



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

**PROTOTIPO DE
APLICACIÓN PARA LA
GESTIÓN DE RELACIONES
CON EL CLIENTE**

Alumno: Agustín Ruiz Linares

Tutor: José Ramón Balsas Almagro

Dpto.: Informática

Septiembre, 2014

Índice de contenido

1.	Introducción.....	1
1.1.	Propósito.....	1
1.2.	Objetivos.....	1
1.3.	Metodología	3
1.4.	Estructura del documento	3
2.	Análisis.....	4
2.1.	Breve historia del CRM y contexto actual	4
2.2.	Análisis preliminar.....	5
2.2.1.	Estado del arte	5
2.2.2.	Desarrollo de aplicaciones web en PHP	5
2.2.3.	Desarrollo en PHP basado en framework.....	6
2.2.4.	Arquitectura Modelo-Vista-Controlador.....	7
2.3.	Requisitos del sistema.....	8
2.3.1.	Requisitos funcionales.....	8
2.3.2.	Requisitos no funcionales.....	10
2.4.	Propuesta de solución	10
2.4.1.	Framework de programación: Codeigniter 2.2.0.....	10
2.4.2.	Framework HTML, CSS y JavaScript	11
2.5.	Planificación de tareas.....	12
2.5.1.	Seguimiento del plan de iteración.....	13
2.5.2.	Seguimiento por sprints	15
2.6.	Estudio de viabilidad	16
2.6.1.	Equipamiento informático	16
2.6.2.	Gastos inmateriales	16
2.6.3.	Licencias de software y servicios externos.....	17
2.6.4.	Gasto en personal	17

2.6.5.	Inversión total y recuperación de la inversión	17
2.7.	Casos de uso	20
2.7.1.	Casos de uso de operaciones CRUD	22
2.8.	Diagrama Entidad-Relación	26
3.	Diseño	27
3.1.	Diagrama de clases	27
3.2.	Diseño de la interfaz	29
3.2.1.	Responsive design	30
3.2.2.	Ejemplos reales de la interfaz.....	31
3.3.	Perfiles de usuario	33
4.	Implementación.....	34
4.1.	Arquitectura.....	34
4.2.	Detalles de la implementación	36
4.2.1.	Controladores	36
4.2.1.1.	Login y variables de sesión	38
4.2.2.	Modelos	38
4.2.2.1.	Tablas en la base de datos.....	39
4.2.2.2.	Clases PHP de los modelos	39
4.2.3.	Vistas.....	41
4.2.3.1.	Framework Bootstrap 3	42
4.2.3.1.1.	Componentes Bootstrap.....	42
4.2.3.1.2.	Modal.....	42
4.2.3.1.3.	Widget DateTimePicker	45
4.2.3.2.	Sistema de alertas	45
5.	Conclusiones.....	46
5.1.	Mejoras y trabajos futuros.....	47
6.	Bibliografía	49

1.	Interfaz	52
2.	Modelos.....	52
3.	Contactos	54
4.	Usuarios	56
5.	Campañas.....	58
6.	Actividades.....	60
7.	Alertas	64
8.	Permisos	66
9.	Soporte.....	67
1.	Acceso a la plataforma y visión general	72
2.	Gestión de Contactos.....	73
2.1.	Crear un nuevo contacto.....	74
2.2.	Seleccionar contacto.....	76
3.	Gestión de Campañas.....	77
3.1.	Crear campañas	78
3.2.	Seleccionar campaña	79
4.	Gestión de Tickets de Soporte	80
4.1.	Crear un nuevo ticket de soporte	81
4.2.	Seleccionar ticket de soporte	82
5.	Gestión de Actividades.....	83
5.1.	Crear actividades.....	83
5.2.	Seleccionar actividad	85
6.	Gestión de Alertas.....	86
6.1.	Crear alerta	87
6.2.	Seleccionar alerta	88
7.	Gestión de Usuarios.....	89
7.1.	Crear un nuevo usuario	89

7.2.	Seleccionar usuario	91
7.3.	Perfiles	92

1. INTRODUCCIÓN

En éste documento presenta la memoria del Trabajo Fin de Grado, TFG, realizado por Agustín Ruiz Linares para la titulación de Grado en Ingeniería Informática, tutorizado por el profesor D. José Ramón Balsas Almagro.

1.1. Propósito

El TFG que ha tomado estos últimos meses ha consistido en el desarrollo de un prototipo de aplicación de gestión de relaciones con clientes, visitas y marketing basado en tecnologías web.

En Teoría de Marketing éste concepto se conoce más como CRM por sus siglas en inglés *Client Relationship Management*. Como se indica en [\[1\]](#), se trata de un modelo de gestión de cualquier organización que esté orientado a sus clientes o usuarios.

Pero, además, el término CRM coincide con la denominación de las herramientas informáticas para implementar el modelo de marketing CRM.

Cualquier organización que esté centrada en el mercado (o a un sector de la población en concreto) la llevará irremediabilmente a procesos centrados en sus clientes. Si estas relaciones son gestionadas de una manera correcta hará que aumente la satisfacción global de dichos clientes y que éstos se fidelicen. Para conseguirlo, es necesario gestionar la información generada en todos los procesos comunicativos con los clientes, que en muchas ocasiones puede resultar ser demasiado densa: preguntas, reuniones, llamadas, citas, eventos relevantes..., lo que puede resultar bastante tedioso e, incluso, llegar a olvidar tareas importantes del proceso.

Es por todo ello que se necesita una metodología eficaz, como CRM.

1.2. Objetivos

Los objetivos que se marcaron para este TFG en la propuesta presentada a la Escuela Politécnica Superior de Jaén fueron los siguientes:

1. *Realizar un estudio del prototipo a realizar, recopilando las principales funcionalidades y requisitos de la aplicación y perfiles de usuario requeridos.*

Como se ha comentado anteriormente, un objetivo primordial es la realización de un software que pueda resultar realmente útil para cualquier PYME o autónomo y

que no resulte traumática la curva de aprendizaje ni la implantación de la herramienta.

Por ello se hace necesaria la definición desde las primeras fases del proyecto la definición de los requisitos, tanto funcionales como no funcionales, que deben alcanzarse con el mismo.

Del mismo modo, han de detallarse los distintos perfiles de usuario que tendrán acceso a la aplicación, teniendo en cuenta los distintos niveles de acceso en función de su perfil profesional: comerciales, técnicos, administrativos, gerencia.

2. Prototipado de la solución a desarrollar.

El propio TFG es la confección de un prototipo de herramienta CRM básica que sienta las bases de la misma. Para ello será necesario la confección de algún tipo de prototipos de diseño y arquitectura de la aplicación que muestre una o varias ventanas tipo que ayude a los usuarios a trabajar con el software en los que se muestre claramente la ubicación de cada uno de los elementos, ya sean prototipos visuales o funcionales. Las ventajas que ello aportará a un desarrollo definitivo de una herramienta pueden ser la depuración del desarrollo en fases tempranas del desarrollo, refinar los requerimientos de la aplicación y mejorar la interfaz de usuario con el objeto de mejorar la usabilidad.

3. Realizar una aplicación web multiusuario de gestión de relaciones con el cliente basada en algún framework PHP.

La aplicación prototipo se elige que sea basada en tecnologías web para asegurar que sea multiusuario y multiplataforma.

Como lenguaje principal de programación se elige PHP [\[2\]](#), un lenguaje de programación del lado del servidor para el desarrollo web de contenido dinámico.

Para ayudar en el proceso de desarrollo de la aplicación se recomienda el uso de algún tipo de framework PHP. Un framework o “marco de trabajo” es un esquema para el desarrollo y la implementación de una aplicación. Éstos pueden ayudarnos con módulos ya preparados que pueden ser usados simplemente haciendo uso de alguna API, asegurándonos que son herramientas con una mínima tasa de errores.

Por otro lado, muchos frameworks cuentan con una comunidad de desarrollo y uso de la que podemos encontrar documentación que poder consultar y conjuntos de librerías y utilidades adaptadas al framework en concreto que poder adaptar a nuestra aplicación.

1.3. Metodología

Para el desarrollo de la aplicación se utilizarán las siguientes metodologías:

- *Estudio de los requisitos del sistema a desarrollar.* Lo principal es definir y desarrollar los objetivos básicos que debe cumplir el proyecto, identificar los actores que intervienen en el sistema, los requisitos funcionales y no funcionales
- *Recopilación bibliográfica sobre la temática y técnicas relacionadas.* Para un correcto desarrollo, será necesario el estudio de las distintas herramientas y técnicas que pueden ayudarnos a conseguir el fin del proyecto.
- *Estimación temporal y costes de desarrollo.* Es deseable que desde un primer momento se sepa aproximadamente el tiempo que tomará el desarrollo de la aplicación, así como los costes que todo el proceso supondrá.
- *Diseño del sistema usando una metodología orientada a objetos utilizando el patrón de diseño Modelo-Vista-Controlador (MVC).* Se utilizarán las herramientas y técnicas necesarias para que la aplicación siga una metodología orientada a objetos y use el patrón de diseño MVC, principalmente, apoyándose en algún framework PHP.
- *Implementación del prototipo del sistema.* Por último, sólo queda desarrollar el software según el diseño realizado.

1.4. Estructura del documento

Éste documento está estructurado en distintos capítulos con subapartados:

En primer lugar analizaremos el problema a resolver, recopilaremos los requisitos del sistema y daremos una planificación estimada de desarrollo. Del mismo modo haremos un estudio de los costes de desarrollo.

A continuación abordaremos el diseño del sistema realizando los diagramas y esquemas que sean necesarios para que el equipo de desarrollo pueda consultar y que les sirva de guía en la programación.

En el siguiente capítulo trataremos los detalles de la implementación de acuerdo con los detalles de diseño estudiados anteriormente.

Por último, tras la finalización del proyecto tendremos una reflexión sobre las conclusiones a las que hayamos llegado y completaremos con aquellas ideas que no hayan podido desarrollarse en el prototipo CRM.

Para finalizar, cerraremos la memoria con diversos apéndices con datos relevantes para comprender datos complementarios sobre el trabajo realizado.

2. ANÁLISIS

Una de las atribuciones principales de un ingeniero de software es el análisis en el proceso de desarrollo de una herramienta informática. Y mucho más importante debe serlo si ésta herramienta se espera que sea de uso empresarial, ya que el buen funcionamiento de la misma puede suponer una mejora del proceso de negocio o producir pérdidas importantes si no aumenta el rendimiento del trabajo desempeñado o contiene fallos que ponen en riesgo los datos de la aplicación.

2.1. Breve historia del CRM y contexto actual

Según afirma David Ribera en [3], el término CRM surge en 1993, cuando Tom Siebel fundó *Siebel CRM Systems, Inc.*, en la que se centró en el desarrollo de software que incrementase la efectividad del contacto con los clientes de cara a mejorar la calidad de los servicios que ofrece. Siebel anteriormente fue director de la división de marketing de *Oracle Corporation*, la cual compró *Siebel CRM System, Inc.* en 2006.

Desde entonces han sido muchos los desarrolladores que han implementado sus propias soluciones CRM y muchas las empresas que han confiado en éstas. Versiones de pago y otras *Open Source* están disponibles en internet, como podemos observar en [4]. La tendencia que se está observando en los últimos años es que éstas soluciones están construidas sobre tecnología *Cloud Computing* y la filosofía *SaaS (Software as a Service)* con el objetivo de que sean accesibles desde cualquier lugar y cualquier dispositivo con conexión a internet.

Hoy en día podemos encontrar numerosas aplicaciones informáticas ya desarrolladas que cubren todos los aspectos requeridos en la implantación de una metodología CRM en la empresa como *SalesForce*, *ForceManager*, *Microsoft Dynamics* o *SugarCRM* ([4]); además de otras soluciones *Open Source* como *SugarCRM Community Edition* ([5]). Sin embargo, el gran problema de las herramientas que he tenido la oportunidad de probar es su lenta curva de aprendizaje que dificulta el proceso a cualquier pequeña o mediana empresa o autónomo que quiera poner en práctica metodologías CRM en su día a día sin tener que poseer conocimientos en marketing más allá de lo que le dicta su intuición y experiencia profesional.

2.2. Análisis preliminar

La necesidad de un CRM puede ser un factor subjetivo en el éxito de un negocio u organización en general. Sin embargo, es incuestionable que su utilización sólo tiene ventajas que añadir a los procesos empresariales. Si bien requiere cierta dedicación de tiempo a la herramienta, puede considerarse directamente este tiempo como el mismo que se invierte en la redacción de informes de actividad de cada empleado sobre las tareas que realiza, pudiendo ser incluso menor, y siempre más cómodo y dinámico.

Según [\[1\]](#), una buena metodología CRM debe cubrir todas las relaciones comerciales basadas en cuatro pilares fundamentales:

- Gestión eficaz de la información y las transacciones realizadas.
- Herramienta de comunicación y notificaciones.
- Gestión de proyectos.
- Proceso de implantación basado en análisis, metodologías y formación.

2.2.1. Estado del arte

En el estudio del estado del arte nos centraremos en la herramienta de licencia gratuita más extendida en descargas: SugarCRM Community Edition ([\[5\]](#)).

Si bien la herramienta es muy completa, para la gran mayoría de PYMES y autónomos puede resultar demasiado compleja, con una curva de aprendizaje demasiado lenta. La gran cantidad de opciones y pasos demasiado rígidos hacen que el uso de la aplicación se convierta en un fin en sí misma, en lugar de ser sólo una ayuda en el proceso de comunicarse con los clientes.

2.2.2. Desarrollo de aplicaciones web en PHP

En los objetivos del TFG se menciona expresamente que se realizará una aplicación basada en algún framework PHP. Pero, ¿por qué PHP? Hay otros muchos lenguajes para ese cometido, como pueden ser JSP de Java, .NET, Python, Ruby... De todos los mencionados, los más usados para el desarrollo web hoy en día son PHP y Java.

En primer lugar hay que tener en cuenta el contexto de desarrollo del proyecto. Se pretende implementar una herramienta que pueda ser fácilmente utilizada por cualquier PYME o autónomo cuyo personal tenga un nivel en las *Tecnologías de la Información y la Comunicación* medio, también denominado nivel TIC medio. La inmensa mayoría de las organizaciones que cumplen con esa premisa suelen

contratar servicios de hosting a terceros capaces de soportar aplicaciones web escritas en PHP y bases de datos MySQL. Una aplicación escrita en Java requeriría un VPS (*Virtual Private Server*) en la mayoría de los proveedores de hosting, lo que supondría un aumento en la inversión del hosting.

Por otro lado, la mayoría de las web de las organizaciones objetivo de la herramienta CRM que pretendemos desarrollar podrían compartir el mismo hosting, estando cada una instalada en un subdominio. Ésta situación, en todo caso, debería estudiarse teniendo en cuenta el tráfico de cada uno de los dominios y si el hosting es capaz de cubrir las necesidades de ambos.

Como grandes características del lenguaje PHP que hacen decantarse por éste lenguaje mencionamos las siguientes:

Simple: Permite generar código de una forma rápida, facilita la reutilización y tiene una curva de aprendizaje realmente rápida. Del mismo modo, el mantenimiento se hace más sencillo.

Rápido: No necesita demasiados recursos del sistema y tiene unos tiempos de respuesta muy aceptables.

Seguro: Bien utilizado, ofrece protección contra ataques.

Estable: Posee un sistema de administración de recursos y manejo de variables bastante fuerte y estable ante caídas.

2.2.3. Desarrollo en PHP basado en framework

Se decidió en la fase de la propuesta del TFG el apoyar el desarrollo del software en un framework PHP. Tal y como se adelantó anteriormente, un framework es un esquema para el desarrollo y la implementación de una aplicación. De entre los frameworks compatibles con PHP más famosos tenemos *Laravel*, *Symphony2* y *Codeigniter* ([6]).

La elección de un framework no es trivial. Hay que tener en cuenta distintos factores para decantarse por uno u otro. Los factores clave son los siguientes:

- Versión PHP en la que esté desarrollado, siendo preferible que sea PHP5.
- Respaldo de una comunidad activa y participativa.
- Documentación del framework, tanto de uso como de desarrollo del mismo.
- Facilidad en el desarrollo mediante la automatización de operaciones *CRUD* (*Create, Read, Update, Delete*).
- Facilidad en la instalación del framework en cualquier servidor.

- Flexibilidad.
- Simplicidad.
- Reutilización.
- Seguridad.
- Tiempos de respuesta del servidor.
- Soporte para MVC, patrón *Modelo-Vista-Controlador*.
- Abstracción del modelo de datos concreto utilizado, si bien también debe poseer soporte para confección de consultas personalizadas sobre el modelo de datos, por ejemplo, en caso de necesitar una consulta compleja que los métodos de acceso del framework no permitan obtener.
- Integrable con bibliotecas de terceros.

Por último, pero no menos importante, hay que tener en cuenta la estructura de la interfaz para lo cual podríamos apoyarnos nuevamente en un framework, ésta vez orientado a los lenguajes HTML, CSS y JavaScript. De éste estilo de frameworks los hay más completos, como *Bootstrap*, *Foundation 5* o *Skeleton* ([\[7\]](#)); o bien podemos programar en CSS sistemas de *grid* (o cuadrícula) para el maquetado de elementos en la página.

La elección de un buen framework para el *front-end* puede agilizar el proceso de maquetado, mantener el código limpio y ordenado y solucionar problemas de incompatibilidades entre navegadores.

2.2.4. Arquitectura Modelo-Vista-Controlador

Uno de los criterios de la metodología de desarrollo de la aplicación era el uso del patrón de arquitectura *Modelo-Vista-Controlador*, MVC. Éste tipo de arquitectura se usa para el desarrollo de sistemas interactivos, tal y como es el caso que se nos presenta.

La arquitectura MVC divide la aplicación en tres grupos de componentes, tal y como se indica en [\[8\]](#):

Modelos: comprenden la funcionalidad principal. Controla la representación lógica de la información y gestiona los accesos a dicha información, esto es, las operaciones *CRUD*. También encapsula las operaciones que se realizan sobre los datos y que están directamente relacionadas con las tareas de los procesos de negocio, esto es, la lógica de negocio.

Vistas: presentan a través de una interfaz al usuario el modelo, recogiendo los cambios que en ellos se realicen y mostrándolos en los controladores para su procesamiento.

Controladores: gestionan las actualizaciones de las vistas y responde a los eventos del sistema accediendo a los métodos de los modelos cuando es necesario. Resumiendo, hace de intermediario entre el modelo y la vista.

2.3. Requisitos del sistema

Pasemos a definir los requisitos del prototipo que se ha diseñado y desarrollado. Éstos se pueden diferenciar en requisitos funcionales o requisitos no funcionales.

Estos requisitos han sido obtenidos principalmente por la experiencia académica y laboral del estudiante, donde intenta expresar las situaciones y problemática más común que puede darse en una PYME u organización estándar o los casos que pueden plantearse a un autónomo que necesite mantener un registro de las comunicaciones con sus clientes.

2.3.1. Requisitos funcionales

Se define como *requisito funcional* aquella característica requerida del sistema que expresa una capacidad de acción del mismo, esto es, una funcionalidad del sistema o de sus componentes.

Para este proyecto se definen los siguientes requisitos funcionales:

1. El sistema debe permitir la gestión de los contactos o clientes de una empresa y poder realizar con ellos las siguientes operaciones:
 - 1.1. Operaciones CRUD.
 - 1.2. Clasificación de contactos en función de su estado: no especificado, cliente potencial, cliente activo o cliente agotado.
 - 1.3. Asignar contactos a un usuario del sistema.
 - 1.4. Controlar las actividades llevadas a cabo con el contacto.
2. El sistema debe permitir la gestión de usuarios de la aplicación y poder realizar con ellos las siguientes operaciones:
 - 2.1. Operaciones CRUD.
 - 2.2. Clasificación de usuarios en función de su rol: comerciales, técnicos, administrativos, gerencia.
 - 2.3. Limitar acceso a los usuarios en función de su rol.

