugerencia y se le avisará de que debe confirmarla a través del mensaje que se le ha enviado a su cuenta de correo para que ésta sea visible para el resto de los usuarios. En este punto la sugerencia no aparece en el listado de la página principal ni tampoco es posible publicar nuevos comentarios. Se trata de una vista que solo el usuario que la ha publicado puede ver.



The screenshot shows a notification banner at the top: "En breves momentos recibirá un email para confirmar su sugerencia" with a close button. Below it is a card for a suggestion titled "Bancos rotos en la zona verde de enfrente de rectorado". The card includes metadata: "Incidencia" (Incident), "Enrique Carrillo" (author), "1 de Julio, 2015" (date), "0 Comentarios" (0 comments), and "0 Firmas" (0 signatures). There are share and report icons. The main text of the suggestion is in Spanish and discusses European languages and a simplified language. At the bottom, there are buttons for "Editar" (Edit) and "Eliminar" (Delete), along with "Tamaño texto" (Text size) and "Reportar" (Report) options.

Figura 36: Sugerencia sin validar

El mensaje que se envía a la dirección de correo electrónico del usuario es similar al mensaje de validación de nuevos comentarios, y tras validar la sugerencia, el usuario será redirigido a la misma y se le informará que su sugerencia ha sido validada y publicada, como se puede ver en la figura 38.



## Bienvenido al Buzón UPV, Enrique Carrillo

Has creado una nueva sugerencia. Para que sea visible debes validarla pulsando el siguiente enlace:

[Enlace de validacion](#)

Gracias por usar el Buzón UPV

Figura 37: Email validación sugerencia

Una vez la sugerencia sea validada por el usuario será visible desde la página principal y cualquier miembro de la universidad podrá verla y realizar nuevos comentarios.

La sugerencia ha sido validada y publicada ×

### Bancos rotos en la zona verde de enfrente de rectorado

📌 Incidencia 👤 Enrique Carrillo 📅 1 de Julio, 2015  
💬 0 Comentarios ✍️ 0 Firmas

🔖 🔒

---

Li European lingues es membres del sam familie. Lor separat existentie es un myth. Por scientie, musica, sport etc, litot Europa usa li sam vocabular. Li lingues differe solmen in li grammatica, li pronunciation e li plu commun vocabules. Omnico directe al desirabilite de un nov lingua franca: On refusa continuar pagar custosi traductores. At solmen va esser necessari far uniform grammatica, pronunciation e plu sommun paroles. Ma quande lingues coalesce, li grammatica del resultant lingue es plu simplic e regulari quam ti del coalescent lingues.

Li nov lingua franca va esser plu simplic e regulari quam li existent European lingues. It va esser tam simplic quam Occidental in fact, it va esser Occidental. A un Angleso it va semblar un simplificat Angles, quam un skeptic Cambridge amico dit me que Occidental es. Li European lingues es membres del sam familie.

Editar Eliminar
🔍 Tamaño texto 🚩 Reportar

#### Deja un comentario



Comentario

**Vota la sugerencia** ⓘ

Firma y comenta

Figura 38: Sugerencia validada y publicada

### 13.1.6 Editando una sugerencia

Debido al enfoque *One Less Password* que se ha seguido para la autenticación de los usuarios cualquier persona puede solicitar la edición de cualquier sugerencia. Sin embargo, solamente el autor de la sugerencia va a poder editarla, ya que se le va a mandar un mensaje a su cuenta de correo electrónico con el enlace que lo autentificará ante el sistema y le dará acceso a la edición de su sugerencia.

Supongamos que un usuario que acaba de publicar una sugerencia quiere editarla para geolocalizarla y que tantos los usuarios como la universidad puedan ver donde se encuentra ubicada físicamente la incidencia a la que hace referencia su sugerencia. Además quiere aportar fotografías tomadas con su *smartphone* para que se vea mejor el estado de los bancos rotos sobre los que informa en su sugerencia y que se vea la ubicación exacta de los mismo. En este caso lo único que el usuario tendría que hacer es ir a su sugerencia (Figura 38) y pulsar el botón de edición.

Al igual que con el resto de acciones, el sistema le informará al usuario de que en breves momentos recibirá un mensaje de correo electrónico para confirmar la edición de la sugerencia.

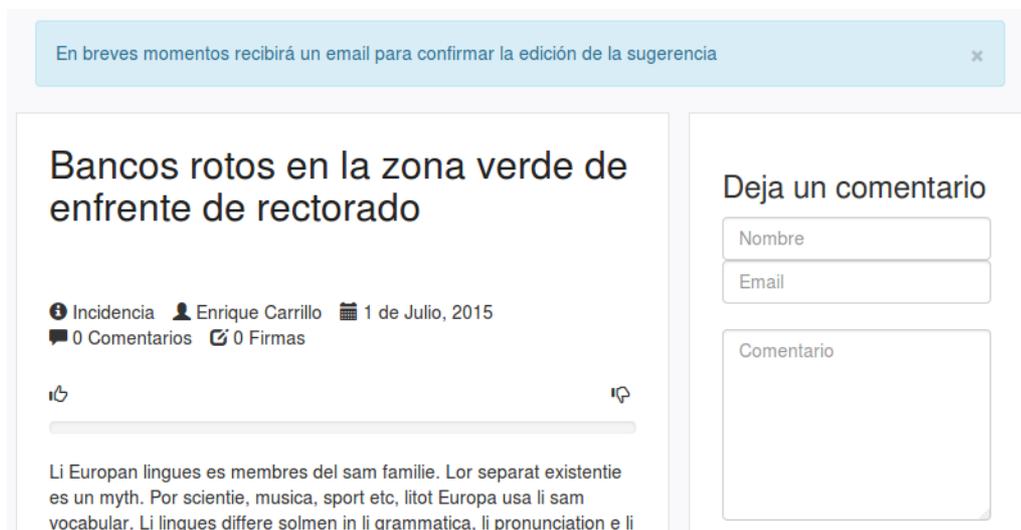
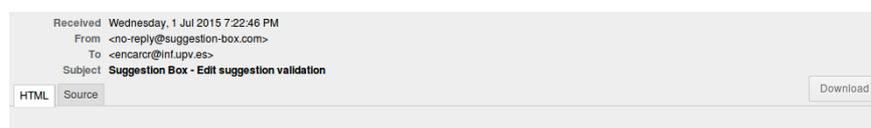