3. El sistema debe permitir la gestión de campañas (ventas, visitas, marketing,...) y poder realizar con ellas las siguientes operaciones:
 - 3.1. Operaciones CRUD.
 - 3.2. Asignar campaña al usuario responsable de la campaña.
 - 3.3. Controlar las actividades llevadas a cabo con cada contacto para la campaña.
4. El sistema debe permitir la gestión de actividades y poder realizar con ellas las siguientes operaciones:
 - 4.1. Operaciones CRUD.
 - 4.2. Clasificación de actividades en función de su tipo: llamada telefónica, correo electrónico, carta, fax, tarea o reunión.
 - 4.3. Clasificación de actividades en función de su prioridad: alta, media o baja.
 - 4.4. Clasificación de actividades en función de su estado: no iniciada, en progreso, completada, pendiente de información, aplazada o cancelada.
 - 4.5. Asociar actividad con un contacto.
 - 4.6. Asociar actividad con una campaña (opcional).
5. El sistema debe permitir la gestión de alertas y notificaciones y poder realizar con ellas las siguientes operaciones:
 - 5.1. Operaciones CRUD.
 - 5.2. Programar alertas.
 - 5.2.1. Asociar un usuario.
 - 5.2.2. Asociar un contacto.
 - 5.2.3. Asociar una actividad (opcional).
 - 5.2.4. Ventana emergente.
 - 5.2.5. Envío por correo electrónico.
6. El sistema debe permitir la gestión de tickets de soporte técnico e incidencias y poder realizar con ellas las siguientes operaciones:
 - 6.1. Operaciones CRUD.
 - 6.2. Comunicación a través de una plataforma Web.

Los requisitos serán descritos con más detalle en la [planificación de tareas](#) cuando se redacten las *historias de usuario*.

2.3.2. Requisitos no funcionales

Se define como *requisito no funcional* aquella característica requerida del sistema, del proceso de desarrollo, del servicio prestado o de cualquier otro aspecto del desarrollo, que señala una restricción del mismo.

Para este proyecto se estiman necesarios los siguientes requisitos no funcionales:

- La aplicación debe ser accesible desde cualquier dispositivo con un navegador web compatible y por distintos usuarios simultáneamente.
- La interfaz debe realizarse siguiendo una filosofía *responsive design*, de tal manera que no importe el dispositivo desde el que se acceda: ordenador personal o portátil, tablet, smartphone o cualquier dispositivo que posea un navegador web compatible con conexión a internet.
- La interfaz debe procurar maximizar la intuición en su uso, consistencia de elementos gráficos y ser amigable con el usuario.
- Debe tener suficiente flexibilidad en el desarrollo como para ampliar funcionalidades, esto es, que se garantice su escalabilidad y ampliabilidad.
- Debe ser fácil de instalar en cualquier equipo local o hosting de internet que cumpla con los requisitos mínimos que se establezcan.
- La aplicación debe tener mecanismos de seguridad suficientes para evitar accesos fraudulentos.
- Allá donde sea posible se aplicará la validación de datos: campos obligatorios, tipos de datos (p.ej.: fecha y hora), contraseñas...

2.4. Propuesta de solución

Con todo lo observado y estudiado en los puntos anteriores se decide que la solución sea una aplicación web desarrollada en PHP.

2.4.1. Framework de programación: Codeigniter 2.2.0

La elección del Framework de desarrollo de aplicaciones PHP se decanta por Codeigniter en su versión 2.2.0 en el momento de iniciar el desarrollo ([9]).

Los puntos a favor de este framework sobre el resto de framework que se consideraron fueron los siguientes:

- **Gran rendimiento.** Las pruebas realizadas muestran tiempos de respuesta del servidor que, en el peor de los casos, toma tan solo un par de décimas de

segundo. Por otro lado, Codeigniter incluye un sistema de almacenamiento en memoria caché de resultados que reduce los tiempos de respuesta del servidor a centésimas.

- **Ligero.** No sobrecarga el servidor con grandes funciones.
- **Adaptado para PHP5.** El uso correcto de la Programación Orientada a Objetos con PHP5 ofrece una aplicación más estable y rápida.
- **Pequeña curva de aprendizaje.** En muy poco tiempo puede aprenderse los conceptos básicos de manejo de controladores, uso de modelos y llamada a vistas; así como el uso de bibliotecas para ampliar funcionalidades del sistema.
- **Patrón MVC.** El framework es compatible con el patrón arquitectónico Modelo-Vista-Controlador, que es requisito de la propuesta del TFG.
- **Flexible.** Es menos rígido que la inmensa mayoría del resto de frameworks. Si bien Codeigniter define una manera específica para trabajar es decisión del programador el seguirlas o no.
- **Facilidad de instalación.** Basta con tener acceso FTP para subir Codeigniter en el servidor y su configuración se realiza en ficheros fácilmente identificables: configuración general, configuración de base de datos, paginación, sesiones... Es relativamente sencillo programar una pequeña aplicación que pregunte al usuario éstos parámetros y los inserte automáticamente en los ficheros adecuados.
- **Bien documentado.** La documentación oficial es fácil de estudiar y entender al estar expresada en forma de tutorial. También la comunidad de desarrollo es muy amplia y existe gran cantidad de dudas ya resueltas en foros de internet fácilmente accesibles desde cualquier buscador y realizando la consulta adecuada.
- **URL amigables.** El framework puede generar URL amigables para mejorar el posicionamiento en motores de búsqueda. Aunque para el objetivo de desde TFG no es relevante podría sí serlo para futuros proyectos, para los cuales la fase de aprendizaje del framework ya estaría superada.

2.4.2. Framework HTML, CSS y JavaScript

Elegimos *Bootstrap 3* como el Framework para la maquetación y diseño del sitio web ([10]). Los motivos que llevan a elegir éste y no otro son los siguientes:

- **Completo.** Incluye estilos CSS para casi cualquier etiqueta HTML. El maquetado se hace sencillo mediante *grid*.

- **Agradable a la vista:** Usa colores suaves pero de alto contraste
- **Fácil de usar y bien documentado:** La documentación oficial explica todas las posibilidades del framework mediante ejemplos que ayudan a iniciarse en la herramienta e, incluso, a hacer modificaciones adiciones al código.
- **Gran comunidad:** En internet hay multitud de comunidades que usan Bootstrap y que desarrollan sus propias bibliotecas para extender funcionalidades sin que rompan con el estilo general de *Bootstrap*.
- **Optimizado:** Los estilos están verificados con los navegadores web más usados, así como los componentes y scripts que pueden usarse. También incluye las reglas CSS necesarias para hacer que el sitio cumpla con la filosofía responsive design y se adapte a cualquier tamaño de dispositivo.
- **Personalizable:** La página de descarga permite personalizar colores, tipografías, iconos, componentes y elementos HTML que personalizará el framework.

2.5. Planificación de tareas

La planificación y la estimación son dos actividades críticas para el éxito de cualquier proyecto de desarrollo software.

El presente TFG está estimado en 12 créditos ECTS. Cada crédito ECTS está estimado entre 25 y 30 horas de esfuerzo del estudiante, por lo que la duración de este TFG en su totalidad debería tomar entre 300 y 360 horas en total.

Desde el primer momento se decide que el proyecto será desarrollado usando una metodología ágil: **SCRUM** ([11]).

En primer lugar se define el **Product Backlog** del proyecto. En él se definen los requisitos del proyecto a través de *historias de usuario*. Estas historias son también valoradas en complejidad mediante *puntos de historia*; y ordenadas según el valor de negocio que éstas tienen.

Las historias de usuario han sido agrupadas en distintos *Temas* para identificar a qué aspecto de la aplicación se identifica cada uno. En la tabla 1 se detallan las historias de usuario y los puntos de historia que suman todas ellas:

Tema	Nº historias de usuario	Nº puntos de historia
Interfaz	2	15
Modelos	8	20
Contactos	8	37
Usuarios	11	44
Campañas	11	39
Actividades	22	55
Alertas	12	49
Perfiles	5	18
Soporte	14	40
Total	93	331

Tabla 1: Resumen del Product Backlog

El detalle de las historias de usuario, así como los puntos de historia asignados a cada una, se puede observar en el *Anexo I: Product Backlog*.

2.5.1. Seguimiento del plan de iteración

Según la experiencia del equipo de desarrollo sabemos que se tiene una velocidad de 50 puntos de historia por sprint, con una holgura de ± 3 puntos de historia. Cada sprint considerado es de 10 días de trabajo (2 semanas naturales). Con estos datos podemos determinar que el proyecto será desarrollado en 7 sprints:

$$\frac{317 \text{ puntos de historia}}{50 \text{ puntos de historia/sprint}} = 6,34 \text{ sprints} = 7 \text{ sprints} \quad (1)$$

La distribución de historias de usuario en sprints se denomina *Sprint Backlog*. El resumen de historias de usuario se puede observar en la siguiente tabla:

Sprint	Historias de usuario (nº total)	Puntos de historia
Sprint 1	INT1, INT2, MOD1, CNT1, CNT2, CNT3, CNT4, CNT5, CNT6, MOD2 (10)	50
Sprint 2	USU1, USU2, USU3, USU4, USU5, USU7, USU8, MOD3, CAM1, CAM2 (10)	49 (-1)
Sprint 3	CAM4, CAM5, CAM6, CAM7, CAM8, MOD4, ACT2, ACT4, ACT5, ACT7, ACT9, ACT11, ACT12, ACT13, ACT14, ACT15, ACT16 (17)	50
Sprint 4	MOD5, ALE1, ALE2, ALE3, ALE4, ALE5, ALE6, ALE7, ALE8, ALE9, ALE10, ALE11 (12)	51 (+1)
Sprint 5	MOD6, PER1, PER2, PER3, USU6, CAM3, ACT1, ACT3, ACT6, ACT8, ACT10, ACT20, ACT21, ALE12, CAM11, ACT22, MOD 8 (17)	49 (-1)
Sprint 6	SOP1, SOP3, SOP5, ACT17, SOP7, SOP9, SOP2, SOP4, SOP6, SOP8, SOP10, SOP13, SOP14, MOD7, CNT7, CNT8 (16)	49 (-1)
Sprint 7	USU10, USU11, CAM9, CAM10, ACT18, ACT19, PER4, PER5, SOP11, SOP12, USU9 (11)	33 (-17)
Total	93	331

Tabla 2: Resumen del Sprint Backlog

Vemos como en la mayoría de los casos que no podía cuadrarse la planificación a la velocidad del equipo se ha procurado que la desviación sea negativa. Esto es para no pecar de asignar más trabajo del que se puede desarrollar.

Sólo en un caso se ha contemplado un sprint de más de la velocidad (el sprint 4, que supera la velocidad en 1 punto de historia), para no dejar un sprint muy por debajo de la velocidad estimada del equipo.

Con estos datos se ha confeccionado una gráfica de progreso estimado del proyecto denominada *Burndown Chart*. Su estado inicial es el siguiente:

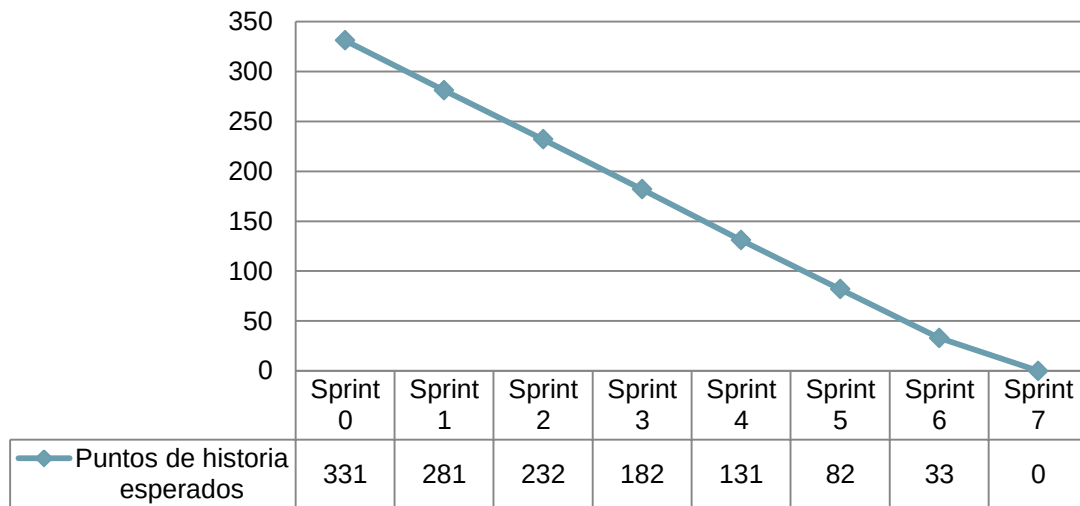


Ilustración 1: Burndown Chart inicial

En la gráfica se controlará el progreso del proyecto contra el plan para reajustar la planificación en caso de que no se cumpla y, en cualquier caso, siempre al finalizar el sprint y nunca durante su ejecución.

2.5.2. Seguimiento por sprints

Cada sprint se planifica independientemente como si de un proyecto con una metodología clásica se tratase. El método más recomendado para hacer el seguimiento es una *Tabla de Tareas*, *Task Board* o *Tabla Kanban*.

<i>Sprint backlog</i>	<i>Por hacer</i>	<i>En progreso</i>	<i>Por verificar</i>	<i>Aceptado</i>

Ilustración 2: Plantilla de Task Board

Se trata de una manera muy visual de representar el trabajo de un sprint. En la tabla se van colocando las fichas de las historias de usuario. Cada miembro del equipo de desarrollo puede tener su propia fila y es el responsable de ir ordenando sus historias de usuario; o bien, puede tenerse un backlog único para todo el equipo.

2.6. Estudio de viabilidad

En éste punto se analizará los costes de desarrollo de la herramienta. Para ello ha de tenerse en cuenta que el equipo de desarrollo está compuesto por un solo miembro y que el tiempo límite de entrega se ha estimado en 3,5 meses (7 sprints).

2.6.1. Equipamiento informático

Se decide que las compras del equipamiento informático se realizará en la tienda online *Pc Componentes*¹.

Portátil Acer Aspire E5-571G. Intel Core i7-4510U RAM 8GB Almacenamiento 1TB Pantalla 15,6" Controlador Gráfico nVidia GeForce GT840M	629,00€
Monitor Asus VS229HA 22" LED Conexiones: VGA, DVI, HDMI, salida audio Jack 3,5mm	109,00€
Altavoces Wortex Big Bass 82	12,95€
Kit teclado y ratón Asus U2000	14,75€
Alfombrilla Genesis M11	13,95€
Gastos de envío	7,75€
Total	787,40€

2.6.2. Gastos inmateriales

Acceso a internet ONO 50Mbps (Internet Fibra). Coste: 41,14€/mes (IVA incluido). Precio: 3,5meses × 41,14€/mes	143,99€
---	----------------

¹ Web: <http://www.pccomponentes.com/>. Precios recuperados en agosto de 2014.

2.6.3. Licencias de software y servicios externos

Sistema Operativo elementaryOS Luna ²	0,00€
Servidor Apache2 + PHP5 + MySQL (instalación y desarrollo local)	0,00€
Repositorio GitHub para control de versiones: Plan Micro para 5 repositorios privados: \$7'00/mes = 5,27€/mes ³ . Precio: 3,5meses × 5,27€/mes	18,45€
Herramienta de prototipado basado en wireframe <i>Moqups</i> ⁴	0,00€
Herramienta de creación de diagramas <i>Dia</i> ⁵	0,00€
Licencia Visual Paradigm: Standard Edition: \$15'00/mes = 11,29€/mes. Precio 3,5meses×11,29€	39,52€
Framework Codeigniter 2.2.0	0,00€
Biblioteca DataMapper (ORM para Codeigniter)	0,00€
Framework Bootstrap 3	0,00€
Editor de texto Sublime Text 3 ⁶	0,00€
Suite ofimática LibreOffice 4.3.0 ⁷	0,00€
Editor de imágenes vectoriales <i>Inkscape</i> ⁸	0,00€
Total	57,97€

2.6.4. Gasto en personal

Tomamos como referencia el salario de un Programador Sénior según el vigente Convenio colectivo de Telefónica Telecomunicaciones Públicas, SA; el cual puede consultarse en [\[12\]](#).

Programador sénior: 2.166,79€/mes. 3,5meses * 2.166,79€/mes	7.583,77€
---	------------------

2.6.5. Inversión total y recuperación de la inversión

Con todo lo anterior tenemos éstos costes de desarrollo de la aplicación:

$$787,40€ + 143,99 + 57,97€ + 7.583,77€ = 8.573,13€ \quad (2)$$

² <http://elementaryos.org/>

³ Cantidad de <http://www.conversormonedas.com/>. Calculado el 26 de agosto de 2014.

⁴ <https://moqups.com/>

⁵ <http://dia-installer.de/>

⁶ <http://www.sublimetext.com/>

⁷ <https://es.libreoffice.org/>

⁸ <http://www.inkscape.org/es/>

A partir de este punto es hora de pensar en qué modelo de negocio se puede tomar para recuperar la inversión realizada y/o generar beneficios con la herramienta desarrollada.

En primer lugar podríamos pensar en una **microfinanciación masiva** o **crowdfunding**. El desarrollo podría sostenerse a través de pequeñas aportaciones de personas de todo el mundo que estén interesadas en su lanzamiento. Posteriormente, podría decidirse la liberación del código y convertirlo en **Software Libre** y mantenerlo por la comunidad.

Sin embargo, la opción que puede resultar más interesante a largo plazo, pero también más arriesgada, es la confección de un **modelo de negocio** para la comercialización del software.

Aunque no es el objeto de este TFG, vamos a introducir los factores que hemos tenido en cuenta:

- **Fuente de inversión:** Puede ir desde la aportación de capital privado de los socios participantes en el negocio, préstamos bancarios o subvenciones gubernamentales de distintos orígenes.
- **Inversión inicial:** Aquí se tendrá en cuenta el acopio de material, equipamiento y mobiliario necesario para la actividad empresarial. Por supuesto, también se incluye la cantidad calculada anteriormente del coste de desarrollo del software.
- **Gastos mensuales del negocio:** En ellos tendremos que incluir salarios de los trabajadores, material fungible, mantenimiento de los bienes de la empresa y otros gastos de necesidad; como puede ser el alquiler de servidores dedicados a terceros. Además, si procede, deberemos tener en cuenta el pago del alquiler de las oficinas de la empresa y los pagos de las mensualidades del préstamo bancario si lo hubiese.
- **Ingresos de corto plazo:** Son aquellos trabajos y proyectos que surgen para el corto plazo y que suponen una inyección rápida de capital a la empresa. Son difíciles de predecir, pero pueden garantizar la subsistencia de la empresa en sus primeros meses.
- **Ingresos de largo plazo:** Es la inversión que se prevé que sea constante y continuada en el tiempo. En nuestro caso, la venta de licencias del software de gestión de relaciones con clientes que se ha desarrollado.

La subsistencia de una empresa se basa en el concepto de *ingeniería económica* llamado **punto de equilibrio**, esto es, conseguir que los ingresos menos los gastos den un balance de cero o, lo que es lo mismo, los ingresos cubran los gastos:

$$\text{Ingresos} - \text{Gastos} = 0 \Rightarrow \text{Ingresos} = \text{Gastos} \quad (3)$$

Como los ingresos a corto plazo son más difíciles de predecir y requieren un estudio más detallado los extraemos de la ecuación, por lo que sólo tendremos en cuenta los ingresos a largo plazo.

Los ingresos a largo plazo mensuales se pueden calcular, para nuestro modelo, con la siguiente fórmula:

$$\text{Ingresos mensuales} = n^{\circ} \text{ clientes} \times \text{Precio mensual licencia software} \quad (4)$$

Supongamos que tenemos unos gastos mensuales que ascienden a 5.000€. Por tanto, la fórmula del punto de equilibrio queda de la siguiente manera:

$$\begin{aligned} \text{Ingresos mensuales} &= \text{Gastos mensuales} = 5.000\text{€} \\ &= n^{\circ} \text{ clientes} \times \text{Precio mensual licencia software} \end{aligned} \quad (5)$$

Ya sólo nos queda fijar una variable para despejar la otra: podemos definir un precio mensual fijo para la licencia y calcular cuántos clientes debemos tener para subsistir; o bien, definir cuántos clientes queremos tener para despejar qué precio debe tener la licencia. En nuestro caso nos decantaremos por la primera opción y definiremos un importe mensual para la licencia de 24,95€. Por tanto, el número de clientes que debe alcanzar la empresa para llegar al punto de equilibrio es:

$$5.000\text{€} = n^{\circ} \text{ clientes} \times 9,95\text{€} \Rightarrow n^{\circ} \text{ clientes} = \frac{5.000\text{€}}{9,95\text{€}} \cong 503 \text{ clientes} \quad (6)$$

Con ésta cifra se debe trazar una estrategia de negocio que consiga no solo superar la cifra de equilibrio económico, sino que la supere para salvar la deuda de la inversión inicial.

En la siguiente gráfica se muestra el acumulado de ingresos y gastos generado en 4 años con una estrategia de aumentar cada mes en 10 los clientes de la empresa.

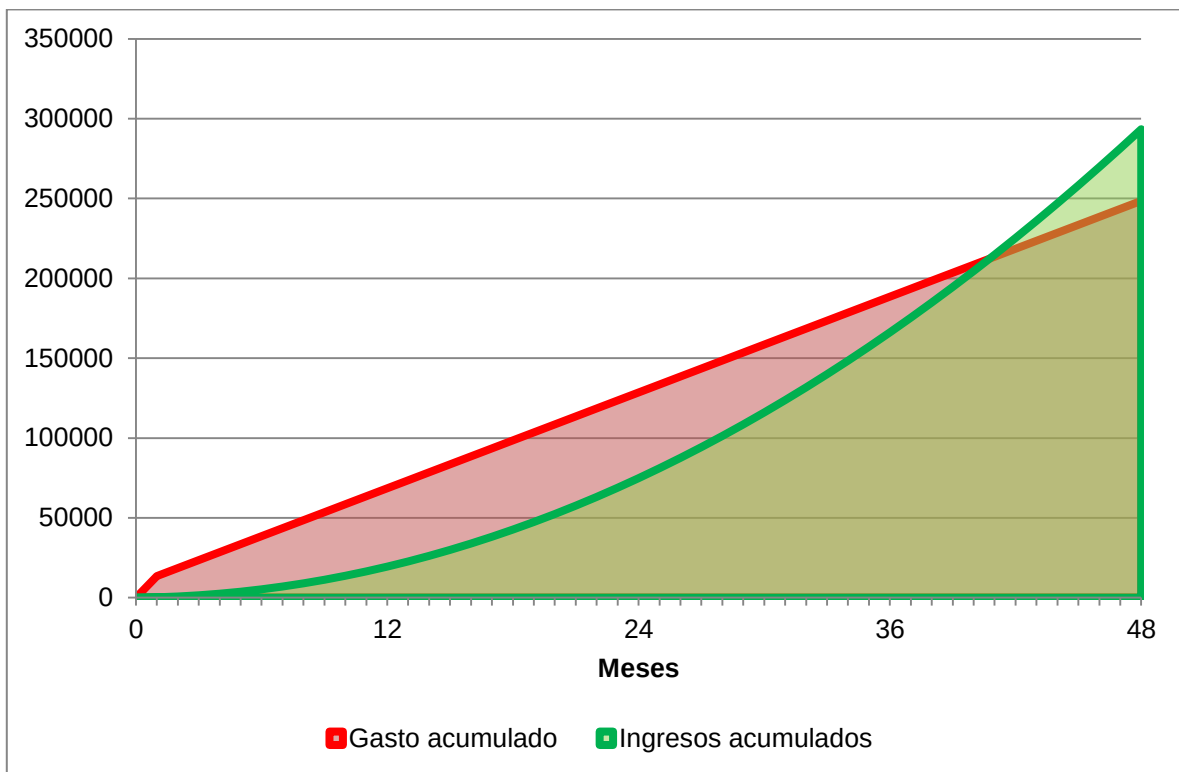


Ilustración 3: Gráfica de amortización

Se puede observar en la gráfica que el punto de equilibrio se consigue en el mes 21 (año 2), a partir del cual la empresa deja de dar pérdidas y empieza a saldar la deuda. Visualmente se identifica la zona en la que la pendiente de la curva de ingresos se iguala a la recta de gastos.

Será en el mes 41 (año 4) donde, siguiendo la progresión, se conseguirá un balance acumulado positivo y el negocio se consideraría un éxito empresarial.

2.7. Casos de uso

La utilización de casos de uso es frecuentemente utilizada para apoyar el análisis tanto en técnicas orientadas a objetos como en técnicas de análisis estructurado.

Según puede leerse en [8], la técnica de casos de uso es usada para describir bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista de un usuario. Permite definir los límites del sistema y las relaciones entre el sistema y el entorno mediante el uso del lenguaje natural.

Sin embargo, desde el principio hemos decidido en éste proyecto utilizar metodologías ágiles de desarrollo. Dean Leffingwell afirma en [13] que los casos de uso rara vez se usan en proyectos ágiles, aunque sí es muy recomendable su uso en proyectos que adquieren cierta envergadura.

De ahí surge el concepto de **Casos de Uso 2.0**. Las diferencias entre los casos de uso tradicionales y los casos de uso 2.0 son las siguientes:

“Caso de Uso: Un caso de uso expresa todas las formas de usar un sistema para alcanzar una meta particular para un usuario. En conjunto, los casos de uso le proporcionan todos los caminos útiles de usar el sistema e ilustran el valor que este provee.

Casos de Uso 2.0: Es una práctica escalable y ágil que emplea los casos de uso para capturar un conjunto de requisitos y conducir el desarrollo incremental de un sistema que los realice.

Los Casos de Uso 2.0 conducen el desarrollo de un sistema ayudándole, en principio, a entender cómo se empleará el sistema y luego ayudándole a evolucionar un sistema apropiado que apoye a los usuarios. Esta técnica se puede usar junto con prácticas administrativas y técnicas seleccionadas para apoyar el desarrollo exitoso de software y otras formas de sistemas. Cómo se verá más adelante, los Casos de Uso 2.0 son:

- *Livianos*
- *Escalables*
- *Versátiles*
- *Fáciles de usar”*

[14]

Siguiendo la filosofía SCRUM, los casos de uso 2.0 pueden seguir un ciclo de vida iterativo e incremental, pudiendo evolucionar el mismo durante todo el proceso de desarrollo.

Los casos de uso no vienen a sustituir a las historias de usuario. Las historias de usuario expresan el *qué* se quiere conseguir, mientras que los casos de uso describen el *cómo* se empleará el sistema para conseguirlo.

2.7.1. Casos de uso de operaciones CRUD

Las operaciones *CRUD* se refieren a los casos de uso que describen las operaciones de *Creación*, *Lectura (Read)*, *Actualización (Update)* y *Borrado (Delete)*.

Como son operaciones muy frecuentes se describirá un patrón genérico que es aplicable a cualquier ámbito de la aplicación.

ID	1
Nombre	Crear
Descripción	Creación de una instancia del objeto X
Actores	-
Operaciones	
1	Seleccionar menú del objeto X
2	Seleccionar "Nuevo X"
3	Comprobar permisos
4	Rellenar los datos del nuevo objeto X (mínimo, los obligatorios)
5	Comprobar datos
6	Guardar instancia
Alternativas	
3.A	¿Permisos adecuados para el usuario?
3.A.1	Sí: continuar
3.A.2	Sí, pero restringidos: aplicar restricciones y continuar
3.A.3	No: salir
5.A	¿Datos correctos?
5.A.1	Sí: continuar
5.A.2	No: volver a 3

El caso de la lectura de registros se puede dividir en dos casos: listar un resumen de todas las instancias o ver en detalle una de ellas:

ID	2
Nombre	Listar
Descripción	Listar un resumen de los objetos X
Actores	-
Operaciones	
1	Seleccionar menú del objeto X
2	Seleccionar el tipo de listado (“Listar todos”, “Listar propios”...)
3	Mostrar listado
Alternativas	
2.A	¿Permisos adecuados para el usuario?
2.A.1	Sí: continuar
2.A.2	No: salir

ID	3
Nombre	Ver
Descripción	Visualizar una instancia del objeto X
Actores	-
Operaciones	Seleccionar menú del objeto X
1	Seleccionar instancia
2	Comprobar permisos
3	Visualizar instancia
Alternativas	
2.A	¿Existe la instancia?
2.A.1	Si: continuar
2.A.2	No: devolver error y salir
3.A	¿Permisos adecuados para el usuario?
3.A.1	Si: continuar
3.A.2	No: devolver error y salir

ID	4
Nombre	Editar
Descripción	Actualizar datos de una instancia del objeto X
Actores	-
Operaciones	
1	Seleccionar menú del objeto X
2	Seleccionar instancia
3	Comprobar permisos
4	Modificar los datos de la instancia
5	Comprobar datos
6	Guardar instancia
Alternativas	
2.A	¿Existe la instancia?
2.A.1	Si: continuar
2.A.2	No: devolver error y salir
3.A	¿Permisos adecuados para el usuario?
3.A.1	Si: continuar
3.A.2	No: devolver error y salir
5.A	¿Datos correctos?
5.A.1	Sí: continuar
5.A.2	No: volver a 4

ID	5
Nombre	Eliminar
Descripción	Eliminar una instancia del objeto X
Actores	-
Operaciones	
1	Seleccionar menú del objeto X
2	Seleccionar instancia
3	Comprobar permisos
4	Eliminar instancia
Alternativas	
2.A	¿Existe la instancia?
2.A.1	Si: continuar
2.A.2	No: devolver error y salir
3.A	¿Permisos adecuados para el usuario?
3.A.1	Si: continuar
3.A.2	No: devolver error y salir

El resto de casos de uso tendrían una mecánica muy parecida. Desde la ventana de visualización de alguna instancia permite ver el listado de instancias de otros objetos que estén relacionados o crear nuevas. En el caso específico de éste proyecto los casos de uso 2.0 no dan valor añadido al análisis, por lo que se descarta la necesidad de desarrollar nuevos.

2.8. Diagrama Entidad-Relación

En el siguiente diagrama se representan las clases más relevantes y cómo están interrelacionadas entre ellas:

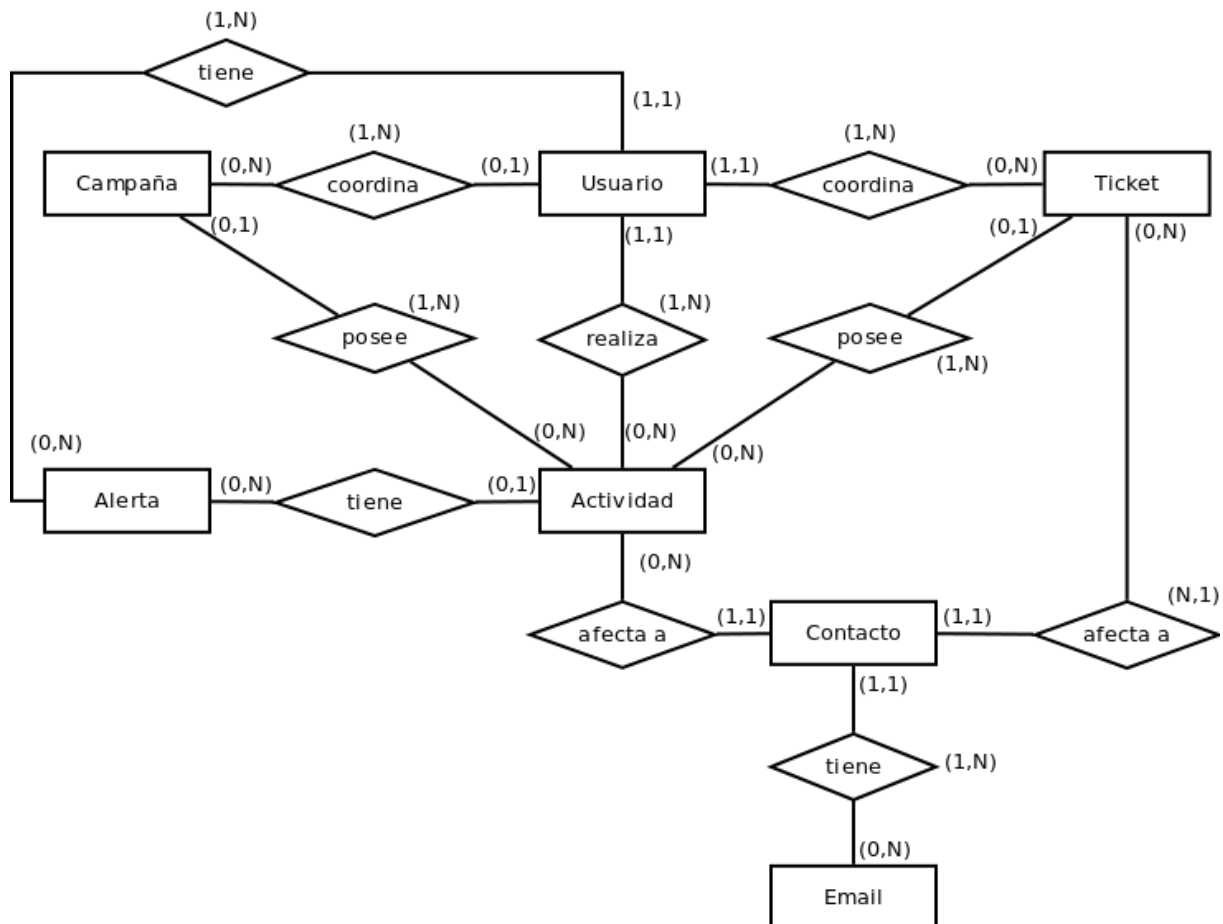


Ilustración 4: Diagrama Entidad-Relación Lógico

Los elementos centrales de toda la lógica son los *Usuarios* y *Actividades*. Un usuario realiza *Actividades*, las cuales afectan a *Contactos* de la organización.

Una *Actividad* puede ser independiente, o bien, estar relacionada con una *Campaña* de la organización o un *Ticket* de soporte de un *Contacto*; pero no las dos a la vez. Esto será algo que se deberá de comprobar en los controladores.

Adicionalmente, se puede programar *Alertas* asociadas a una *Actividad* para que aparezcan de manera emergente en la aplicación, o bien, se envíen por correo electrónico al *Usuario* asociado a la misma. Una *Alerta* también puede ser independiente y no estar asociada a ninguna *Actividad* concreta.

3. DISEÑO

El diseño y el análisis de un sistema son dos conceptos muy distintos, aunque relacionados:

El **análisis** indica **qué** debe hacer el sistema. El analista debe entender la organización, los requisitos y los objetivos a cumplir.

El **diseño** especifica **cómo** debe hacerse. El diseñador busca especificar un sistema que se ajuste a las necesidades o provea sus requisitos de una manera eficaz y permita alcanzar los objetivos.

En el apartado de [planificación de tareas](#) se propuso seguir una metodología ágil para el desarrollo de la aplicación. Martin Fowler afirma en [15] que el hecho de seguir una metodología ágil no significa dar de lado la burocracia excesiva ni la gran cantidad de documentación de las metodologías tradicionales de desarrollo, pero tampoco seguirlas al pie de la letra. La clave está en encontrar el compromiso de “suficiente documentación”.

Al contrario que con las ingenierías tradicionales en las que encontramos poco esfuerzo de diseño y gran esfuerzo en construcción; en la ingeniería de software el esfuerzo puede llegar a invertirse, suponiendo mucho mayor esfuerzo el diseño que la propia construcción de la aplicación.

Un buen diseño se demuestra, sobre todo, a largo plazo. Durante el mantenimiento y ampliación del software un diseño correcto suficientemente documentado ayuda a “retomar el hilo” sin excesivos esfuerzos.

3.1. Diagrama de clases

Apoyándonos en el [diagrama entidad-relación](#) podemos realizar un diagrama de clases. En ésta fase hemos de empezar a modelar las clases que tendrá la aplicación, así como las relaciones y clases de apoyo que necesitemos.

En la siguiente figura se puede observar el diagrama de clases general que ha guiado el proceso de implementación de la aplicación:

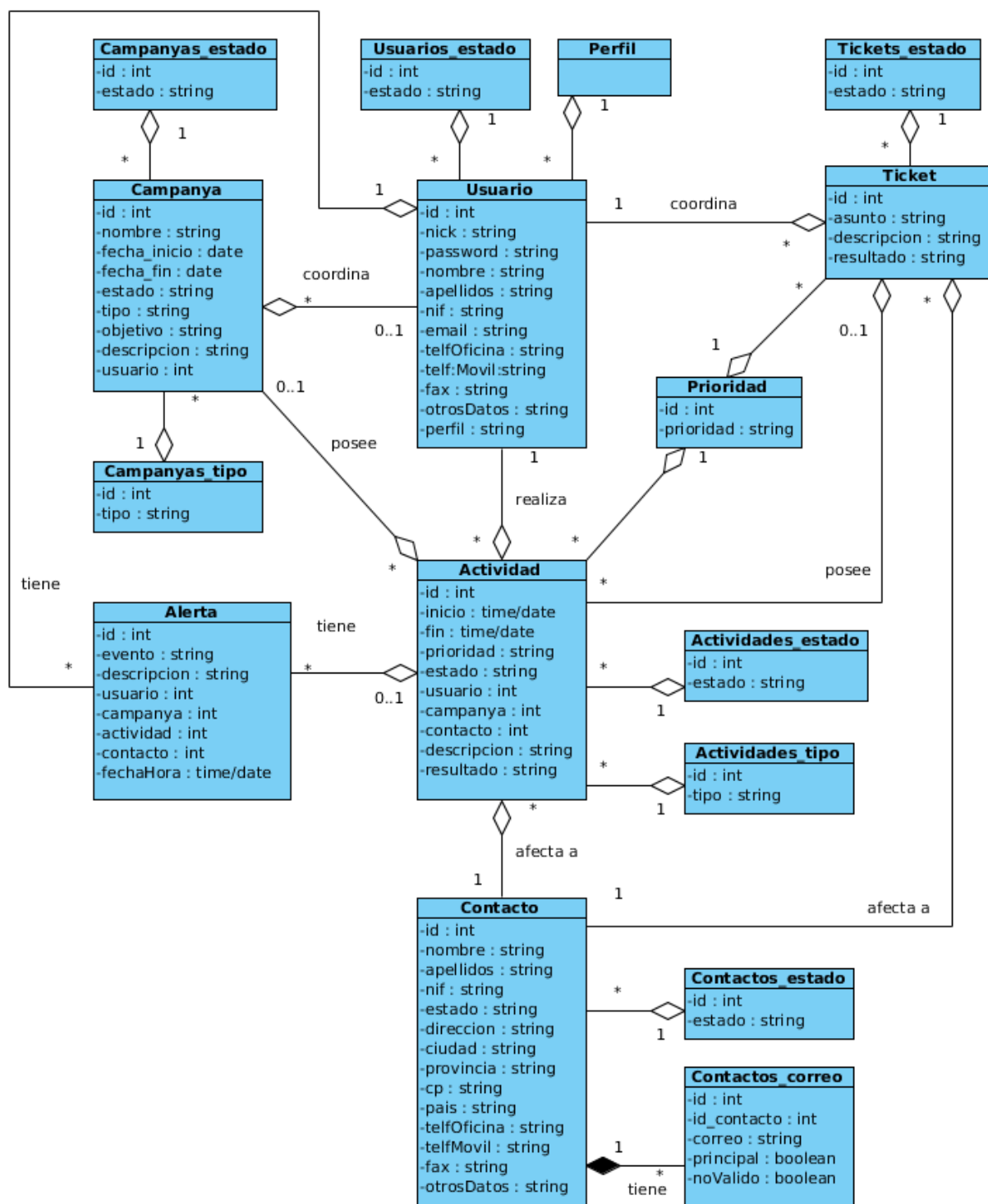


Ilustración 5: Diagrama de clases

3.2. Diseño de la interfaz

Desde los [requisitos no funcionales](#) de la herramienta se definen varios que están directamente relacionados con la interfaz. Ésta debe cumplir los siguientes objetivos principales respecto a la usabilidad:

- **Fácil de aprender:** El tiempo desde el no conocimiento hasta su uso productivo debe ser mínimo. Los operadores gráficos deben ser sintetizables y familiares.
- **Consistencia:** Todos los mecanismos son siempre usados de la misma manera. Esto se consigue siguiendo un *Look&Feel* común.
- **Disminuir la carga cognitiva:** Favorecer el reconocimiento frente al recuerdo de las operaciones que se pueden realizar en cada momento.