Figura 39: Mensaje edición sugerencia

Acto seguido el usuario recibirá en su cuenta de correo electrónico un mensaje como el siguiente donde se le informa de que ha solicitado editar su sugerencia y dónde se le proporciona un enlace de confirmación con acceso directo a la edición de la sugerencia.



#### Buzón UPV

Hemos recibido una petición de edición de la sugerencia "Bancos rotos en la zona verde de enfrente de rectorado"

Para continuar con la edición haga click en el siguiente enlace:

[Editar sugerencia](#)

Gracias por utilizar el Buzón UPV

Figura 40: Email información edición sugerencia

Cuando el usuario haga clic sobre el enlace de edición de sugerencia automáticamente será redirigido a la ventana de edición de la sugerencia, donde podrá añadir la geolocalización de su sugerencia y las fotografías que ha tomado.

**Titulo** Bancos rotos en la zona verde de enfrente de rectorado **Tipo** Incidencia

**Autor** Enrique Carrillo **Email** encarcrc@inf.upv.es

**Comentario**

**Bold** *Italic* Underline [List] [Table] [Link] [Image] [Undo]

Li European lingues es membres del sam familie. Lor separat existentie es un myth. Por scientie, musica, sport etc, lliot Europa usa li sam vocabular. Li lingues difere solmen in li grammatica, li pronunciation e li plu commun vocabules. Omnico directe al desirabilite de un nov lingua franca: On refusa continuar pagar custoso traductores. At solmen va esser necessari far uniform grammatica, pronunciation e plu sommun paroles. Ma quande lingues coalesce, li grammatica del resultant lingue es plu simplic e regulari quam ti del coalescent lingues.

**Adjuntar Imágenes**

Si lo deseas puedes adjuntar imágenes que aporten más información a la sugerencia.

[Image 1] [Image 2]

[Modificar] [Eliminar] [Modificar] [Eliminar]

**Geolocaliza la sugerencia**

Pon una marca haciendo click en el mapa o busca una dirección para geolocalizar la sugerencia. Si no quieres geolocalizar la sugerencia elimina la marca del mapa.

Carrer Paul Samuelson, 46022 València, Valencia, España

[Mapa] [Satélite]

[Publicar Sugerencia] [Volver]

Figura 41: Edición de sugerencia

Una vez confirmados los cambios en la sugerencia ésta será publicada y los usuarios podrán ver las nuevas imágenes y consultar en el mapa la ubicación exacta de la incidencia.

# Sugerencia

Si piensas que la sugerencia es importante, deja una firma o un comentario

La sugerencia ha sido actualizada

## Bancos rotos en la zona verde de enfrente de rectorado

 Incidencia
  Enrique Carrillo
  1 de Julio, 2015  
 0 Comentarios
  0 Firmas




Li European lingües es membres del sam familie. Lor separat existentie es un myth. Por scientie, musica, sport etc, liot Europa usa li sam vocabular. Li lingües differe solmen in li grammatica, li pronunciation e li plu commun vocabules. Omnicos directe al desirabilite de un nov lingua franca: On refusa continuar pagar custosi traductores. At solmen va esser necessari far uniform grammatica, pronunciation e plu sommun paroles. Ma quande lingües coalesce, li grammatica del resultant lingüe es plu simplic e regulari quam ti del coalescent lingües.

Li nov lingua franca va esser plu simplic e regulari quam li existent European lingües. It va esser tam simplic quam Occidental in fact, it va esser Occidental. A un Angleso it va semblar un simplificat Angles, quam un skeptik Cambridge amico dit me que Occidental es. Li European lingües es membres del sam familie.

Editar **Eliminar**
 Tamaño texto
  Reportar

### Deja un comentario




### Vota la sugerencia




### Imágenes



### Geolocalización

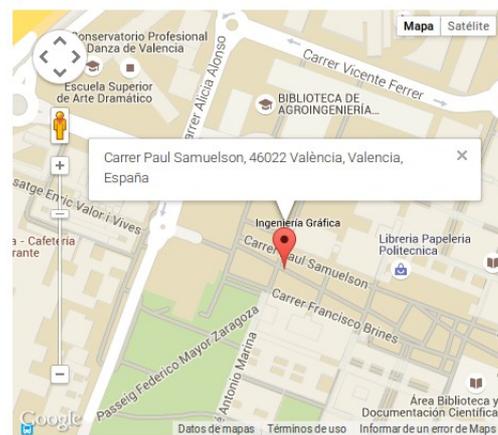


Figura 42: Sugerencia actualizada

### 13.1.7 Eliminando sugerencias

El proceso para eliminar una sugerencia es similar al de editarla. El dueño de la misma tiene que pulsar el botón de eliminar (Ver figura 42) y acto seguido se le enviará un mensaje de confirmación a su cuenta de correo electrónico.