Usando la herramienta *Moqups* de prototipado se ha confeccionado una serie de bocetos para entender la disposición de los elementos en la pantalla:

The wireframe illustrates a user interface layout. At the top is a navigation bar with a 'logo' and six menu items: 'Menú 1', 'Menú 2', 'Menú 3', 'Menú 4', 'Menú 5', and 'Menú 6'. On the right side of the navigation bar are a notification bell icon and a user profile icon labeled 'Hola Usuario'. Below the navigation bar is a main content area. On the left side of this area is a search bar with the placeholder text 'inserta aquí la cadena a buscar...' and search and close icons. To the right of the search bar is a table with five columns, each labeled with a dropdown arrow and the letter 'H'. The table contains nine rows of data, each with a number in the first column and 'xxxxx' in the others. Below the table is a pagination control with buttons for '<', '1', '2', '3', '4', '>', and '>>'. To the right of the table is a vertical list of options, grouped into two sections. The first section contains five options labeled 'Opción 1.1' through 'Opción 1.5'. The second section contains two options labeled 'Opción 2.1' and 'Opción 2.2'. At the bottom of the main content area is a footer labeled 'Footer'.

Ilustración 6: Diseño wireframe de listados

Ilustración 7: Diseño wireframe de formularios

En ambos ejemplos se puede observar cuatro zonas principales:

- **Área superior:** Tiene un menú general
- **Área central:** A su vez, dividida en dos secciones.
 - **Zona central-izquierda:** Área de trabajo donde se mostrará el elemento actual sobre el que se está navegando en la aplicación.
 - **Zona derecha:** Menú secundario de operaciones. Éste menú cambiará mostrando las posibles operaciones según la opción del menú general que se seleccione.
- **Área inferior:** Pié de página donde se puede insertar información que será visible en todas las páginas (al igual que el menú superior), pero que no tiene protagonismo en el día a día del uso de la aplicación ni es de vital importancia conocerla.

3.2.1. Responsive design

También se indicó anteriormente que la interfaz debía ser respetuosa con todos los posibles dispositivos que utilizarasen la herramienta y por ello debe usar una filosofía *responsive design*.

De esta manera se decide que, tanto el menú superior como el lateral derecho queden colapsados cuando la ventana de visualización sea menor de un cierto

tamaño. Será pulsando un botón bien visible cuando se muestre el menú que se desee.

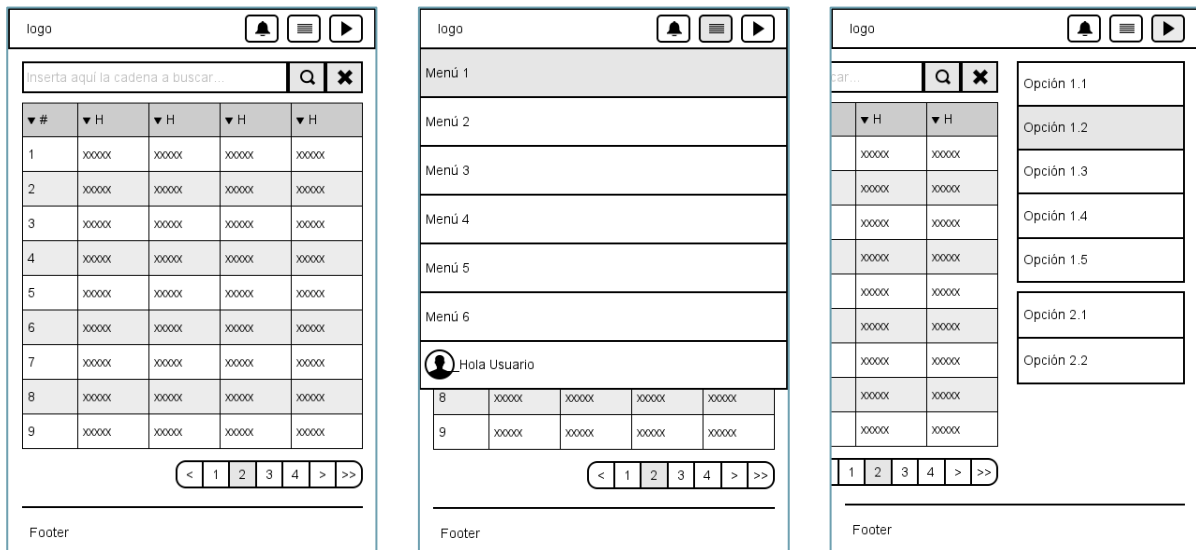


Ilustración 8: Diseños *responsive design* para pantallas pequeñas

3.2.2. Ejemplos reales de la interfaz

Con el diseño general que se desea de la interfaz y haciendo uso del framework Bootstrap 3 se diseña la disposición general de los elementos de la interfaz.

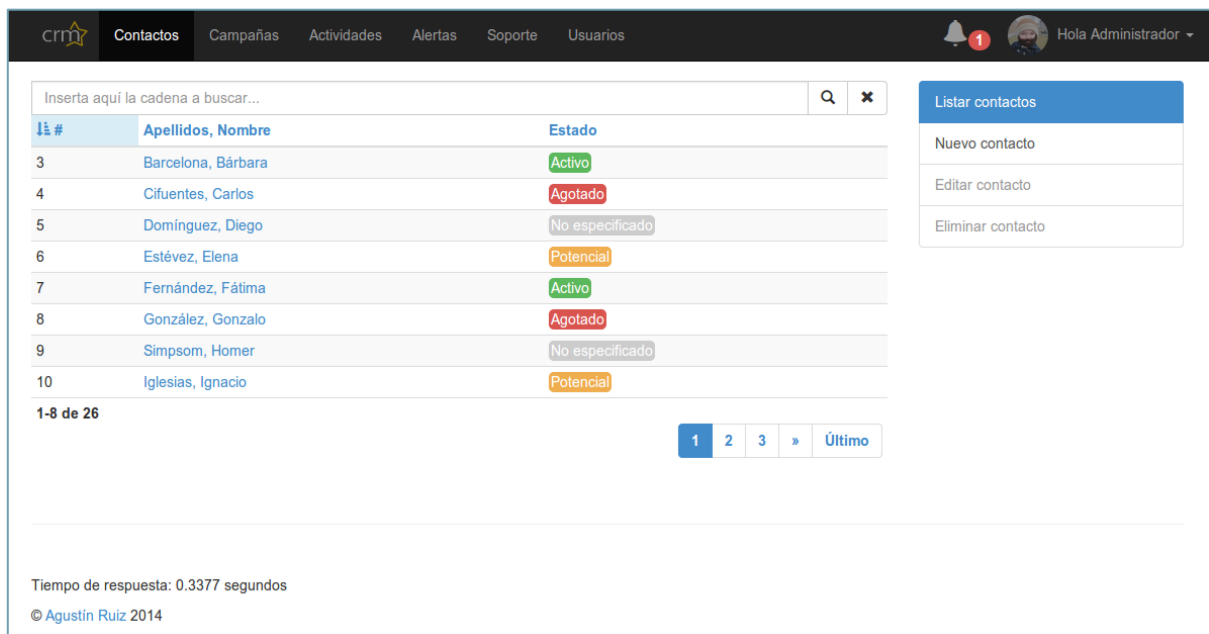


Ilustración 9: Interfaz real de un listado

CRM Contactos Campañas Actividades **Alertas** Soporte Usuarios

Hola Administrador

Asunto*

Fecha/Hora*

Descripción

Enviar email*

Ventana emergente*

Actividad

Los campos marcados con un asterisco (*) son obligatorios

Listar alertas

Nueva alerta

Editar alerta

Eliminar alerta

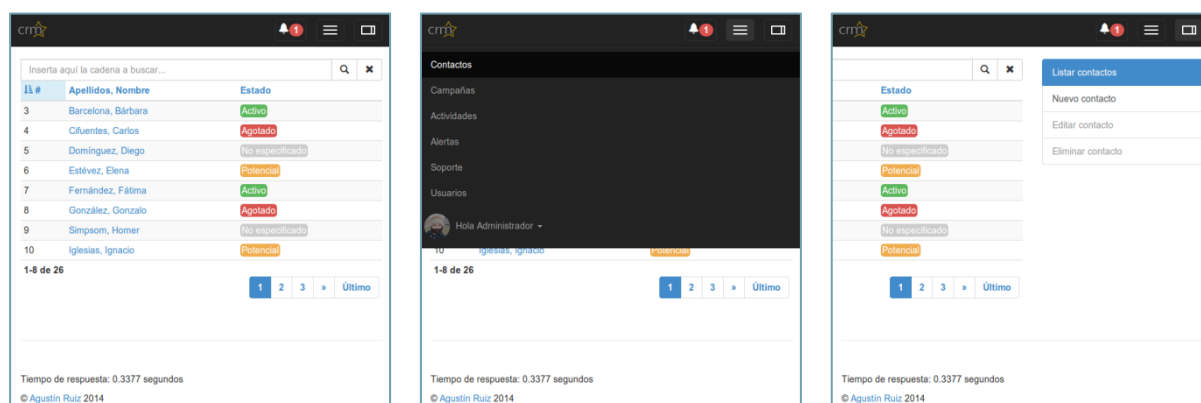
Tiempo de respuesta: 0.1079 segundos

© Agustín Ruiz 2014

Ilustración 10: Interfaz real de un formulario

Para nuestro caso, se ha utilizado un diseño ya creado en Bootstrap llamado *Offcanvas*⁹ y se ha adaptado a las necesidades del diseño que se requería en el análisis.

En las siguientes ilustraciones se puede ver el resultado real de la interfaz *responsive*.

Ilustración 11: Interfaz real *responsive design*

⁹ <http://getbootstrap.com/examples/offcanvas/>

3.3. Perfiles de usuario

No todos los usuarios de la aplicación deberían tener el mismo nivel de acceso. Por ello, se predefinen unos perfiles de usuario con distintos niveles de acceso. Estos perfiles son los siguientes:

Permiso	Superusuario	Coordinación	Comercial	Técnico
Listar contactos	SÍ	SÍ	SÍ	SÍ
Crear contactos	SÍ	SÍ	SÍ	NO
Editar contactos	SÍ	SÍ	SÍ	NO
Eliminar contactos	SÍ	SÍ	NO	NO
Listar campañas propias	SÍ	SÍ	SÍ	SÍ
Listar todas las campañas	SÍ	SÍ	SÍ	SÍ
Crear campañas	SÍ	SÍ	NO	NO
Editar campañas	SÍ	SÍ	NO	NO
Eliminar campañas	SÍ	SÍ	NO	NO
Listar todas las actividades	SÍ	SÍ	NO	NO
Crear actividades	SÍ	SÍ	NO	NO
Editar actividades	SÍ	SÍ	NO	NO
Eliminar actividades	SÍ	SÍ	NO	NO
Listar actividades propias	SÍ	SÍ	SÍ	SÍ
Crear actividades propias	SÍ	SÍ	SÍ	SÍ
Editar actividades propias	SÍ	SÍ	SÍ	SÍ
Eliminar actividades propias	SÍ	SÍ	SÍ	SÍ
Listar todos los tickets de soporte	SÍ	SÍ	SÍ	NO
Listar tickets de soporte propios	SÍ	SÍ	SÍ	SÍ
Crear tickets	SÍ	NO	NO	NO
Crear tickets propios	SÍ	SÍ	SÍ	SÍ
Editar tickets	SÍ	NO	NO	NO
Editar tickets propios	SÍ	SÍ	SÍ	SÍ
Eliminar tickets	SÍ	NO	NO	NO
Eliminar tickets propios	SÍ	SÍ	SÍ	SÍ
Listar usuarios	SÍ	SÍ	SÍ	SÍ
Crear usuarios	SÍ	NO	NO	NO
Editar usuarios	SÍ	NO	NO	NO
Eliminar usuarios	SÍ	NO	NO	NO
Listar todos los perfiles	SÍ	SÍ	NO	NO

4. IMPLEMENTACIÓN

Una vez que tenemos claro el diseño podemos comenzar la fase de implementación de la aplicación web. Aunque no es 100% necesario, porque en desarrollo ágil basta con tener diseñado lo que se implementará en el futuro inmediato pero con la vista puesta a las consecuencias que pueda tener en el diseño (o cambio de diseño) en fases posteriores.

4.1. Arquitectura

La arquitectura de software define y organiza un sistema en términos de sus componentes, subsistemas y detalles de diseño. Tiene los siguientes objetivos:

- Analizar la eficacia (o no) del diseño para la consecución de los requisitos propuestos.
- Considerar alternativas arquitectónicas.
- Reducir los riesgos del desarrollo.

Con un sistema ciertamente complejo la estructura del software en módulos, subsistemas y componentes es necesario para controlar los flujos de información y acceso entre ellos.

En primer lugar debemos entender la infraestructura general que tendrá la aplicación una vez se despliegue:

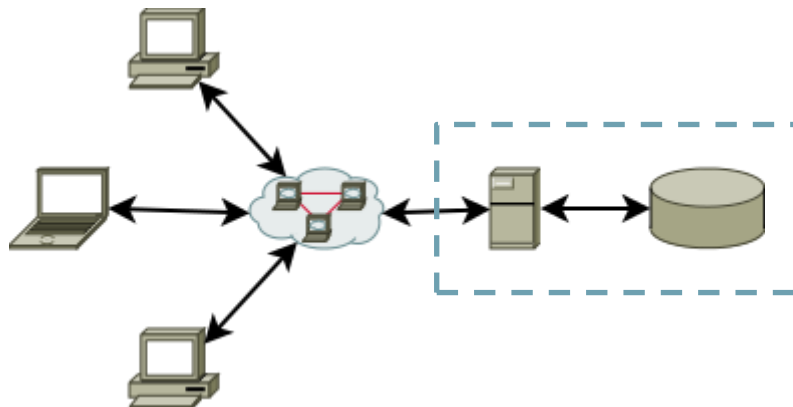


Ilustración 12: Infraestructura general

Desde cualquier dispositivo con navegador web y conexión a internet puede acceder al servidor que contenga la aplicación instalada. Ésta, a su vez, se conectará a una base de datos donde se almacena todos los datos generados de la utilización de la herramienta. Por tanto, entraremos sólo en el detalle de la arquitectura que definiremos es la que comprende el área señalada.

La arquitectura del servidor está estructurada por capas según el siguiente esquema de arquitectura:

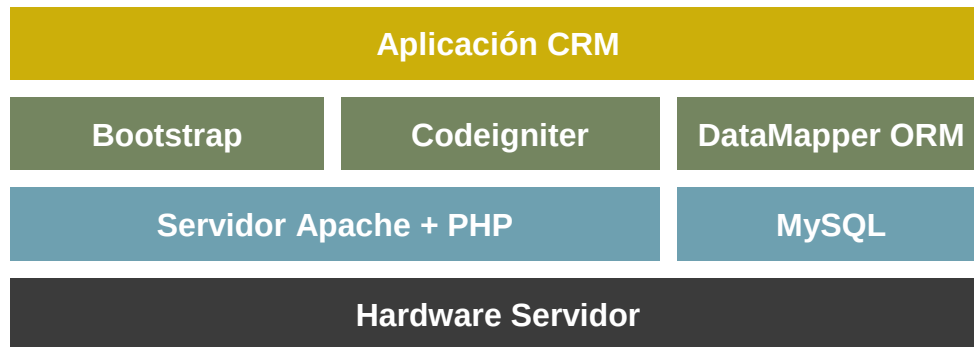


Ilustración 13: Arquitectura del servidor

Sobre el servidor que aloja la aplicación está instalado el servidor Apache+PHP y el gestor de base de datos MySQL. En ellos están instalados los framework en los que se apoya el desarrollo. En especial, cabe destacar la biblioteca DataMapper ORM, que será la que gestione las comunicaciones con la base de datos.

El funcionamiento general del framework Codeigniter se resume en el siguiente diagrama de flujo:

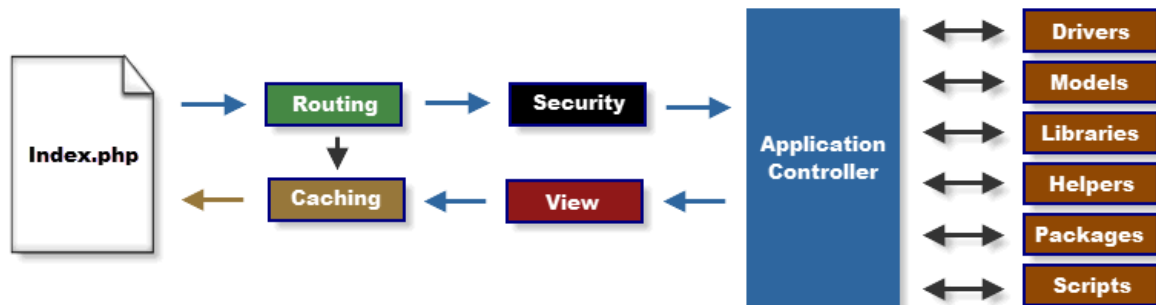


Ilustración 14: Diagrama de flujo de Codeigniter (fuente [16])

El framework se encarga de gestionar los módulos principales de manera autónoma. Como se indica en [16], los datos fluyen de la siguiente manera:

4. La página **index.php** sirve como controlador frontal, inicializando los recursos básicos que necesita Codeigniter para ejecutar.
5. El módulo de **Routing** examina la solicitud HTTP para determinar que debería hacer con ella.
6. Si existe el archivo de **caché**, se lo envía directamente al navegador, sin pasar por la ejecución normal del sistema.

7. **Seguridad.** Antes que se cargue el controlador de la aplicación, por razones de seguridad se filtran la solicitud HTTP y cualquier otro dato enviado por el usuarios.
8. El **controlador** carga el **modelo**, las bibliotecas del núcleo, helpers, y cualquier otro recurso requerido para procesar una solicitud específica.
9. La **Vista** terminada se procesa y se envía al navegador para que se pueda ver. Si el caché está habilitado, la vista se cachea primero para que las siguientes solicitudes que la necesiten puedan ser servidas.

4.2. Detalles de la implementación

El aceptar la utilización de un framework siempre supone el tener que acatar ciertas normas de implementación. Sin embargo, aunque para Codeigniter se han publicado manuales de buenas prácticas, como el que podemos observar en [\[17\]](#), no son totalmente obligatorias. La herramienta es lo suficientemente flexible como para aceptar implementaciones que se salgan de la línea de la línea recomendada, siempre en favor de las decisiones de diseño y programación del propio desarrollador.

La recomendación que sí se va a tener en cuenta por ser una práctica muy aceptada en el desarrollo de aplicaciones orientadas a objetos es el seguimiento de una arquitectura Modelo-Vista-Controlador.

4.2.1. Controladores

La definición de los controladores se realiza en el directorio *application/controllers*. Para cada controlador se crea un archivo PHP que contiene las funciones que determinan cómo se manejan las solicitudes HTTP.

Cada controlador debe tener un constructor (que puede omitirse por defecto), una función *index* de acceso por defecto y otras funciones para acceder a ellas. Por ejemplo, para acceder a la función por defecto del controlador *Contactos* entraremos en la siguiente URL:

```
http://ejemplo.com/index.php/contactos
```

Las funciones que se definan pueden recibir parámetros usando distintos segmentos en la URI. Por ejemplo, para visualizar el contacto con *id=3* usaremos una URL similar a la siguiente:

```
http://ejemplo.com/index.php/contactos/ver/3
```

También puede indicarse un valor por defecto al parámetro para que se use ese en caso de que no se indique en la URI.

Adicionalmente podemos omitir la entrada *index.php* de las URL creando un fichero de configuración *.htaccess* en la raíz del directorio de la aplicación que redirija cualquier petición añadiéndolo de manera transparente a la página *index.php*. Suponiendo que la aplicación esté instalada en el directorio */crm* del servidor, el fichero básico de configuración sería similar al siguiente:

```
<IfModule mod_rewrite.c>
    RewriteEngine On
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteRule . $ /crm/index.php/$1 [PT, L]
</IfModule>
```

Cabe destacar que en todas las funciones de todos los controladores se realiza una comprobación de permisos de ejecución de la misma, redirigiendo a una vista de “no disponible” si no procede el acceso al recurso solicitado.

Los controladores que se han desarrollado han sido los siguientes:

- **Main**: Controlador por defecto (menú principal).
- **Contactos**: Controlador de contactos.
- **Campañas**: Controlador de campañas.
- **Actividades**: Controlador de actividades.
- **Alertas**: Controlador de alertas.
- **Tickets**: Controlador de tickets.
- **Usuarios**: Controlador de usuarios.
- **Perfiles**: Controlador de perfiles. Cabe destacar que éste controlador no permite la creación de nuevos perfiles: sólo consultar los existentes y, en su caso, los usuarios asignados al perfil en cuestión.
- **Login**: Controlador de identificación en la aplicación.
- **Cron**: Controlador para la ejecución de tareas programadas como, por ejemplo, envío automático de correos electrónicos.
- **Error404**: Controlador para la generación de la vista de la página personalizada de error 404. Se usa en aquellos casos en los que no se encuentra el objeto que

se busca, por ejemplo, cuando se intenta recuperar un contacto con un id que no existe en la base de datos.

Todos los controladores tienen una función para cada operación que se puede realizar. De manera general, todos los controladores tienen funciones necesarias para las operaciones CRUD (excepto perfiles).

Adicionalmente, se ha estimado oportuno extraer en ficheros separados algunas funciones que se llaman haciendo uso del comando *include* de PHP. Son los siguientes:

- ***include_mail***: Funciones de envío de correo electrónico usando el comando *mail* de PHP.
- ***include_password***: Funciones de generación de contraseñas aleatorias y *hashes* para almacenar en la base de datos y comparar en el login.

4.2.1.1. Login y variables de sesión

El caso del controlador de *login* no es complicado, pero sí digno de mención. Posee simplemente dos funciones: una de *login* y otra de *logout*.

La función de *login* consulta el modelo de usuarios el busca de un usuario cuyo identificador y hash de la contraseña introducida coincidan con los introducido en el formulario de acceso. Si existe, se crea una variable de sesión de Codeigniter con los datos básicos del usuario, así como los permisos de su perfil.

Será de la variable de sesión de la que se consulten los permisos de acceso, ahorrando consultas a la base de datos en cada llamada a las funciones de los controladores.

Será en todos los constructores de los controladores con acceso restringido donde se compruebe si existe una sesión activa. De no ser así, redirigirá automáticamente al controlador de *login*.

La función de *logout* simplemente limpia la variable de sesión y redirige al controlador de *login*.

4.2.2. Modelos

Los modelos son clases de PHP que se diseñan para trabajar con la información almacenada en la base de datos. Para una gestión más cómoda de los objetos en la base de datos se ha decidido usar una biblioteca de Mapeado Relacional de Objetos para Codeigniter llamado *DataMapper*. Podemos encontrar toda la documentación necesaria acerca de *DataMapper* en [\[18\]](#).

Los requisitos mínimos que necesita son que el servidor ejecute una versión de PHP igual o superior a 5.1.2, que se use una versión de Codeigniter igual o superior a 1.7.2 y que se use una base de datos compatible con el framework base.

La propia biblioteca se encarga de la carga de los modelos durante la ejecución y no hace falta hacerlo de manera expresa en los controladores.

Su uso es muy sencillo: se crea una instancia del objeto a utilizar y se personaliza la instancia usando los métodos que incluye la propia clase *DataMapper*. Posee los métodos más usuales para ejecutar operaciones CRUD sobre las instancias. También se puede construir consultas más complejas mediante el anidamiento de operadores. Todas las posibilidades están indicadas mediante ejemplos en la documentación de su web oficial.

4.2.2.1. Tablas en la base de datos

El ORM DataMapper está implementado según la Tercera Forma Normal. Resumiendo, significa que cada tabla tiene solamente constancia de su propia existencia, con sus atributos más significativos en ella misma.

Cada relación (aunque no es obligatorio) se almacena en una tabla especialmente destinada a ello, teniendo cada una tres campos: una para el *ID* del registro, otra para el *ID* del primer elemento relacionado y otra para el *ID* del segundo elemento de la relación.

En el fichero de configuración de DataMapper permite configurar un prefijo para las tablas y otro prefijo para identificar las tablas que almacenan las relaciones entre registros.

4.2.2.2. Clases PHP de los modelos

La plantilla genérica usada para crear modelos en DataMapper ha sido la siguiente:

```
class Clase extends DataMapper{
    var $table = 'clases'; // Nombre de la tabla
    // var $model = 'template'; // Nombre del modelo
    var $cascade_delete = false;
    var $auto_populate_has_many = true;
    var $auto_populate_has_one = true;

    // Constructor
```

```
function __construct($id = null){
    parent::__construct($id);
}

// Relaciones
public $has_one = array(/*...*/);
public $has_many = array(/*...*/);

// Validación de campos
public $validation = array(/*...*/);
}
```

En primer lugar se puede observar que la clase debe heredar de la clase principal `DataMapper`.

Adicionalmente, se puede indicar el nombre de la tabla que con la que se corresponde en la base de datos (sin el prefijo) y el nombre del modelo que se utilizará. Si se omite, `DataMapper` usará como nombre de tabla y del modelo el mismo nombre de la clase en minúscula. En el caso del nombre de tabla, añadirá una “s” para hacerlo plural. `DataMapper` recomienda que los nombre de las tablas sean en plural, por eso añade una “s”, pero como algunos plurales no son regulares es necesario sobrescribir ésta variable para hacer el código más legible.

La variable `$cascade_delete` igualada a *false* evita que las relaciones de un objeto se eliminen con el objeto, teniendo que indicarlo expresamente en la construcción de la base de datos si fuese necesario. Por defecto, el valor es *true*.

Las variables `$auto_populate_has_one` y `$auto_populate_has_many` activan o desactivan la *carga en cascada* o la *carga perezosa* cuando su valor es *true* o *false* (respectivamente).

Las relaciones con otros objetos se indican en las variables `$has_one` y `$has_many`, ya sean de *1-a-1* o *1-a-varios* (respectivamente). Se especifican mediante un array con los nombres de las tablas.

Una de las grandes ventajas del uso del ORM `DataMapper` es la validación automática de campos. Se indica en el array `$validation`. Cada elemento debe tener el siguiente formato:

```
'campo' => array(
```

```
'label' => 'etiqueta del campo',  
  'rules' => array('required', 'min_length' => 3,  
    'max_length' => 100)  
)
```

Las reglas de validación son variadas e, incluso, pueden crearse para los objetos relacionados. Todas ellas pueden consultarse en la documentación de DataMapper.

Además de las reglas de validación es importante indicar una etiqueta para el campo. Esto es porque DataMapper genera automáticamente los mensajes de error integrando dicha etiqueta en la entrada.

4.2.3. Vistas

Una vista en Codeigniter es una página web o fragmento de página que se llama desde un controlador. Las vistas no pueden ser llamadas directamente, sino que debe ser siempre el controlador el que haga referencias a ellas.

Varias vistas pueden ser llamadas desde un controlador, pudiendo crear una jerarquía de fragmentos de páginas que son llamadas en función de las necesidades. Por ejemplo, se puede separar fragmentos comunes a todas las páginas (como el encabezado y pie de página), lo que minimiza el código duplicado en las vistas. En nuestro proyecto se ha dividido el contenido en cuatro vistas:

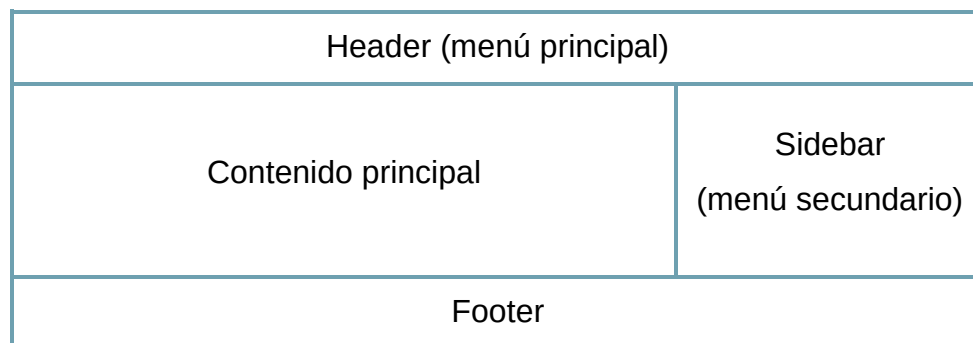


Ilustración 15: Estructura general de las vistas

Para cargar una vista se llama directamente al archivo sin especificar la extensión PHP. Las vistas pueden organizarse por carpetas. Opcionalmente se puede pasar parámetros a la vista para ser mostrados en ella en como un objeto como un array de objetos. La orden genérica para cargar una vista es de la siguiente manera:

```
$this->load->view('ruta_carpeta/nombre_archivo', $datos);
```

Cada uno de los elementos del array *\$datos* son accesibles en la vista como variables clásicas de PHP. Un elemento puede ser una colección de objetos, de tal manera que pueden recorrerse con un bucle, por ejemplo, para mostrar en una tabla todos los resultados de una consulta.

4.2.3.1. Framework Bootstrap 3

Para el maquetado de todas las vistas se ha utilizado el framework Bootstrap 3. Con él se ha conseguido una interfaz consistente, responsive design, clara y eficaz.

Todos los ficheros relacionados con el framework se han colocado en la raíz de la aplicación. Todos ellos se encuentran en los directorios *css*, *fonts*, y *js*.

Además de las clases que ya incorpora se han creado nuevos estilos de visualización al estilo visual de Bootstrap, así como otras funciones JavaScript. Ellos han sido insertados en ficheros separados para un mejor mantenimiento y en las mismas carpetas de recursos de Bootstrap. Adicionalmente, también se ha hecho uso de otros recursos que han sido insertados en los directorios *img* y *audio*.

Por último, ha sido necesario redefinir el fichero *.htaccess* de traducción de los controladores a la página *index.php* para que ignore estos casos y los interprete como controladores.

4.2.3.1.1. Componentes Bootstrap

Bootstrap incluye una serie de componentes basados en jQuery. De entre ellos, se han usado los siguientes:

4.2.3.1.2. Modal

Las ventanas modales se usan en varios lugares de la aplicación. Se utilizan para mostrar información o confirmar una acción sin necesidad de recargar la página completa.

Uno de los lugares más socorridos donde son usados es para la **eliminación** de registros. Al pulsar la opción *eliminar* se muestra un modal de confirmación. Si se acepta, la página se redirige al controlador de eliminación.

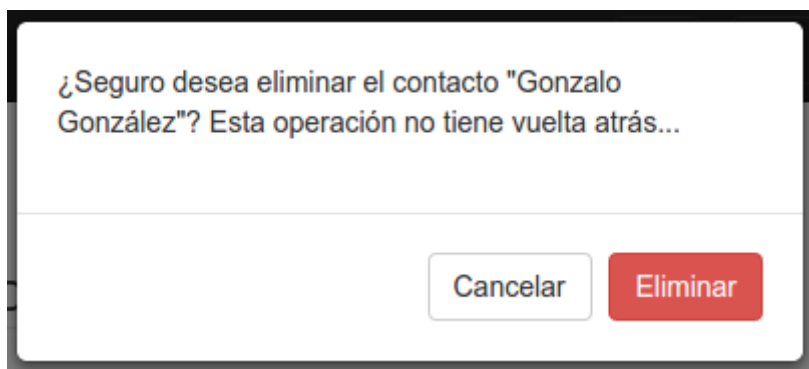


Ilustración 16: Ejemplo de modal de eliminación

Otro de los lugares donde se utiliza es como ventanas de ***selección de un registro relacionado***. Por ejemplo, para seleccionar el *contacto* relacionado con una *actividad*. El modal contiene un *iframe* a una página auxiliar de búsqueda de registros. Cada línea tiene un botón que modifica los campos adecuados en el marco superior y cierra el modal.

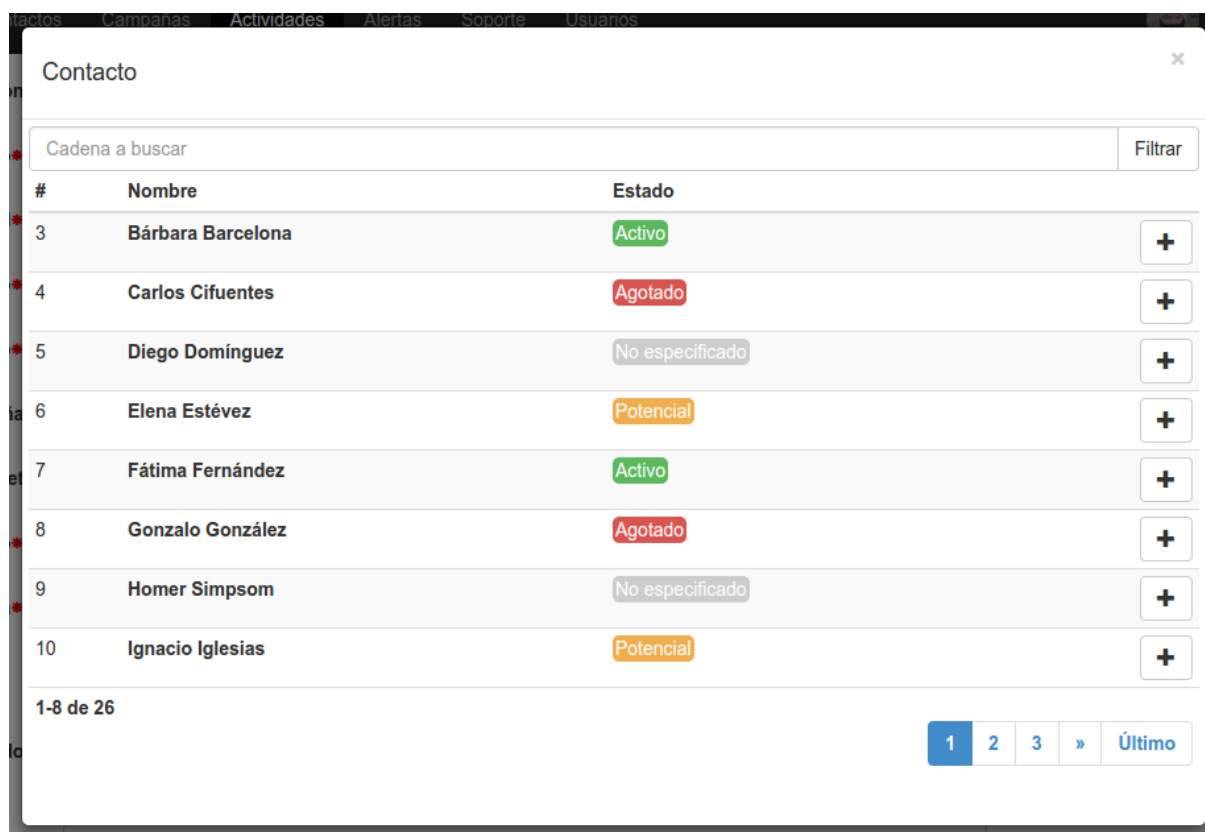


Ilustración 17: Ejemplo de modal de selección de registro relacionado

Por último, otro lugar donde se usan los modales es en las **ventanas emergentes de alertas**. Durante el uso de la herramienta, si una alarma ha sido programada, aparecerá un modal avisando del evento.

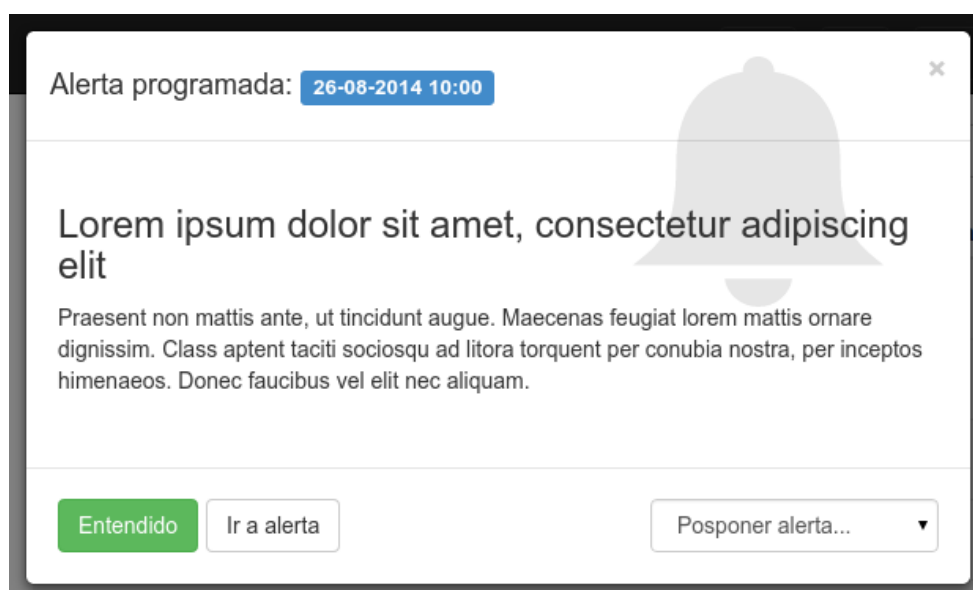


Ilustración 18: Ejemplo de modal alerta emergente

El sistema de alertas será detallado más adelante.

4.2.3.1.3. Widget DateTimePicker

Una de las grandes carencias de Bootstrap es la falta de un selector de fecha y hora propio integrado en el framework. Sin embargo, es algo que la comunidad de usuarios ha solucionado.

En la aplicación web hemos seleccionado el widget desarrollado por Eonasdan y cuyo desarrollo y documentación podemos consultar en [80].

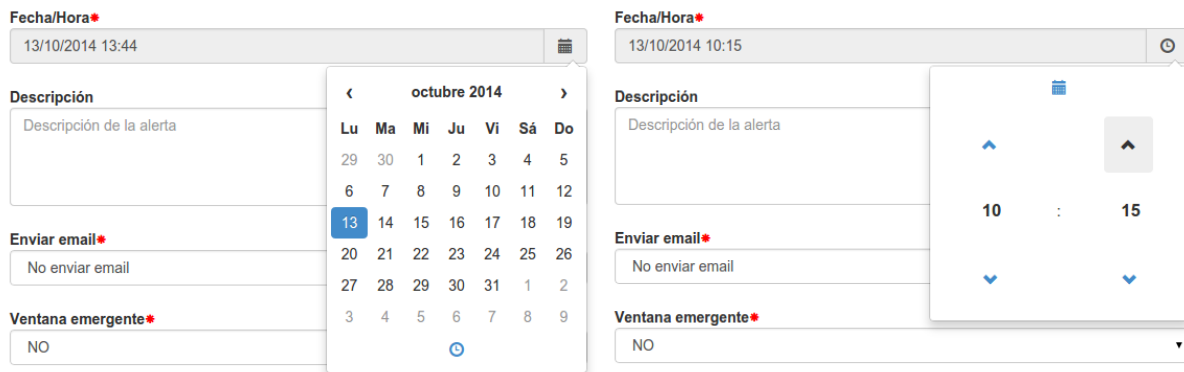


Ilustración 19: Ejemplo de widgets datetime picker

4.2.3.2. Sistema de alertas

El sistema de notificaciones de alertas requiere un estudio para su implementación usando MVC, ya que requiere un método de propagación para que la vista sepa qué alerta debe mostrar y cuándo debe hacerlo.

El método que se ha estimado mejor es el de *escrutinio* o *polling*. En la ventana de información se ejecuta una función de JavaScript que consulta un controlador que le devuelve un objeto en formato JSON. Si el objeto es una alerta la muestra en el modal; si el objeto es nulo no se realiza ninguna acción. Ésta función debe consultar el controlador cada cierto tiempo.

Éste método de propagación puede llegar a hacer un uso abusivo de las comunicaciones con el sistema, tal y como se indica en [90], por ello es importante procurar que las comunicaciones sean lo más. Respecto a la sobrecarga de las consultas sobre la base de datos no hay otra opción, ya que aunque se eligiese un método de *subscripción* de la vista al modelo seguiría siendo necesaria la consulta a la base de datos para saber qué alerta está pendiente de ser mostrada. Por tanto, la única solución es llegar a un compromiso de sobrecarga de comunicaciones y tiempo máximo de retraso en las notificaciones emergentes.

5. CONCLUSIONES

Para cualquier organización el tener una cultura centralizada en sus contactos y clientes puede suponer el éxito o el fracaso. Tener una buena metodología CRM garantiza la organización, control y planificación de comunicaciones con clientes potenciales y fidelizados, incluso, identificar aquellos que ya han sido agotados y en los que no merece la pena seguir invirtiendo recursos de la empresa.

La dirección que se decide tomar en un principio puede no gustar a los expertos puristas de la metodología CRM, porque puede parecer que se queda demasiado corto en la implementación de la metodología. Precisamente, ése era uno de los objetivos implícitos que me planteé para el proyecto: que la metodología no copase los procesos comunicativos del negocio, sino que fuesen un simple apoyo al registro y programación de los eventos con los clientes.