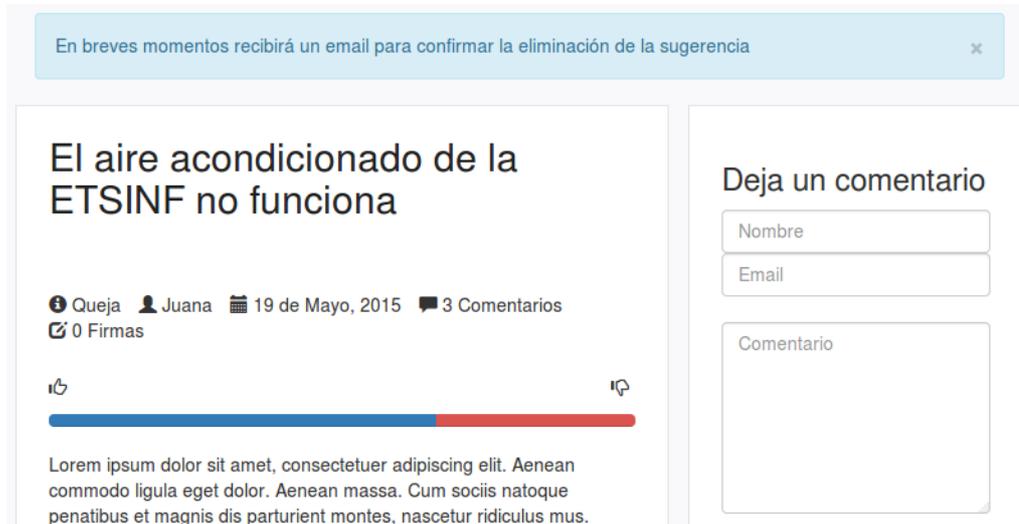
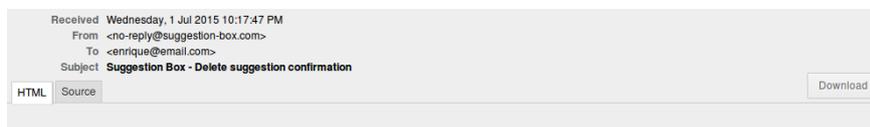


Figura 43: Mensaje eliminar sugerencia

Al hacer clic sobre el enlace para eliminar la sugerencia ésta será eliminada automáticamente y el usuario será redirigido a la página principal del *Buzón UPV* dónde se le notificará de que su sugerencia ha sido eliminada.



#### Buzón UPV

Hola Juana, hemos recibido una petición para eliminar la sugerencia "El aire acondicionado de la ETSINF no funciona". Si deseas eliminar la sugerencia de forma definitiva haz click en el siguiente enlace. En caso contrario ignora este email.  
[Eliminar sugerencia](#)  
 Gracias por utilizar el Buzón UPV

Figura 44: Email eliminar sugerencia



# Buzón UPV

Deja tus sugerencias, comentarios, peticiones y quejas

Publica una sugerencia

La sugerencia ha sido eliminada

## Filtro de búsqueda

Título

Dirección

Tipo

Todos

Estado

Todos

Distancia

Seleccionone una distancia

Limpiar

Buscar

## Bancos rotos en la zona verde de enfrente de rectorado

 Incidencia  Enrique Carrillo  1 de Julio, 2015  0 Comentarios

 Carrer Paul Samuelson  46022  València

 0 Firmas

## Biblioteca abierta 24h en época de exámenes

 Sugerencia  Sergio92  15 de Junio, 2015  7 Comentarios

 Cami Vera, 14  46022  València

 2 Firmas

## Precio de la matrícula demasiado alto

 Queja  Manolo Garcia  30 de Mayo, 2015  3 Comentarios  0 Firmas

Figura 45: Mensaje sugerencia eliminada

### 13.1.8 Reportando sugerencias y comentarios

Para evitar que un usuario malintencionado pueda crear sugerencias y comentarios con contenido publicitario o fraudulento, el sistema permite a los usuarios reportar las sugerencias y los comentarios para informar a los responsables del buzón y que puedan tomar medidas en el caso de que el contenido de la sugerencia o del comentario reportado viole las normas del *Buzón UPV*.

Como el funcionamiento de reportar es exactamente el mismo tanto para las sugerencias como para los comentarios vamos a ver cuál sería su funcionamiento para éstos últimos.

Supongamos que un miembro cualquiera de la UPV se mete en una sugerencia y leyendo los comentarios de la misma se encuentra con el siguiente comentario.

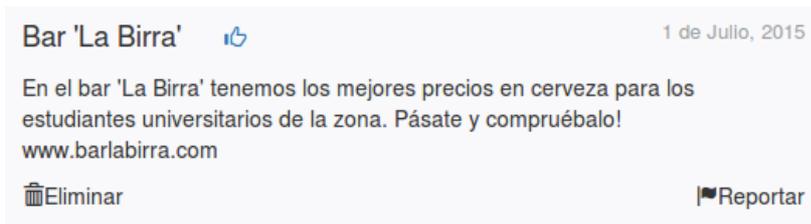
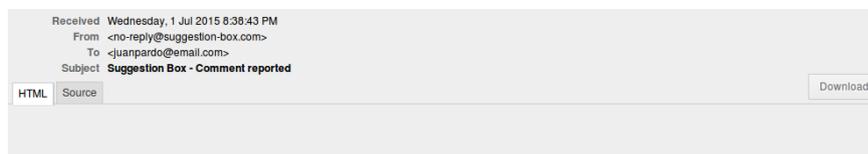


Figura 46: Comentario spam

En este caso el usuario solo tiene que pulsar el enlace de reportar que hay debajo del comentario y automáticamente le llegará un mensaje de correo electrónico al responsable del *Buzón UPV* informándole de que un usuario de la UPV ha reportado el comentario y que en el caso de que el comentario viole alguna de las normas del mismo debe ser eliminado.