Mi experiencia con otros CRM (como *SugarCRM*) fueron muy decepcionantes por convertirse la herramienta en si como el fin: usar la aplicación era el objetivo principal de las comunicaciones. Ése planteamiento es erróneo bajo mi punto de vista, pues no debe ser el objetivo, sino el medio para conseguir el objetivo.

Es innegable que cualquier experto en materia de CRM echará en falta muchas opciones y funcionalidades para la herramienta; pero para una PYME opino que las ventajas en agilidad de los procesos y la flexibilidad que ofrece la aplicación web la convierten en una herramienta útil. Haciendo un símil con el concepto de postagilismo frente al agilismo tradicional: *“El postagilismo es simplemente hacer lo mejor para ti, por encima de una metodología ágil en detalle”* [100].

Con todo y con ello, los objetivos del proyecto han sido cumplidos: se ha estudiado las funcionalidades generales que debía tener la herramienta y se han detallado los requisitos en forma de historias de usuario. Tras esto se ha desarrollado un prototipo usando el framework PHP Codeigniter 2.2.0 y el framework Bootstrap 3 para la interfaz.

Desde un punto de vista personal la decisión de elegir frameworks para el desarrollo ha sido un acierto de pleno. Aunque los detalles de implementación y funcionamiento general de la aplicación están sujetos las decisiones de diseño de los framework elegidos, las ventajas son muy superiores a los inconvenientes. El hecho de no empezar desde cero ayuda a ver software funcionando en los primeros sprints sin demasiado esfuerzo. Aunque ya había desarrollado interfaces web

usando Bootstrap, para mí ha sido una experiencia nueva en el desarrollo basado en Codeigniter, ya que nunca antes había utilizado antes ningún framework PHP. Sin duda, el tiempo invertido para su aprendizaje no ha sido en balde y estoy seguro que lo seguiré usando en proyectos futuros en lo que tenga que usar el mismo lenguaje; al igual que rentabilicé el tiempo que dediqué a asimilar el uso de los estilos y componentes de Bootstrap.

Todo el proceso de implementación se ha realizado usando el paradigma de programación Modelo-Vista-Controlador y una metodología de desarrollo basada en SCRUM. No ha sido el primer proyecto desarrollado con metodologías ágiles en el tiempo que he estado en la Universidad de Jaén y estoy seguro que no será la última vez que lo haga. En la gran mayoría de ofertas de trabajo con el perfil de ingeniería de software demandan experiencia con SCRUM y también con la práctica de programación TDD o *Test-Driven-Development*.

La programación TDD no ha sido puesta en práctica en este proyecto, pero conozco la existencia del plugin *PHPUnit* para la creación de test unitarios PHP y guiar el proceso de desarrollo mediante test que deban cumplirse. Supongo que éste también sería otro caso de tiempo de aprendizaje bien invertido.

Si he de resaltar algún aspecto del proyecto es la simplicidad del diseño. El objeto de ello no es más que el que se ha expresado anteriormente de mantener una herramienta flexible, con operaciones sencillas y rápidas.

5.1. Mejoras y trabajos futuros

Hay que resaltar que el proyecto desarrollado no es más que un prototipo. Aunque funcional, le quedan bastantes “fleclos sueltos” que con relativo poco esfuerzo podrían ser cubiertos.

- Implementar el envío de alertas por correo electrónico a los usuarios. Por detalles del proveedor de hosting utilizado para las pruebas del prototipo no se ha podido programar mediante CRON el envío de correos.
- Repositorio documental relacionado con las entidades del sistema para almacenar en el servidor ficheros que puedan estar a disposición de cualquier usuario.
- Envío de correos electrónicos desde la plataforma a los contactos.
- Actualización masiva de contactos: cambio de estado, envío de correos...
- Personalización de la interfaz y el logotipo de la aplicación.

- Pantalla de vista pública para los tickets de soporte técnico. Éstos se enviarían por correo electrónico y el usuario podría consultar el estado en cualquier momento, así como hacer apreciaciones.
- Histórico de operaciones donde se reflejase qué usuario hizo qué operación sobre qué objeto y cuándo fue realizada.
- Recordar las últimas operaciones realizadas y mostrarlas en la ventana principal.
- Búsquedas avanzadas, filtrando por distintos campos.
- Integrar un instalador, al estilo de los CMS más comunes. En él se preguntará por los parámetros generales del alojamiento y el método de acceso a la base de datos. Después, generará los ficheros de configuración y dejaría funcionando la aplicación.

6. BIBLIOGRAFÍA

- [1] Carrola Rodrigues, Jorge (2012). *CRM for dummies*. Chichester, England: John Wiley & Sons Ltd. Free eBook recuperado en agosto, de 2014 de http://www.preact.co.uk/media/images/media/pdf/CRM_FOR_DUMMIES.pdf.
- [2] *PHP* (n.d.). Wikipedia: recuperado en agosto de 2014, de <http://es.wikipedia.org/wiki/PHP>.
- [3] Ribera, David. (Septiembre 2013). *¿Conoce la historia del CRM? Un CIO como tú en una Cloud como ésta*: recuperado en agosto, 2014 de <http://www.jmgris.com/index.php/conoce-la-historia-del-crm/>.
- [4] *Nuestros clientes* (n.d.). SugarCRM.com en Español: recuperado en agosto de 2014, de <http://www.sugarcrm.com/es>.
- [5] *SugarCRM Community Edition* (n.d.). SugarForge: Open Source CRM, SugarCRM: recuperado en agosto de 2014, de <http://www.sugarforge.org/content/open-source/>.
- [6] Skvorc, Bruno (diciembre, 2013). *Best PHP frameworks for 2014*. SitePoint.com: recuperado en agosto de 2014, de <http://www.sitepoint.com/best-php-frameworks-2014/>.
- [7] Smith, Grace (abril, 2013). *20 Exceptional CSS Boilerplates and Frameworks*. Marshable.com: recuperado en agosto de 2014, de <http://mashable.com/2013/04/26/css-boilerplates-frameworks/>.
- [8] Bennett, Simon; McRobb, Steve; Farmer, Ray (2007). *Análisis y diseño orientado a objetos de sistemas: usando UML*. 3ª edición. Madrid: Mac Graw-Hill Interamericana.
- [9] *Codeigniter* (n.d.). Codeigniter, a fully baked PHP framework: recuperado en agosto de 2014, de <https://ellislab.com/codeigniter>.
- [10] *Bootstrap* (n.d.) Bootstrap: recuperado en agosto de 2014, de <http://getbootstrap.com/>.
- [11] Cohn, Mike (2010). *Succeeding with agile: software development using Scrum*. New Jersey: Addison-Wesley. Disponible en Recursos Electrónicos de la Universidad de Jaén.

- [12] *Resolución de 8 de febrero de 2013, de la Dirección General de Empleo, por la que se registra y publica el IX Convenio colectivo de Telefónica Telecomunicaciones Públicas, SA.* (26 de febrero de 2013). Agencia Estatal Boletín Oficial del Estado: recuperado en agosto de 2014, de <https://www.boe.es/buscar/doc.php?id=BOE-A-2013-2153>.
- [13] Leffingwell, Dean (2007). *Scaling Software Agility: Best Practices for Large Enterprises*. Addison-Wesley Professional. Disponible en Recursos Electrónicos de la Universidad de Jaén.
- [14] Jacobson, Ibar; Spence, Ian; Bittner, Kurt (2013). *Casos de uso 2.0*. Ibar Jacobson International SA. Libro electrónico libre recuperado en agosto de 2014 de <http://bit.ly/1pXqemC>.
- [15] Fowler, Martin (2005). *The new methodology*. Martin Fowler website. Recuperado en agosto de 2014, de <http://www.martinfowler.com/articles/newMethodology.html>.
- [16] Ellis, Rick (2014). *Codeigniter User Guide Version 2.2.0*. Codeigniter / Elislab: recuperado en agosto de 2014, de <https://ellislab.com/codeigniter/user-guide/>.
- [17] Buenas prácticas y consejos en Codeigniter (marzo, 2013). Uno de piedra. Recuperado en agosto de 2014, de <http://uno-de-piera.com/buenas-practicas-y-consejos-en-codeigniter/>.
- [18] *DataMapper ORM* (n.d.). *DataMapper Codeigniter ORM Library*: recuperado en agosto de 2014, de <http://datamapper.wanwizard.eu/>.
- [80] Eonasdan. *Bootstrap datetimepicker*. Recuperado en agosto de 2014, de <http://eonasdan.github.io/bootstrap-datetimepicker/>.
- [90] Buschmann, Frank (1996). *Pattern-oriented software architecture. A pattern language for distributed computing. Vol 1 A system of patterns*. Chichester, England: John Wiley & Sons Ltd. Disponible en Recursos Electrónicos de la Universidad de Jaén.
- [100] Garzas, Javier (2012). *La agilidad está muriendo. Bienvenido el postagilismo*. www.javiergarzas.com, recuperado en agosto de 2014, de <http://www.javiergarzas.com/2012/05/postagilismo.html>.

Anexo I: Product Backlog

El *product backlog* en SCRUM es un documento de alto nivel del proyecto. En él se definen los requisitos del proyecto a través de *historias de usuario*. Estas historias son también valoradas en complejidad mediante *puntos de historia*; y ordenadas según el valor de negocio que éstas tienen.

Las historias de usuario han sido agrupadas en distintos *Temas* para identificar a qué aspecto de la aplicación se identifica cada uno.

1. INTERFAZ

ID	Historia de usuario	Criterio de aceptación	Puntos
INT1	Como usuario Quiero Tener una interfaz fácil de usar Para poder optimizar mi tiempo de uso de la herramienta		13.0
INT2	Como usuario Quiero tener una plantilla de correo electrónico Para poder apoyar el diseño de correos electrónicos	a) Debe estar escrita en HTML	2.0

2. MODELOS

ID	Historia de usuario	Criterio de aceptación	Puntos
MOD1	Como desarrollador Quiero tener tablas para gestionar contactos Para poder apoyar el desarrollo de la herramienta	a) Tabla para contactos b) Tabla para estado de contacto c) Tabla para correos	2.0

ID	Historia de usuario	Criterio de aceptación	Puntos
MOD2	Como desarrollador Quiero tablas para gestionar usuarios Para poder apoyar el desarrollo de la herramienta	a) Tabla para usuarios b) Tabla para estado de usuario	2.0
MOD3	Como desarrollador Quiero tablas para gestionar campañas Para poder apoyar el desarrollo de la herramienta	a) Tabla para campañas b) Tabla para tipo de campaña c) Tabla para estado de campaña	3.0
MOD4	Como desarrollador Quiero tablas para gestionar actividades Para poder apoyar el desarrollo de la herramienta	a) Tabla para actividades b) Tabla para tipo de actividad c) Tabla para prioridad de actividades d) Tabla para estados de actividades	3.0
MOD5	Como desarrollador Quiero tablas para gestionar alertas Para poder apoyar el desarrollo de la herramienta	a) Tabla para alertas	2.0
MOD6	Como desarrollador Quiero tablas para gestionar perfiles de usuario Para poder apoyar el desarrollo de la herramienta	a) Tabla para perfiles b) Completar registros con los perfiles	3.0
MOD7	Como desarrollador Quiero tablas para gestionar configuraciones de usuario Para poder guardar las configuraciones de cada usuario	a) Tabla para configuraciones	2.0

ID	Historia de usuario	Criterio de aceptación	Puntos
MOD8	Como desarrollador Quiero tablas para gestionar tickets de soporte Para poder apoyar el desarrollo de la herramienta	a) Tabla para soporte b) Tabla para actuaciones de soporte	3.0

3. CONTACTOS

ID	Historia de usuario	Criterio de aceptación	Puntos
CNT1	Como usuario Quiero crear nuevos contactos Para poder almacenar sus datos en el sistema	a) No puede dejarse en blanco los campos obligatorios b) Un contacto puede tener varios correos electrónicos c) El NIF debe ser único, o bien, quedar en blanco	8.0
CNT2	Como usuario Quiero listar todos los contactos Para poder ver de una manera rápida los datos básicos de todos los contactos	a) Los resultados deben aparecer paginados	5.0
CNT3	Como usuario Quiero visualizar la ficha de un contacto Para poder ver los detalles del contacto seleccionado		5.0

ID	Historia de usuario	Criterio de aceptación	Puntos
CNT4	Como usuario Quiero editar cualquier contacto Para poder corregir o ampliar datos del contacto	a) La interfaz debe ser similar a la de nuevo contacto b) No puede dejarse en blanco los campos obligatorios c) Un contacto puede tener varios correos electrónicos d) El NIF debe ser único, o bien, quedar en blanco	8.0
CNT5	Como usuario Quiero eliminar cualquier contacto Para poder eliminar del sistema cualquier dato del contacto	a) Los datos asociados deben guardarse asociados a un usuario genérico para no perder el registro	2.0
CNT6	Como usuario Quiero controlar el estado de los contactos Para poder diferenciar contactos según estado	a) Los distintos estados son: "No especificado", "Potencial", "Activo" o "Agotado"	3.0
CNT7	Como usuario Quiero buscar contactos Para poder filtrar contactos del listado	a) Usar un campo de texto b) La cadena debe buscarse en todos los campos relevantes del contacto (y relacionados)	3.0
CNT8	Como usuario Quiero ordenar los listados por columnas Para poder ver el listado ordenado por según qué columna	a) Orden ascendente o descendente b) Activar en el título de la columna c) Debe mantener la página	3.0

4. USUARIOS

ID	Historia de usuario	Criterio de aceptación	Puntos
USU1	Como usuario Quiero crear nuevos usuarios Para poder tener un usuario para cada empleado	a) No puede dejarse en blanco los campos obligatorios b) El NIF debe ser único, o bien, estar vacío c) La contraseña debe generarse aleatoriamente, no se puede insertar a mano d) La contraseña generada debe mandarse por correo a la dirección indicada en el formulario	8.0
USU2	Como usuario Quiero listar todos los usuarios Para poder ver de una manera rápida los datos básicos de todos los usuarios	a) Los resultados deben aparecer paginados	5.0
USU3	Como usuario Quiero visualizar la ficha de un usuario Para poder ver todos los detalles del usuario seleccionado		5.0
USU4	Como usuario Quiero editar cualquier usuario Para poder corregir o ampliar datos	a) La interfaz debe ser similar a la de nuevo usuario b) No puede dejarse en blanco los campos obligatorios c) El NIF debe ser único, o bien, estar vacío	8.0

ID	Historia de usuario	Criterio de aceptación	Puntos
USU5	Como usuario Quiero eliminar cualquier usuario Para poder eliminar los datos del usuario que se trate	a) Los datos asociados se eliminarán también del sistema	2.0
USU6	Como usuario Quiero asignar un perfil a un usuario Para poder filtrar los accesos que tiene cada usuario	a) El usuario "admin" no puede cambiarse de perfil	2.0
USU7	Como usuario Quiero asignar un estado a los usuarios Para poder controlar el acceso a los usuarios al sistema sin necesidad de eliminarlos	a) Los distintos estados son "activo", "excedencia" y "despedido"	2.0
USU8	Como usuario Quiero resetear la contraseña de otro usuario Para poder enviarle una nueva en caso de pérdida	a) No se puede insertar la contraseña: debe generarse aleatoriamente b) La nueva contraseña debe mandarse por correo a la dirección del usuario	3.0
USU9	Como usuario Quiero cambiar la imagen de un usuario Para poder identificar rápidamente un perfil por su avatar	a) Debe comprobarse que sea una imagen b) Debe guardarse en el servidor	3.0
USU10	Como usuario Quiero buscar un usuario Para poder filtrar el listado	a) Usar un campo de texto b) La cadena debe buscarse en todos los campos relevantes y relacionados	3.0

ID	Historia de usuario	Criterio de aceptación	Puntos
USU11	Como usuario Quiero ordenar los listados por columnas Para poder ver el listado ordenador según qué columnas	a) Orden ascendente o descendente b) Activar en el título de la columna c) Mantener la página	3.0

5. CAMPAÑAS

ID	Historia de usuario	Criterio de aceptación	Puntos
CAM1	Como usuario Quiero crear nuevas campañas Para poder gestionar campañas de la organización	a) No pueden dejarse en blanco los campos obligatorios	8.0
CAM2	Como usuario Quiero listar todas las campañas Para poder ver de una manera rápida los datos básicos de todas las campañas	a) Los resultados deben aparecer paginados	5.0
CAM3	Como usuario Quiero listar todas las campañas propias Para poder ver de una manera rápida los datos básicos de las campañas propias	a) Los resultados deben aparecer paginados	2.0
CAM4	Como usuario Quiero visualizar la ficha de una campaña Para poder ver todos los detalles de la campaña seleccionada		5.0

ID	Historia de usuario	Criterio de aceptación	Puntos
CAM5	Como usuario Quiero editar campañas Para poder corregir o ampliar datos de la campaña	a) La interfaz debe ser similar a la de crear nueva campaña b) No puede dejarse en blanco los campos obligatorios	5.0
CAM6	Como usuario Quiero eliminar campañas Para poder eliminar cualquier dato del sistema relacionado con la campaña	a) Las actividades asociadas no deben eliminarse (sólo se eliminará la relación)	2.0
CAM7	Como usuario Quiero controlar el tipo de campaña Para poder diferenciar las campañas según el tipo	a) Los posibles tipos son "Ventas", "Correo", "Email", "Web", "Radio", "Televisión", "Boletín de noticias" b) El control del tipo se realiza en la creación y en la edición de la campaña	3.0
CAM8	Como usuario Quiero controlar el estado de campaña Para poder identificar las campañas según el estado de desarrollo de la campaña	a) Los posibles estados son "En Planificación", "Activa", "Inactiva", "Completada", "Suspendida", "Aplazada" b) El control del estado se realiza en la creación y edición de la campaña	3.0
CAM9	Como usuario Quiero buscar campañas Para poder filtrar las campañas del listado	a) Usar un campo de texto b) La cadena debe buscarse en todos los campos relevantes y relacionados	3.0
CAM10	Como usuario Quiero ordenar los listados por columnas Para poder ver el listado ordenado según qué columnas	a) Orden ascendente o descendente b) Activar con el título de la columna c) Mantener la paginación	3.0

ID	Historia de usuario	Criterio de aceptación	Puntos
CAM11	Como usuario Quiero ver las campañas asignadas a un usuario en la ficha de visualización del usuario Para poder ver los datos más relevantes de un usuario en su ficha de visualización	a) Debe mostrarse sólo las campañas asignadas al usuario	2.0

6. ACTIVIDADES

ID	Historia de usuario	Criterio de aceptación	Puntos
ACT1	Como usuario Quiero crear actividades propias Para poder gestionar las actividades	a) No puede dejarse en blanco los campos obligatorios	8.0
ACT2	Como usuario Quiero crear actividades para cualquier usuario Para poder gestionar las actividades	a) No puede dejarse en blanco los campos obligatorios	2.0
ACT3	Como usuario Quiero listar actividades propias Para poder ver los datos básicos de las actividades	a) Los resultados deben aparecer paginados	5.0
ACT4	Como usuario Quiero listar todas las actividades Para poder ver los datos básicos de las actividades	a) Los resultados deben aparecer paginados	2.0

ID	Historia de usuario	Criterio de aceptación	Puntos
ACT5	Como usuario Quiero visualizar la ficha de cualquier actividad Para poder ver todos los datos de la actividad seleccionada		5.0
ACT6	Como usuario Quiero visualizar la ficha de una actividad propia Para poder ver todos los datos de la actividad seleccionada		2.0
ACT7	Como usuario Quiero editar cualquier actividad Para poder corregir o ampliar datos de la actividad	a) La interfaz debe ser similar a la de crear actividad b) No puede dejarse en blanco los campos obligatorios	3.0
ACT8	Como usuario Quiero editar una actividad propia Para poder corregir o ampliar datos de la actividad	a) La interfaz debe ser similar a la de crear actividad b) No puede dejarse en blanco los campos obligatorios	1.0
ACT9	Como usuario Quiero eliminar cualquier actividad Para poder eliminar datos del sistema		2.0
ACT10	Como usuario Quiero eliminar una actividad propia Para poder eliminar datos del sistema		1.0