#### Buzón UPV

Hola Juan

El siguiente comentario ha sido reportado por un usuario

**Autor:** Bar 'La Birra'

**Comentario:** En el bar 'La Birra' tenemos los mejores precios en cerveza para los estudiantes universitarios de la zona. Pásate y compruébalo! www.barlabirra.com

Si el comentario no incumple las normas del Buzón UPV puedes ignorar este mensaje, en caso contrario el comentario debe ser eliminado.

[Ir a la sugerencia](#)

Figura 47: Email comentario reportado

## Capítulo 14. Conclusiones

El objetivo de desarrollar un buzón de sugerencias moderno, social y adaptado a los nuevos tiempos que corren se ha cumplido, llegando incluso a ampliarlo en varias ocasiones con funcionalidades que no estaban previstas en un principio y que lo dotan de mayor atractivo.

La realización del presente trabajo de fin de grado me ha permitido adentrarme en el desarrollo de aplicaciones web y poder aprender nuevas tecnologías que espero me sirvan una vez obtenga el título de Graduado en Ingeniería Informática.

Al inicio del proyecto apenas tenía algunos conocimientos muy básicos de programación web y no estaba al tanto de las nuevas tecnologías que se utilizan en la actualidad. Realizar este trabajo de fin de grado me ha supuesto, por una parte, aprender un fantástico framework que si bien no es de los más usados, si es de los más solicitados por las empresas debido a la falta de gente con experiencia. Por otra parte me ha hecho crecer como programador y he aprendido nuevas técnicas relacionadas con la programación web que yo desconocía, además de poner en práctica las metodologías ágiles en el desarrollo de un proyecto de software real.

Quizás lo más difícil, y que en ocasiones me ha frustrado un poco, ha sido el hecho de enfrentarme a lenguajes y tecnologías totalmente desconocidas para mí. Sin contar los meses de aprendizaje y estudio antes de empezar con el desarrollo del *Buzón Ciudadano*. El hecho de enfrentarme con tecnologías nuevas me ha hecho perder el miedo a lo desconocido y madurar como programador.

En resumen, creo que la elección de este proyecto ha sido acertada porque me ha permitido suplir las carencias de la carrera en cuanto al desarrollo web se refiere y además es una de las profesiones más demandadas en la actualidad.



## Capítulo 15. Trabajo futuro

Aunque el objetivo de desarrollar una aplicación totalmente funcional se ha cumplido de sobra (incluso se han incluido funcionalidades que no estaban planificadas en un principio), la aplicación aún tiene mucho trabajo por delante y se han dejado varias funcionalidades interesantes sin implementar por no alargar demasiado el desarrollo del *Buzón Ciudadano*.

Las principales funcionalidades que se han quedado pendientes son las comentadas en el apartado 7.2 del capítulo 7. Estas funcionalidades ya estaban pensadas y puestas en el backlog del proyecto a la espera de ser definidas y pasadas a producción, pero debido a la falta de tiempo no se han podido implementar.

Una de las principales funcionalidades que ha sido imposible implementar por tratarse de un proyecto personal es la validación de usuarios a partir de las bases de datos de la empresa, institución u organización que ponga a disposición de sus usuarios el *Buzón Ciudadano*. La idea es que solo los clientes de una determinada empresa, los vecinos de una población o los estudiantes de una universidad puedan utilizar el buzón.

Una funcionalidad importante que se ha implementado a medias es el control de sugerencias y comentarios fraudulentos o inapropiados. La idea era implementar un sistema de votos negativos de forma que al llegar a un determinado número de votos el sistema borrara u ocultara las sugerencias o comentarios fraudulentos. Sin embargo, al no tener unas estadísticas de uso reales del *Buzón Ciudadano* no sabemos cual sería el número de votos idóneos para llevar a cabo este control, ni si sería un método adecuado, por lo que se ha optado por que los usuarios reporten las sugerencias y comentarios que crean que violan las normas del *Buzón Ciudadano* y el responsable del buzón será el encargado de revisar estas sugerencias y comentarios reportados y eliminarlos cuando sea necesario.

Además de las funcionalidades ya planificadas que han quedado pendientes, también surgieron nuevas funcionalidades que no fueron incluidas en el backlog del proyecto por falta de tiempo para desarrollarlas y que vamos a detallar a continuación.

Un aspecto importante relacionado con la mantenibilidad y la legibilidad del código de las vistas de la aplicación, y que surgió en la recta final del TFG, es el uso de algún motor de plantillas que permita deshacernos del anticuado lenguaje HTML y la inclusión de código “del lado servidor” embebido en las mismas. Como trabajo futuro se ha pensado en la utilización de HAML (*HTML Abstraction Markup Language*) para describir el formato de las vistas de la aplicación sin emplear el código embebido tradicional. HAML sigue el principio DRY (*Don't Repeat Yourself*) y evita la duplicación de las etiquetas HTML (etiqueta de apertura y cierre) además de evitar el código embebido en las mismas. Con esto se consigue un código más limpio, legible, mantenible y sobre todo menos propenso a errores debido a su marcado sangrado.

También ha quedado pendiente la detección inteligente de *spam* para evitar que los cientos de robots que recorren Internet publiquen publicidad en las sugerencias o en los comentarios y mantenerlo de esta forma limpio.