ID	Historia de usuario	Criterio de aceptación	Puntos
ACT11	Como usuario Quiero gestionar el tipo de actividad Para poder diferenciar actividades por tipo	a) Es un campo obligatorio	2.0
ACT12	Como usuario Quiero gestionar la prioridad de la actividad Para poder diferenciar actividades por prioridad	a) Es un campo obligatorio	2.0
ACT13	Como usuario Quiero gestionar el estado de la actividad Para poder diferenciar actividades según estado	a) Es un campo obligatorio	2.0
ACT14	Como usuario Quiero gestionar la campaña asociada a la actividad Para poder clasificar actividades por campaña	a) Es un campo opcional b) Incompatible si está asociado a un ticket de soporte	3.0
ACT15	Como usuario Quiero gestionar el contacto asociado a la actividad Para poder identificar el contacto objeto de la actividad	a) Es un campo obligatorio	3.0
ACT16	Como usuario Quiero gestionar el usuario relacionado con la actividad Para poder identificar qué usuario realizó dicha actividad	a) Es un campo obligatorio	3.0

ID	Historia de usuario	Criterio de aceptación	Puntos
ACT17	Como usuario Quiero gestionar el ticket de soporte asociado a la actividad Para poder clasificar actividades por ticket	a) Es un campo opcional b) Incompatible si está asociado a una campaña	3.0
ACT18	Como usuario Quiero buscar actividades Para poder filtrar las actividades del listado	a) Usar campo de texto b) La cadena debe buscarse en todos los campos relevantes y relacionados	3.0
ACT19	Como usuario Quiero ordenar los listados por columnas Para poder ver el listado ordenado según qué columnas	a) Orden ascendente o descendente b) Activar con el título de la columna c) Mantener la paginación	3.0
ACT20	Como usuario Quiero ver las actividades de un contacto en su ficha de visualización Para poder concentrar en una página los datos más relevantes de un contacto	a) Mostrar sólo las actividades del contacto en cuestión	2.0
ACT21	Como usuario Quiero ver las actividades de una campaña en la ficha de visualización de la campaña Para poder concentrar en una página los datos más relevantes de la campaña	a) Mostrar sólo las actividades de la campaña en cuestión	2.0

ID	Historia de usuario	Criterio de aceptación	Puntos
ACT22	Como usuario Quiero ver las actividades de un usuario en la ficha de visualización del usuario Para poder concentrar en una página los datos más relevantes del usuario	a) Mostrar sólo las actividades del usuario en cuestión	2.0

7. ALERTAS

ID	Historia de usuario	Criterio de aceptación	Puntos
ALE1	Como usuario Quiero crear nuevas alertas Para poder recibir avisos	a) El usuario relacionado debe ser el mismo usuario que la cree b) No puede dejarse en blanco los campos obligatorios	8.0
ALE2	Como usuario Quiero listar las alertas propias Para poder ver de una manera rápida un listado de las alertas propias	a) Debe ordenarse por fecha, de más reciente a más antiguo b) Visualmente debe identificarse las que han expirado de las que serán en un futuro c) Los resultados deben aparecer paginados	5.0
ALE3	Como usuario Quiero visualizar la ficha de una alerta Para poder ver todos los detalles de la alerta seleccionada		5.0

ID	Historia de usuario	Criterio de aceptación	Puntos
ALE4	Como usuario Quiero editar una alerta propia Para poder corregir o ampliar datos de una alerta propia	a) La interfaz debe ser similar a la de crear alerta	5.0
ALE5	Como usuario Quiero eliminar una alerta propia Para poder evitar que aparezca la alerta programada		2.0
ALE6	Como usuario Quiero asociar una actividad a la alerta Para poder mantener un vínculo entre la alerta y una actividad (si la hubiese)	a) Es un campo opcional b) Debe ser una actividad propia	3.0
ALE7	Como usuario Quiero ver en una ventana modal una alerta no visualizada Para poder ver rápidamente un resumen de la alerta	a) Debe aparecer una alerta visual en el menú principal b) Debe tener un botón para marcar la alerta como visualizada c) Debe tener un enlace a la visualización de la alerta d) Debe tener la opción de aplazar alerta	5.0
ALE8	Como usuario Quiero tener la opción de que se dispare automáticamente la ventana modal de una alerta que haya expirado Para poder ver la alerta mientras estoy trabajando con el sistema	a) La alerta debe estar marcada como no visualizada y la fecha y hora haber expirado	5.0

ID	Historia de usuario	Criterio de aceptación	Puntos
ALE9	Como usuario Quiero poder recibir un correo electrónico con la alerta Para poder recibir en cualquier lugar la alerta programada	a) Debe tener un atributo para saber si hay que enviar o no el mail b) Debe tener un atributo para saber si se ha enviado o no el correo	5.0
ALE10	Como usuario Quiero buscar alertas Para poder filtrar alertas del listado	a) Usar un campo de texto b) La cadena debe buscarse en todos los campos relevantes y relacionados	3.0
ALE11	Como usuario Quiero ordenar los listados por columnas Para poder ver el listado ordenado según qué columnas	a) Orden ascendente o descendente b) Activar con el título de la columna c) Mantener la paginación	3.0
ALE12	Como usuario Quiero ver las alertas de una actividad en la ficha de visualización de la actividad Para poder concentrar en la visualización los datos más relevantes de una actividad	a) Debe mostrar únicamente las alertas de la actividad en cuestión	2.0

8. PERMISOS

ID	Historia de usuario	Criterio de aceptación	Puntos
PER1	Como usuario Quiero listar los perfiles de usuario Para poder ver los datos básicos de todos los perfiles	a) Los resultados deben aparecer paginados	5.0

ID	Historia de usuario	Criterio de aceptación	Puntos
PER2	Como usuario Quiero visualizar la ficha de un perfil Para poder ver la política que tiene el perfil	a) Debe verse por grupos de directivas b) Debe verse los usuarios que tienen asignado el perfil	5.0
PER3	Como usuario Quiero visualizar la ficha de mi perfil Para poder ver la política que tiene mi perfil		2.0
PER4	Como usuario Quiero buscar perfiles Para poder filtrar los perfiles del listado	a) Usar campo de texto b) La cadena debe buscarse en todos los campos relevantes y relacionados	3.0
PER5	Como usuario Quiero ordenar los listados por columnas Para poder ver el listado ordenado según qué columnas	a) Orden ascendente o descendente b) Activar con el título de la columna c) Mantener la paginación	3.0

9. SOPORTE

ID	Historia de usuario	Criterio de aceptación	Puntos
SOP1	Como usuario Quiero crear tickets de soporte para cualquier usuario Para poder gestionar incidencias con los contactos	a) Los campos obligatorios no pueden dejarse en blanco b) Debe estar asociado a un contacto c) Al guardarse debe mandarse un correo electrónico al contacto	8.0

ID	Historia de usuario	Criterio de aceptación	Puntos
SOP2	Como usuario Quiero crear tickets de soporte propios Para poder gestionar incidencias con los contactos	a) Los campos obligatorios no pueden dejarse en blanco b) Debe estar asociado a un contacto c) Al guardarse debe enviarse un correo electrónico al contacto	2.0
SOP3	Como usuario Quiero listar todos los tickets de soporte Para poder ver de una manera rápida los datos básicos de los tickets	a) Los resultados deben aparecer paginados	5.0
SOP4	Como usuario Quiero listar tickets de usuario propios Para poder ver de una manera rápida los datos básicos de los tickets	a) Los resultados deben aparecer paginados	2.0
SOP5	Como usuario Quiero visualizar todos los tickets de soporte Para poder ver todos los detalles de un ticket		5.0
SOP6	Como usuario Quiero visualizar los tickets de soporte propios Para poder ver todos los detalles de un ticket		2.0

ID	Historia de usuario	Criterio de aceptación	Puntos
SOP7	Como usuario Quiero editar cualquier ticket de soporte Para poder corregir o ampliar datos	a) Los campos obligatorios no pueden dejarse en blanco b) Debe estar asociado a un contacto c) Al terminar debe enviarse un correo electrónico al contacto	5.0
SOP8	Como usuario Quiero editar tickets de soporte propios Para poder corregir o ampliar datos	a) Los campos obligatorios no pueden dejarse en blanco b) Debe estar asociado a un contacto c) Al terminar debe enviarse por correo electrónico al contacto	2.0
SOP9	Como usuario Quiero eliminar cualquier ticket de soporte Para poder borrar datos del sistema		2.0
SOP10	Como usuario Quiero eliminar un ticket de soporte propio Para poder borrar datos del sistema		1.0
SOP11	Como usuario Quiero buscar tickets de soporte Para poder filtrar tickets del listado	a) Usar campo de texto b) La cadena debe buscarse en todos los campos relevantes y relacionados	3.0

ID	Historia de usuario	Criterio de aceptación	Puntos
SOP12	Como usuario Quiero ordenar los listados por columnas Para poder ver el listado ordenado según qué columnas	a) Orden ascendente o descendente b) Activar con el título de la columna c) Mantener la paginación	3.0
SOP13	Como usuario Quiero ver los tickets de soporte de un contacto en la ficha de visualización del contacto Para poder concentrar en la visualización los datos más relevantes del contacto	a) Mostrar únicamente los tickets del contacto en cuestión	2.0
SOP14	Como usuario Quiero ver los tickets de soporte de un usuario en la ficha de visualización del usuario Para poder concentrar en la visualización los datos más relevantes del usuario	a) Mostrar únicamente los tickets del usuario en cuestión	2.0

Anexo II: Manual de instalación del sistema

Para la instalación del CRM en un hosting debemos asegurarnos que el servidor cumple con los siguientes requisitos:

- Servidor Apache
- PHP 5.1.6 o superior
- Base de datos MySQL (4.1+), MySQLi, MS SQL, Postgres, Oracle, SQLite o ODBC.

Para su instalación hay que tomar la carpeta `crmStar` y pegarla en la carpeta pública. El contenido de la carpeta puede ubicarse en la raíz del servidor, o bien, dejar los ficheros en una carpeta.

A continuación, modificar los siguientes ficheros:

- ***application/config/config.php***

Buscar la siguiente línea:

```
$config['base_url'] = 'http://localhost/crm/';
```

Y escribir la dirección absoluta a la raíz de la aplicación.

- ***application/config/database.php***

Buscar las siguientes líneas y configurar los datos de conexión con la base de datos:

```
$db['default']['hostname'] = 'localhost';  
$db['default']['username'] = 'user';  
$db['default']['password'] = 'password';  
$db['default']['database'] = 'crm_tfg';  
$db['default']['dbdriver'] = 'mysql';
```

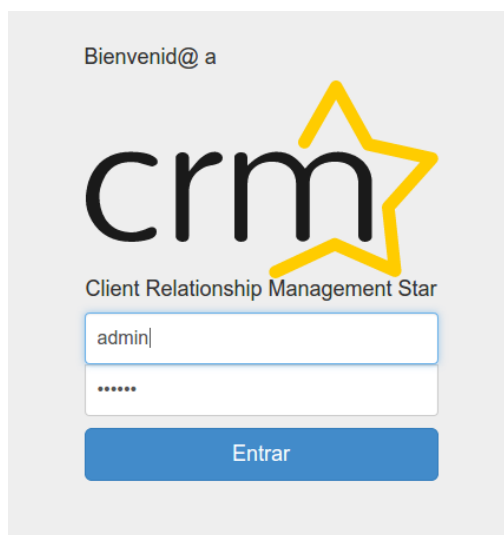
Por último, en la base de datos importar el fichero situado en la siguiente ruta:

```
/sql/crm_tfg_inicializacion.sql
```

Anexo III: Manual de usuario

1. ACCESO A LA PLATAFORMA Y VISIÓN GENERAL

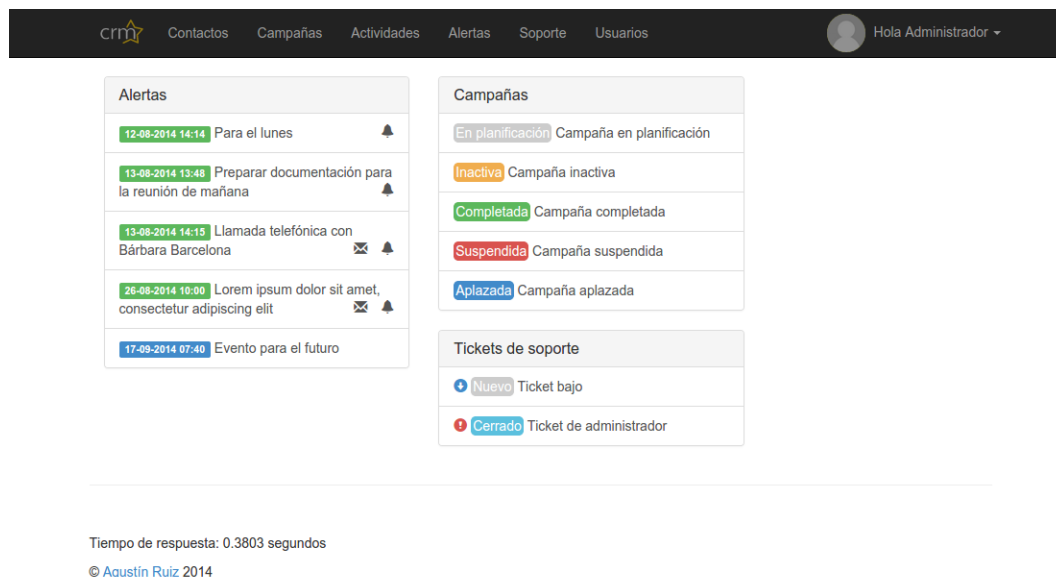
Para acceder a la plataforma CRM introducimos la dirección URL en cualquier navegador web. A continuación, aparecerá una ventana similar a la siguiente donde deberemos introducir nuestro nombre de usuario y contraseña:



Formulario de acceso a la plataforma CRM. El formulario está centrado en una pantalla gris. En la parte superior, dice "Bienvenid@ a". Debajo, hay un logo con el texto "crm" en negro y una estrella amarilla a su derecha. Debajo del logo, dice "Client Relationship Management Star". Hay dos campos de entrada: el primero contiene "admin|" y el segundo contiene "*****". Debajo de los campos, hay un botón azul con el texto "Entrar".

Figura 1: Formulario de acceso

La pantalla de usuario principal tiene un aspecto similar a la siguiente:



Pantalla de usuario principal. La barra superior es negra y contiene el logo CRM, los menús: Contactos, Campañas, Actividades, Alertas, Soporte, Usuarios, y un usuario logueado "Hola Administrador". El contenido principal está dividido en tres columnas. La columna izquierda muestra "Alertas" con una lista de eventos. La columna central muestra "Campañas" con una lista de campañas en diferentes estados. La columna derecha muestra "Tickets de soporte" con una lista de tickets. En la parte inferior, se muestra el tiempo de respuesta y el copyright.

Alertas	
12-08-2014 14:14	Para el lunes
13-08-2014 13:48	Preparar documentación para la reunión de mañana
13-08-2014 14:15	Llamada telefónica con Bárbara Barcelona
26-08-2014 10:00	Lorem ipsum dolor sit amet, consectetur adipiscing elit
17-09-2014 07:40	Evento para el futuro

Campañas	
En planificación	Campaña en planificación
Inactiva	Campaña inactiva
Completada	Campaña completada
Suspendida	Campaña suspendida
Aplazada	Campaña aplazada

Tickets de soporte	
Nuevo	Ticket bajo
Cerrado	Ticket de administrador

Tiempo de respuesta: 0.3803 segundos
© Agustín Ruiz 2014

Figura 2: Visión general

En la parte superior podemos encontrar el menú principal.

En la misma barra a la derecha veremos el menú del usuario actual, donde podremos ver los permisos asignados, cambiar nuestra contraseña o cerrar la sesión cuando hayamos terminado de usar la herramienta.

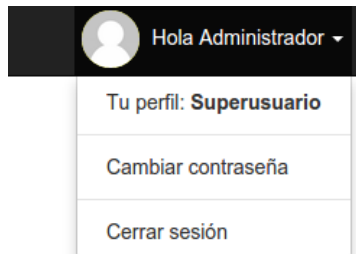


Figura 3: Menú de usuario

En la zona centrar encontraremos las operaciones que tenemos pendientes, tales como alertas, campañas asignadas y tickets de soporte.

2. GESTIÓN DE CONTACTOS

Los contactos son los registros de personas físicas o jurídicas con las que tenemos comunicaciones frecuentes.

Para el acceso al módulo de contactos entramos en el apartado *Contactos* del menú superior. Por defecto, encontraremos un listado con todos los contactos.

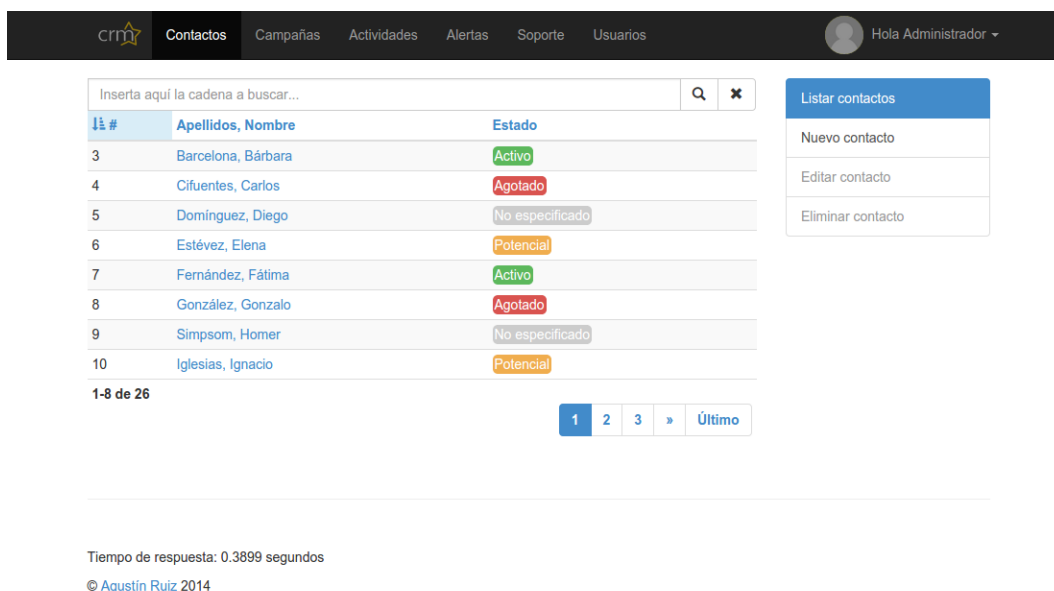


Figura 4: Listar contactos

En la parte superior tenemos un campo de búsqueda genérico.

Haciendo clic sobre el encabezado de las columnas podemos elegir por qué criterio se ordenan los resultados. Para cambiar el orden de ascendente a descendente (y viceversa), deberemos volver a hacer clic en el encabezado.

2.1. Crear un nuevo contacto

Hacemos clic sobre la entrada *Nuevo contacto* del menú lateral.

CRM **Contactos** Campañas Actividades Alertas Soporte Usuarios Hola Administrador ▾

Datos generales

Nombre*

Apellidos*

NIF

Estado

Dirección

Ciudad

Provincia

Cód. Postal

País

Telf. Oficina

Telf. Móvil

FAX

Correo electrónico ☒ Principal ☐ No válido

Otros datos

Los campos marcados con un asterisco (*) son obligatorios

Timepo de respuesta: 0.1790 segundos
© Agustín Ruiz 2014

Figura 5: Formulario de nuevo contacto

Los campos marcados con un asterisco son obligados. Los campos del formulario a cumplimentar son los siguientes:

- **Nombre:** Nombre del contacto. Es un campo obligatorio.

- **Apellidos:** Apellidos del contacto.
- **NIF:** NIF de la persona o empresa.
- **Estado:** Estado del contacto. Los valores posibles son *No especificado*, *Potencial*, *Activo* o *Agotado*.
- **Dirección:** Dirección del contacto.
- **Ciudad:** Ciudad del contacto.
- **Provincia:** Provincia del contacto.
- **Cód. Postal:** Código postal del contacto.
- **País:** País del contacto.
- **Telf. Oficina:** Teléfono de oficina del contacto.
- **Telf. Móvil:** Teléfono móvil del contacto.
- **Correo electrónico:** Correo electrónico del contacto. Puede introducirse más de un correo. Al menos, uno de ellos debe marcarse como correo principal. Adicionalmente, se puede marcar un correo como no válido.
- **Otros datos:** Datos adicionales acerca del contacto.