Otro aspecto que queda como trabajo futuro es la compatibilidad de la aplicación con dispositivos móviles. Aunque la utilización del framework Bootstrap nos garantiza que nuestra aplicación se va a ver bien en cualquier dispositivo independientemente del tamaño su pantalla, hay muchas funcionalidades que no funcionan en este tipo de dispositivos debido a la limitación de sus navegadores web. No todos los navegadores web móviles son compatibles con tecnologías como Ajax o con ciertas funcionalidades de jQuery o JavaScript, lo que hace que ciertas funcionalidades no funcionen correctamente o que su utilización en este tipo de dispositivos no sea igual de satisfactoria que en la versión de escritorio. Para solucionar estos problemas sería interesante desarrollar una aplicación web exclusivamente para móviles de forma que su usabilidad fuese exactamente la misma que en su versión de escritorio.



## Bibliografía

- [1] **COOPER, Peter** [2009]. *Beginning Ruby: From Novice to Professional*, 2ªed. Apress.
- [2] **HARTL, Michael** [2013]. *Ruby on Rails Tutorial: Learn Web Development with Rails*, 2ªed. Addison-Wesley.
- [3] **RUBY, Sam, THOMAS, Dave y HEINEMEIER HANSSON, David** [2013]. *Agile Web Development with Rails 4*, 4ªed. Pragmatic Bookshelf.
- [4] **HUNT, Andrew y THOMAS, David** [1999]. *The Pragmatic Programmer: From Journeyman to Master*. Addison-Wesley.
- [5] **OTTO, Mark y THORNTON, Jacob** [2013]. *Bootstrap 3: Manual Oficial*. <[http://librosweb.es/bootstrap\\_3/](http://librosweb.es/bootstrap_3/)>. LIBROSWEB
- [6] **SAWYER McFARLAND, David** [2014]. *JavaScript & jQuery: The Missing Manual*, 3ªed. O'Reilly.
- [7] **Ruby on Rails Guides** [online]. <http://guides.rubyonrails.org/>
- [8] **Rails for Zombies**. *Learn Rails the Zombie Way* [online]. <http://railsforzombies.org/>
- [9] **Rspec Manual** [online]. <https://www.relishapp.com/rspec/>
- [10] **RailsCasts**. *Ruby on Rails Screencasts* [online]. <http://railscasts.com/>
- [11] **Matthew Willse**. “One Less Password”. *Mozilla Learning* [blog]. <https://blog.webmaker.org/one-less-password>
- [12] **Matthew Willse**. “One Less Password”. *Notebook for posterity* [blog]. <http://notebook.ideapublic.org/2014/one-less-password/>

## Anexo A. Modelo de datos

Ruby on Rails dispone de un ORM (*Object-relational mapping*) llamado *ActiveRecord* que nos permite trabajar con objetos en vez de hacerlo directamente con la base de datos sin tener que preocuparnos por crear tablas ni relacionarlas entre si. Esto nos permite centrarnos únicamente en la programación de nuestra aplicación, encargándose Rails de la migración de éstos objetos a su representación en tablas en el modelo relacional. El modelo de datos utilizado para el *Buzón Ciudadano* es el siguiente.

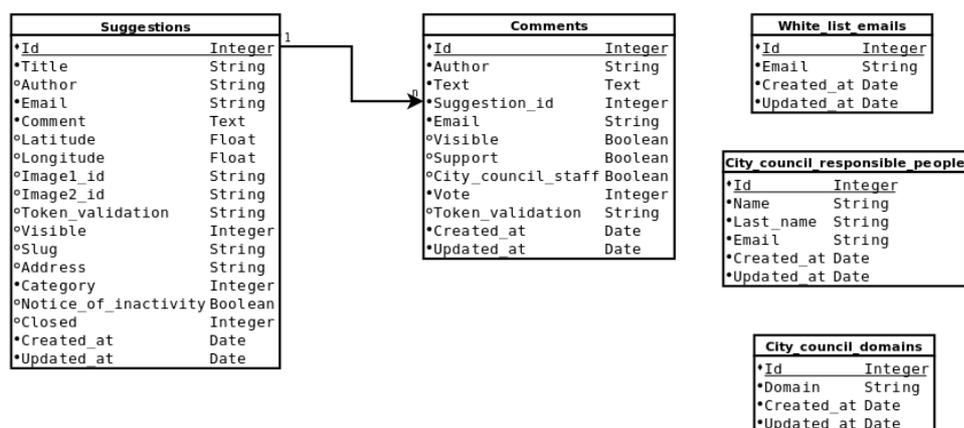


Figura 48: Modelo de datos

- *Suggestions:* Se utiliza para almacenar las sugerencias publicadas por los usuarios
- *Comments:* Se utiliza para almacenar los comentarios que se publican en las sugerencias
- *White\_list\_emails:* Se utiliza para almacenar las direcciones de correo electrónico que han sido validadas por los usuarios.
- *City\_council\_responsible\_people:* Se utiliza para almacenar los datos de contacto de los responsables del *Buzón ciudadano*, que son las personas a quienes se avisará cuando una sugerencia se cierre o un usuario reporte una sugerencia o un comentario.
- *City\_council\_domains:* Se utiliza para almacenar el dominio de correo electrónico del personal de la empresa, institución u organización que adopta el *Buzón Ciudadano*. A partir de estos dominios la aplicación es capaz de diferenciar a los usuarios normales del personal.

Las dos últimas tablas comienzan por *city\_council* (ayuntamiento en inglés) debido a que la primera versión del *Buzón Ciudadano* fue pensada exclusivamente para ayuntamientos.

## Anexo B. Estructura del proyecto

Una de las ventajas de utilizar un framework al desarrollar una aplicación es que el sistema de ficheros y directorios suele ser común para todas las aplicaciones desarrolladas con él. Esto permite que cualquier programador acostumbrado a trabajar con un framework determinado pueda empezar a trabajar en otro proyecto distinto y saber donde se encuentra cada cosa al estar familiarizado con la estructura del mismo.

Al crear una aplicación en Rails el generador del framework se encarga de crear por nosotros toda la estructura básica de directorios de la nueva aplicación y de poblarla con los ficheros correspondientes e incluso introducir código con la funcionalidad básica para interactuar con la aplicación. A continuación se describe la estructura básica de cualquier aplicación Rails.

<b>app/</b>	Contiene el código fuente del núcleo de la aplicación
<b>app/assets</b>	Ficheros JavaScript, CSS y/o imágenes de la aplicación
<b>app/controllers</b>	Controladores de la aplicación
<b>app/helpers</b>	Ficheros con métodos globales accesibles desde cualquier parte de la aplicación
<b>app/mailers</b>	Contiene los controladores de los mailers (envío de correos electrónicos)
<b>app/models</b>	Modelos de la aplicación utilizados para la persistencia de los datos
<b>app/presenters</b>	Ficheros con código fuente para evitar en la medida de lo posible el código ruby embebido en las vistas
<b>app/services</b>	Clases Ruby con la lógica de negocio de la aplicación
<b>app/views</b>	Ficheros HTML con las vistas de la aplicación y de los mailers
<b>bin/</b>	Contiene los binarios ejecutables
<b>config/</b>	Configuración de la aplicación
<b>db/</b>	Ficheros de la base de datos
<b>doc/</b>	Documentación de la aplicación
<b>lib/</b>	Módulos y librerías
<b>lib/assets</b>	Ficheros JavaScript, CSS y/o imágenes requeridos por las librerías
<b>log/</b>	Ficheros de log de la aplicación
<b>public/</b>	Ficheros de acceso público (a través del navegador), como páginas de error
<b>bin/rails</b>	Script para generar código, arrancar el servidor local...
<b>test/</b>	Tests de la aplicación (obsoleto si se usa spec)
<b>spec/</b>	Tests de la aplicación utilizando RSpec

<b>tmp/</b>	Ficheros temporales
<b>vendor/</b>	Código y <i>plugins</i> (gemas) de terceras partes
<b>vendor/assets</b>	Ficheros JavaScript, CSS y/o imágenes requeridos por los <i>plugins</i> de terceros
<b>README.rdoc</b>	Fichero para describir la aplicación
<b>Rakefile</b>	Herramienta de Rails disponible a través del comando <i>rake</i>
<b>Gemfile</b>	Librerías (o gemas) requeridas por la aplicación
<b>Gemfile.lock</b>	Lista de gemas usadas por la aplicación para asegurarnos de que todas las versiones de la misma usan las mismas versiones de gemas
<b>config.ru</b>	Fichero de configuración del <i>rack middleware</i>
<b>.gitignore</b>	Contiene los patrones de los ficheros que serán ignorados por Git cuando se hace <i>push</i> al repositorio



## Anexo C. Buzón Ciudadano en producción

Como se ha comentado en capítulos anteriores, para el despliegue del *Buzón Ciudadano* en producción se han utilizado los servicios de Heroku. La URL para acceder a la aplicación online es la siguiente: <http://tfg-suggestion-box.herokuapp.com/>.

El echo de utilizar un hosting gratuito para alojar nuestra aplicación conlleva una serie de inconvenientes que pueden dificultar su uso. Uno de estos inconvenientes es que la aplicación entra en un estado de hibernación para ahorrar recursos cuando no recibe peticiones y permanece inactiva durante más de una hora. Esto se debe a que se está utilizando un plan gratuito para aplicaciones con poco tráfico. Si la aplicación se encuentra en estado de hibernación puede que tarde unos segundos en ser mostrada en nuestro navegador debido a que los servicios de Heroku tienen que “levantarla”, pero una vez esté disponible ya funcionará con normalidad hasta que vuelva a entrar en un período de inactividad de 60 minutos y se suspenda de nuevo.

Otro de los inconvenientes de utilizar un servicio gratuito para alojar nuestra aplicación es que el envío de mensajes de correo electrónico puede demorarse en llegar a nuestra cuenta de correo hasta en más 20 minutos. Como el uso del *Buzón Ciudadano* puede llegar a ser un poco lento en determinados casos debido a esta limitación, se han dado de alta varias direcciones de correo electrónico en el sistema para poder publicar sugerencias y comentarios sin tener que esperar a que nos llegue el mensaje de validación a nuestra cuenta. Para poder probar las funcionalidades de editar o eliminar sugerencias y comentarios sí que deberemos utilizar nuestra dirección de correo electrónico real para publicarlas y que nos lleguen los mensajes de validación y confirmación.

Como no disponemos de cuentas de la universidad para hacernos pasar por personal de la UPV en el caso concreto del *Buzón UPV*, hemos utilizado el servicio gratuito de Mailinator (<https://mailinator.com/>) para disponer de cuentas de acceso público ilimitadas. Utilizar Mailinator es muy sencillo, simplemente accedemos a la URL arriba indicada y en el apartado *Check An Inbox!* introducimos la dirección de correo electrónico a la que queramos acceder. El acceso a este tipo de cuentas es libre, por lo que puede ser que la dirección introducida contenga mensajes de correo electrónico de otras personas.

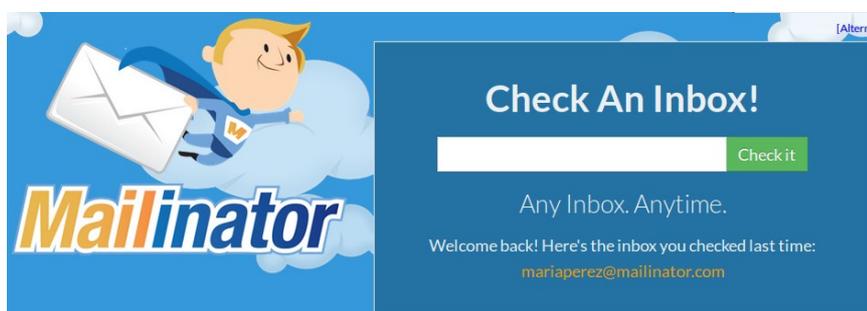


Figura 49: Mailinator

En la siguiente tabla tenemos las direcciones que han sido previamente dadas de alta en el sistema para poder probar su funcionamiento.

<b>Usuario validado</b>	enrique_carrillo_7@msn.com
<b>Responsable del buzón</b>	mariaperez@mailinator.com
<b>Personal UPV</b>	xxx@mailinator.com <i>Siendo xxx cualquier nombre que queramos utilizar</i>

*Tabla 3: Direcciones de correo electrónico validadas*

Para evitar estos problemas en el Anexo D se explica cómo configurar un entorno local para poder ejecutar la aplicación y poder probar su funcionamiento de una forma fluida sin los inconvenientes del hosting gratuito de Heroku.



# Anexo D. Configuración y puesta en marcha

En este anexo se va a explicar paso a paso cómo configurar el entorno de desarrollo local, descargar el código fuente del *Buzón Ciudadano* alojado en GitHub y desplegarlo en un servidor local para ejecutar el *Buzón Ciudadano* en nuestra propia máquina.

## D.1 Consideraciones iniciales

El *Buzón Ciudadano* ha sido desarrollado íntegramente en el sistema operativo Ubuntu, por lo que esta guía de instalación y puesta en marcha está enfocada a sistemas operativos Linux. Esta guía puede utilizarse para para el sistema operativo Windows, pero algunos pasos podría diferir ligeramente.

## D.2 Preparando el entorno de desarrollo

### D.2.1 Instalar RVM y Ruby

El primero paso para poder ejecutar una aplicación Ruby on Rails en nuestro sistema es instalar el lenguaje de programación Ruby. Para ello vamos a instalar una aplicación llamada RVM (*Ruby Version Manager*) que nos facilitará la instalación y gestión de todas las versiones de Ruby que tengamos instaladas en nuestro sistema.

Para instalar RVM es necesario que previamente tengamos instaladas algunas herramientas en nuestro sistema. Para ello tenemos que ejecutar el siguiente comando desde el terminal.

```
sudo apt-get install build-essential git-core curl
```

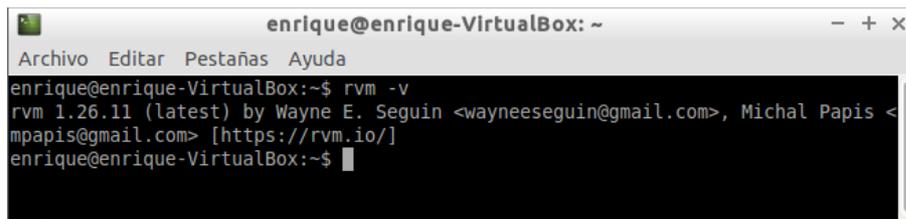
Una vez tengamos instaladas las herramientas requeridas vamos a proceder a instalar RVM ejecutando los siguientes comandos.

```
curl -sSL https://rvm.io/mpapis.asc | gpg --import  
curl -L https://get.rvm.io | bash -s stable
```

Para comprobar si se ha instalado el RVM correctamente debemos cerrar y volver a abrir el terminal y ejecutar el siguiente comando.

```
rvm -version
```

Si todo ha ido bien deberíamos de poder ver la versión instalada en el sistema.



```
enrique@enrique-VirtualBox: ~  
Archivo Editar Pestañas Ayuda  
enrique@enrique-VirtualBox:~$ rvm -v  
rvm 1.26.11 (latest) by Wayne E. Seguin <wayneeseguin@gmail.com>, Michal Papis <mpapis@gmail.com> [https://rvm.io/]  
enrique@enrique-VirtualBox:~$
```

Figura 50: Versión RVM instalado

Una vez instalado el RVM vamos a proceder a instalar Ruby en nuestro sistema. Podemos instalar cualquier versión de Ruby superior a la 1.9.3 (se recomienda la 2.2). En mi caso tengo instalada la versión 2.2.1p85 y es la que recomiendo para evitar posibles incompatibilidades.

```
rvm get head && rvm reload  
rvm install 2.2.1
```

La instalación de Ruby puede llevar un rato debido a que el asistente descargará también sus dependencias. Para comprobar que todo ha ido bien podemos ejecutar el comando `ruby -v` y comprobar la versión instalada

Después de instalar Ruby hay que definir un nuevo gemset, que es donde se instalarán las gemas que utiliza el *Buzón Ciudadano*, incluida la gema de Ruby on Rails.

```
rvm use 2.2.1@buzonCiudadano --create --default
```

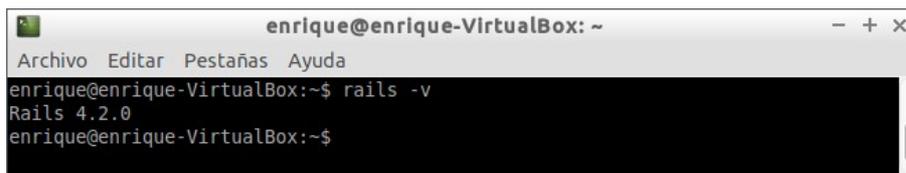
Este comando crea (`--create`) el gemset **buzonCiudadano** asociado a la versión 2.2.1 de Ruby y además hace que sea el gemset utilizado por defecto (`--default`).

## D.2.2 Instalar Ruby on Rails 4.2

Una vez instalado Ruby y creado el gemset para nuestra aplicación, es el momento de instalar Ruby on Rails 4.2 para poder ejecutar el *Buzón Ciudadano*.

```
gem install rails -v 4.2
```

El proceso de descarga e instalación de Ruby on Rails puede llevar un tiempo. Una vez finalice la instalación podremos comprobar si se ha instalado correctamente ejecutando el comando `rails -v`



```
enrique@enrique-VirtualBox: ~  
Archivo Editar Pestañas Ayuda  
enrique@enrique-VirtualBox:~$ rails -v  
Rails 4.2.0  
enrique@enrique-VirtualBox:~$
```

Figura 51: Versión de Rails instalada

A partir de este momento ya tendremos el entorno de desarrollo completamente configurado para poder crear o ejecutar aplicaciones Rails en local.



## D.3 Despliegue de la aplicación

### D.3.1 Descargar el Buzón Ciudadano

El código fuente y los ficheros de configuración para poner en marcha el *Buzón Ciudadano* se encuentran en un repositorio público alojado en GitHub ([https://github.com/ecarrillo88/Suggestion\\_Box](https://github.com/ecarrillo88/Suggestion_Box)). También es posible descargar el proyecto entero comprimido en un archivo ZIP desde el siguiente enlace [https://github.com/ecarrillo88/Suggestion\\_Box/archive/master.zip](https://github.com/ecarrillo88/Suggestion_Box/archive/master.zip).

### D.3.2 Ejecución y puesta en marcha

Una vez descargado el proyecto hay que descargar e instalar las dependencias con software de terceros utilizado durante el desarrollo de la aplicación. Para ello nos dirigimos a la raíz del proyecto (directorio `Suggestion_Box`) y ejecutamos el siguiente comando:

```
bundle install
```

Este comando se encargará de descargar e instalar todas las librerías externas (conocidas como gemas) utilizadas por la aplicación y que son necesarias para el funcionamiento de la misma. Si durante la instalación da algún error relacionado con la gema de PostgreSQL (pg 0.18.1) seguramente que sea porque nuestro sistema no dispone de la librería `libpq-dev`. Para solucionarlo solamente tenemos que instalarla con el siguiente comando:

```
sudo apt-get install libpq-dev
```

Antes de poner en marcha la aplicación es necesario crear la base de datos y cargar algunos datos en ella para que la aplicación pueda funcionar. Para ello tenemos que ejecutar los siguientes comandos desde la raíz del proyecto:

```
rake db:schema:load
rake db:seed
```

El primer comando se encarga de crear la base de datos y todas las tablas de la aplicación y el segundo comando se encarga de poblar algunas de estas tablas con datos necesarios para que el sistema pueda funcionar, como por ejemplo las direcciones de correo electrónico de los responsables del buzón o el dominio de correo electrónico del personal de la UPV.

Una vez tengamos creada la base de datos ya podremos poner en marcha la aplicación y empezar a utilizarla. Para ejecutar la aplicación tendremos que arrancar el servidor web que incluye Rails por defecto con el siguiente comando:

```
rails server
```

Para ver la aplicación en funcionamiento abrimos cualquier navegador y accedemos a <http://localhost:3000/>. Si todo ha ido bien deberíamos de poder ver la página principal del *Buzón UPV*.

### D.3.3 Recibir emails en local

Para poder enviar y recibir mensajes de correo electrónico en nuestro entorno local se utiliza la gema MailCatcher (<http://mailcatcher.me/>). Esta gema nos permite levantar en nuestro entorno local un servidor de correo electrónico muy ligero que intercepta todos los mensajes de correo enviados desde dentro de nuestra red local y nos permite visualizarlos en su cliente como si de un servidor de correo real se tratase. Para ejecutarlo hay que ir hasta la raíz del proyecto del *Buzón Ciudadano* (directorio `Suggestion_Box`) y ejecutar el comando `mailcatcher`. Una vez ejecutado el comando anterior se lanzará un servidor de correo local. Accediendo a <http://localhost:1080/> tendremos acceso a la interfaz de usuario de MailCatcher dónde recibiremos los mensajes de correo electrónico enviados por el *Buzón Ciudadano*.

### D.3.4 Direcciones de correo electrónico

Las direcciones de correo electrónico del personal de la UPV y del responsable del buzón son exactamente las mismas que las de la tabla 3 del anexo C, con la única diferencia de que ya no es necesario acceder a la web de Mailinator para leer los mensajes, puesto que Mailcatcher los interceptará y los mostrará en <http://localhost:1080/>.

Ya no existe un usuario estándar previamente validado como en el caso de la versión en producción del *Buzón UPV*. Ahora podremos utilizar cualquier dirección de correo electrónico y validar nosotros mismos las acciones de publicar, editar y eliminar desde el servidor local de correo electrónico.