Para finalizar, pulsaremos el botón *Crear contacto* para guardar el nuevo registro o *Cancelar* para salir.

2.2. Seleccionar contacto

Para seleccionar un contacto haremos clic sobre su nombre desde el listado de contactos.

Agustín Ruiz Linares

Datos generales

Nombre	Agustín	Apellidos	Ruiz Linares
NIF	15510111S	Estado	Potencial

Dirección

Dirección	C/ Navas, 21	Cód. Postal	23300
Ciudad	Villacarrillo	Provincia	Jaén

Métodos de contacto

Correo	agustinruizlinares@gmail.com (principal)	Telf. Oficina	953442061
Telf. Móvil	600798782	FAX	953442053

Otros

Otros datos	Editado
	Bien

Actividades

#	Prioridad/Asunto	Campaña(C)/Ticket(T)	Inicio	Tipo	Estado	Usuario
36	Actividad de Tres		15-08-2014 00:23	Llamada telefónica	No iniciada	Tres Tres
34	Aún estás a tiempo		08-08-2014 13:56	Llamada telefónica	No iniciada	Administrador
31	Actividad de Agustín		08-08-2014 00:25	Correo electrónico	No iniciada	Agustín Ruiz Linares
33	Otra más	(C) Campaña activa	14-05-2014 13:32	Carta	No iniciada	Administrador

4 actividades

Tickets de soporte

#	Prioridad/Asunto	Estado	Usuario
No hay tickets			

Tiempo de respuesta: 0.0972 segundos

© Agustín Ruiz 2014

Figura 6: Visualización de contactos

En la pantalla veremos todos los datos del contacto, las actividades realizadas con él y los tickets de soporte que tiene asociado. Podremos acceder a dichos registros pulsando directamente sobre el nombre o el asunto.

En el menú derecho tendremos dos botones nuevos: *Nueva actividad para este contacto* y *Nuevo ticket para este contacto*.

En ésta ventana se habilitan los botones *Editar contacto* y *Eliminar contacto*. En el primer caso se verá un formulario similar al de creación de un nuevo contacto para la edición del mismo.

En el caso de eliminar el contacto se mostrará un mensaje modal para confirmar si se desea eliminar el contacto definitivamente.

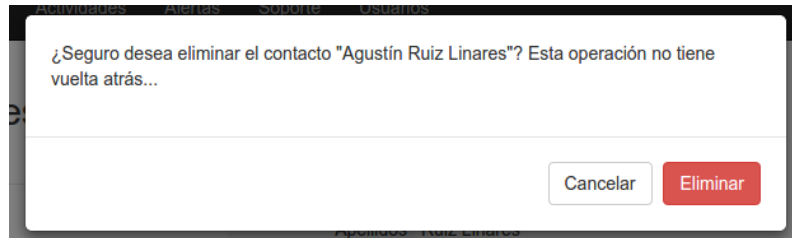


Figura 7: Modal para eliminar un contacto

3. GESTIÓN DE CAMPAÑAS

Las campañas son los distintos proyectos que gestiona la empresa y de los que se pretende registrar las comunicaciones generadas.

Para entrar en éste módulo seleccionaremos *Campañas* del menú superior.

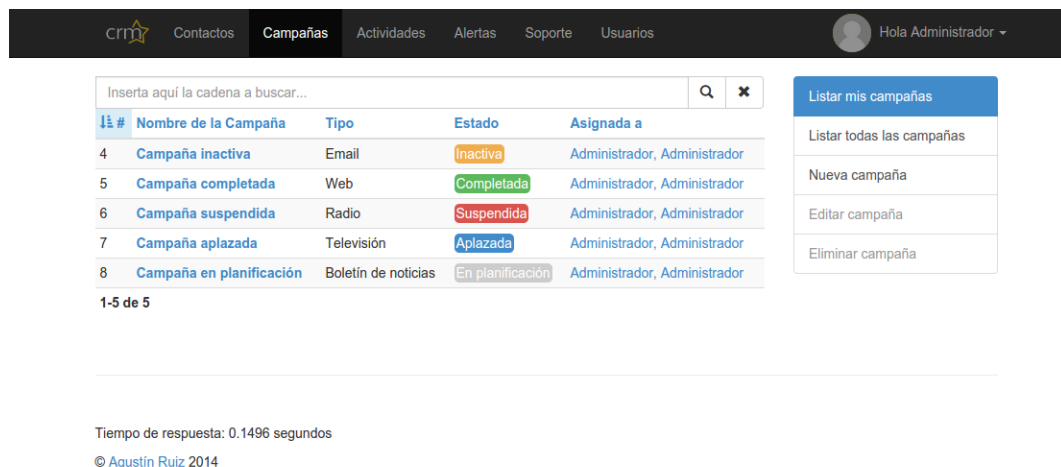


Figura 8: Listar campañas

Por defecto, se muestra un listado con las campañas propias del usuario. En el menú de la derecha se puede seleccionar la opción *Listar todas las campañas* para ver las asignadas a otros usuarios.

En la parte superior tenemos un campo de búsqueda genérico.

Haciendo clic sobre el encabezado de las columnas podemos elegir por qué criterio se ordenan los resultados. Para cambiar el orden de ascendente a descendente (y viceversa), deberemos volver a hacer clic en el encabezado.

3.1. Crear campañas

Hacemos clic sobre la entrada *Nueva campaña* del menú lateral.

The screenshot shows a web application interface for creating a new campaign. The top navigation bar includes links for 'CRM', 'Contactos', 'Campañas' (active), 'Actividades', 'Alertas', 'Soporte', and 'Usuarios'. A user profile 'Hola Administrador' is visible on the right. The main form for 'Nueva campaña' contains the following fields:

- Nombre** (required, marked with a red asterisk): Text input with placeholder 'Nombre de la campaña (OBLIGATORIO)'.
- Inicio**: Date picker.
- Finalización**: Date picker.
- Tipo campaña**: Dropdown menu with 'Ventas' selected.
- Estado**: Dropdown menu with 'En planificación' selected.
- Descripción**: Text area with placeholder 'Descripción de la campaña'.
- Objetivo**: Text area with placeholder 'Objetivo de la campaña'.
- Usuario responsable**: Searchable dropdown with a magnifying glass icon.

At the bottom of the form are two buttons: 'Crear campaña' (green) and 'Cancelar' (red). Below the buttons is a note: 'Los campos marcados con un asterisco (*) son obligatorios'. On the right side, a sidebar menu lists actions: 'Listar mis campañas', 'Listar todas las campañas', 'Nueva campaña' (highlighted in blue), 'Editar campaña', and 'Eliminar campaña'. At the very bottom, the page shows 'Tiempo de respuesta: 0.1287 segundos' and '© Agustín Ruiz 2014'.

Figura 9: Formulario nueva campaña

Los campos marcados con un asterisco son obligados. Los campos del formulario a cumplimentar son los siguientes:

- **Nombre**: Nombre de la campaña. Es un campo obligatorio.
- **Inicio**: Fecha de inicio de la campaña.
- **Finalización**: Fecha de finalización de la campaña.
- **Tipo de campaña**: Los posibles valores son *Ventas*, *Correo*, *Email*, *Web*, *Radio*, *Televisión* o *Boletín de noticias*.
- **Estado**: Estado de la campaña. Los posibles valores son *En planificación*, *Activa*, *Inactiva*, *Completada*, *Suspendida* o *Aplazada*.
- **Descripción**: Descripción de la campaña.
- **Objetivo**: Resultado de la campaña.

- **Usuario responsable:** Usuario asignado como responsable de la campaña. Se abrirá un formulario de búsqueda de que poder buscar y seleccionar el usuario responsable.

Para finalizar, pulsaremos el botón *Crear campaña* para guardar el nuevo registro o *Cancelar* para salir.

3.2. Seleccionar campaña

Para seleccionar una campaña haremos clic sobre su nombre desde el listado de campañas.

Campaña completada

Datos generales

Campaña	Campaña completada				
Inicio	-	Finalización	-		
Tipo	Web	Estado	Completada		
Descripción					
Objetivo					

Asignación

Asignada a [Administrador Administrador](#) (Total campañas asignadas: 5)

Actividades

#	Prioridad/Asunto	Inicio	Tipo	Prioridad/Estado	Contacto	Usuario
No hay actividades						

Tiempo de respuesta: 0.0581 segundos
© Agustín Ruiz 2014

[Listar mis campañas](#)
[Listar todas las campañas](#)
[Nueva campaña](#)
[Editar campaña](#)
[Eliminar campaña](#)
[Nueva actividad para esta campaña](#)

Figura 10: Vista de campaña

En la pantalla veremos todos los datos de la campaña, el usuario al que ha sido asignada y las actividades. Podremos acceder a dichos registros pulsando directamente sobre el nombre o el asunto.

En el menú derecho tendremos un botón nuevo: *Nueva actividad para esta campaña*.

En ésta ventana se habilitan los botones *Editar campaña* y *Eliminar campaña*. En el primer caso se verá un formulario similar al de creación de una nueva campaña para la edición de la misma.

En el caso de eliminar la campaña se mostrará un mensaje modal para confirmar si se desea eliminar la campaña definitivamente.

4. GESTIÓN DE TICKETS DE SOPORTE

Los tickets de soporte sirven para gestionar los problemas e incidencias relacionados con los contactos.

Para el acceso al módulo de soporte entramos en el apartado *Soporte* del menú superior. Por defecto, encontraremos un listado con todos los tickets de soporte.

The screenshot shows the 'Soporte' (Support) module in a CRM application. At the top is a navigation bar with tabs: 'Contactos', 'Campañas', 'Actividades', 'Alertas', 'Soporte' (selected), and 'Usuarios'. A user profile 'Hola Administrador' is on the right. Below the navigation bar is a search bar with the placeholder 'Inserta aquí la cadena a buscar...'. The main area displays a table of tickets with columns: 'Id #', 'Asunto', 'Estado', 'Contacto', and 'Usuario'. Two tickets are listed: 'Ticket bajo' (Nuevo) and 'Ticket de administrador' (Cerrado). A sidebar on the right contains a 'Listar mis tickets' button and a list of actions: 'Listar todos los tickets', 'Nuevo ticket', 'Editar ticket', and 'Eliminar ticket'. At the bottom, it shows 'Tiempo de respuesta: 0.1435 segundos' and '© Agustín Ruiz 2014'.

Id #	Asunto	Estado	Contacto	Usuario
12	Ticket bajo	Nuevo	Dominguez, Diego	Administrador, Administrador
14	Ticket de administrador	Cerrado	Barcelona, Bárbara	Administrador, Administrador

Figura 11: Listar tickets de soporte

Por defecto, se muestra un listado con los tickets de soporte asignados al usuario. En el menú de la derecha se puede seleccionar la opción *Listar todos los tickets* para ver los asignados a otros usuarios.

En la parte superior tenemos un campo de búsqueda genérico.

Haciendo clic sobre el encabezado de las columnas podemos elegir por qué criterio se ordenan los resultados. Para cambiar el orden de ascendente a descendente (y viceversa), deberemos volver a hacer clic en el encabezado.

4.1. Crear un nuevo ticket de soporte

Hacemos clic sobre la entrada *Nuevo ticket* del menú lateral. También se puede crear un ticket desde la ventana de visualización de un contacto.

The screenshot shows a web application interface for creating a new support ticket. At the top is a navigation bar with tabs: 'CRM', 'Contactos', 'Campañas', 'Actividades', 'Alertas', 'Soporte' (selected), and 'Usuarios'. On the right of the navigation bar is a user profile icon and the text 'Hola Administrador'. Below the navigation bar is a sidebar menu with the following items: 'Listar mis tickets', 'Listar todos los tickets', 'Nuevo ticket' (highlighted in blue), 'Editar ticket', and 'Eliminar ticket'. The main content area contains a form for creating a new ticket. The form has the following fields: 'Asunto*' (text input, placeholder: 'Asunto del ticket (OBLIGATORIO)'), 'Descripción*' (text area, placeholder: 'Descripción de la incidencia'), 'Prioridad*' (dropdown menu, value: 'Media'), 'Estado*' (dropdown menu, value: 'Nuevo'), 'Contacto*' (text input with search and clear icons), 'Usuario*' (text input with search and clear icons, value: 'Administrador Administrador'), and 'Resolución' (text area, placeholder: 'Resultado de la incidencia'). At the bottom of the form are two buttons: 'Crear ticket' (green) and 'Cancelar' (red). Below the buttons is a note: 'Los campos marcados con un asterisco (*) son obligatorios'. At the very bottom of the page, there is a footer with the text: 'Tiempo de respuesta: 0.1362 segundos' and '© Agustín Ruiz 2014'.

Figura 12: Formulario de nuevo ticket de soporte

Los campos marcados con un asterisco son obligados. Los campos del formulario a cumplimentar son los siguientes:

- **Asunto:** Asunto del ticket.
- **Descripción:** Descripción de la incidencia.
- **Prioridad:** Prioridad del ticket. Los posibles valores son *Alta*, *Media* y *Baja*.
- **Estado:** Estado del ticket. Los posibles valores son *Nuevo*, *Asignado*, *Cerrado*, *Pendiente de información* o *Rechazado*.
- **Contacto:** Contacto relacionado con el ticket.
- **Usuario:** Usuario responsable del ticket.
- **Resolución.** Resultado de la incidencia.

Para finalizar, pulsaremos el botón *Crear ticket* para guardar el nuevo registro o *Cancelar* para salir.

4.2. Seleccionar ticket de soporte

Para seleccionar un ticket de soporte haremos clic sobre su nombre desde el listado tickets.

The screenshot shows a web application interface for a CRM system. At the top is a dark navigation bar with the 'crm' logo and menu items: 'Contactos', 'Campañas', 'Actividades', 'Alertas', 'Soporte' (highlighted), and 'Usuarios'. On the right of the bar is a user profile icon and the text 'Hola Administrador'. Below the navigation bar, the main content area is divided into two columns. The left column, titled 'Datos generales', contains a form with the following fields: 'Asunto' (Factura duplicada), 'Descripción' (Factura 123 cobrada por duplicado), 'Contacto' (Barcelona, Bárbara), 'Estado' (Cerrado), 'Prioridad' (Alta), 'Asignada a' (Administrador Administrador), and 'Resolución' (Segundo cobro devuelto). The right column contains a vertical list of buttons: 'Listar mis tickets', 'Listar todos los tickets', 'Nuevo ticket', 'Editar ticket', and 'Eliminar ticket'. Below the 'Datos generales' section is an 'Actividades' section with a table. The table has columns for '#', 'Asunto', 'Inicio', 'Tipo', and 'Estado'. It contains one row with the following data: '# 38', 'Asunto Llamada a Bárbara', 'Inicio 26-08-2014 09:15', 'Tipo Llamada telefónica', and 'Estado Completada'. Below the table, it says '1 actividad'. At the bottom of the page, there is a footer with the text 'Tiempo de respuesta: 0.0656 segundos' and '© Agustín Ruiz 2014'.

Datos generales

Asunto: Factura duplicada

Descripción: Factura 123 cobrada por duplicado

Contacto: Barcelona, Bárbara

Estado: Cerrado

Prioridad: Alta

Asignada a: Administrador Administrador

Resolución: Segundo cobro devuelto

Actividades

#	Asunto	Inicio	Tipo	Estado
38	Llamada a Bárbara	26-08-2014 09:15	Llamada telefónica	Completada

1 actividad

Tiempo de respuesta: 0.0656 segundos

© Agustín Ruiz 2014

Figura 13: Vista de ticket de soporte

En la pantalla veremos todos los datos del ticket, el contacto relacionado, el usuario al que ha sido asignado y las actividades relacionadas con él. Podremos acceder a dichos registros pulsando directamente sobre el nombre o el asunto.

En ésta ventana se habilitan los botones *Editar ticket* y *Eliminar ticket*. En el primer caso se verá un formulario similar al de creación de una nuevo ticket para la edición del mismo.

En el caso de eliminar el ticket se mostrará un mensaje modal para confirmar si se desea eliminar el ticket definitivamente.

5. GESTIÓN DE ACTIVIDADES

Las actividades son las distintas comunicaciones particulares que se realizan con un contacto y que pueden ser independientes o estar asociadas a una campaña o un ticket.

Para entrar en éste módulo seleccionaremos *Actividades* del menú superior.

The screenshot shows the CRM interface with the 'Actividades' menu selected. The main area displays a table of activities with columns for #, Asunto, Campaña, Ticket, Inicio, Tipo, Estado, Contacto, and Usuario. The table lists five activities, with the last one being 'Otra' (Ticket de administrador) in a 'Pendiente información' state. A sidebar on the right contains a 'Listar mis actividades' menu with options: 'Listar todas las actividades', 'Nueva actividad', 'Editar actividad', and 'Eliminar actividad'. At the bottom, it shows 'Tiempo de respuesta: 0.1045 segundos' and '© Agustín Ruiz 2014'.

#	Asunto	Campaña	Ticket	Inicio	Tipo	Estado	Contacto	Usuario
33	Otra más	Campaña activa		14-05-2014 13:32	Carta	No iniciada	Ruiz Linares, Agustín	Administrador, Administrador
34	Aún estás a tiempo			08-08-2014 13:56	Llamada telefónica	No iniciada	Ruiz Linares, Agustín	Administrador, Administrador
38	Llamada a Bárbara		Factura duplicada	26-08-2014 09:15	Llamada telefónica	Completada	Barcelona, Bárbara	Administrador, Administrador
37	Actividad nueva			26-08-2014 16:01	Correo electrónico	Completada	Barcelona, Bárbara	Administrador, Administrador
32	Otra		Ticket de administrador	13-11-2014 13:30	Llamada telefónica	Pendiente información	Barcelona, Bárbara	Administrador, Administrador

1-7 de 7

Tiempo de respuesta: 0.1045 segundos
© Agustín Ruiz 2014

Figura 14: Listar actividades

Por defecto, se muestra un listado con las actividades propias del usuario. En el menú de la derecha se puede seleccionar la opción *Listar todas las actividades* para ver las asignadas a otros usuarios.

En la parte superior tenemos un campo de búsqueda genérico.

Haciendo clic sobre el encabezado de las columnas podemos elegir por qué criterio se ordenan los resultados. Para cambiar el orden de ascendente a descendente (y viceversa), deberemos volver a hacer clic en el encabezado.

5.1. Crear actividades

Hacemos clic sobre la entrada *Nueva actividad* del menú lateral.

CRM Contactos Campañas **Actividades** Alertas Soporte Usuarios Hola Administrador

Asunto* Nombre de la actividad

Inicio*

Finalización No especificado

Tipo* Llamada telefónica

Prioridad* Media

Estado* No iniciada

Contacto*

Campaña

Ticket

Usuario* Administrador Administrador

Descripción* Descripción de la actividad

Resultado Resultado de la actividad

Crear actividad Cancelar

Los campos marcados con un asterisco (*) son obligatorios
Los campos **Campaña** y **Ticket** son excluyentes

Listar mis actividades
Listar todas las actividades
Nueva actividad
Editar actividad
Eliminar actividad

Tiempo de respuesta: 0.0611 segundos
© Agustín Ruiz 2014

Figura 15: Formulario nueva actividad

Los campos marcados con un asterisco son obligados. Los campos del formulario a cumplimentar son los siguientes:

- **Asunto:** Asunto de la actividad.
- **Inicio:** Fecha y hora de inicio de la actividad.
- **Finalización:** Fecha y hora de finalización de la actividad.
- **Tipo:** Tipo de actividad. Los posibles valores son Llamada telefónica, Correo electrónico, Carta, Fax, Tarea o Reunión.
- **Prioridad:** Prioridad de la actividad. Los posibles valores son Alta, Media o Baja.
- **Estado:** Estado de la actividad. Los posibles valores son No iniciada, En progreso, Completada, Pendiente de información, Aplazada o Cancelada.
- **Contacto:** Contacto relacionado con la actividad.
- **Campaña:** Campaña relacionada con la actividad. Es excluyente con el campo Ticket.
- **Ticket:** Ticket relacionado con la actividad. Es excluyente con el campo Campaña.

- **Usuario:** Usuario responsable de la actividad.
- **Descripción:** Descripción de la actividad.
- **Resultado:** Resultado de la actividad.

Para finalizar, pulsaremos el botón *Crear actividad* para guardar el nuevo registro o *Cancelar* para salir.

5.2. Seleccionar actividad

Para seleccionar una actividad haremos clic sobre su nombre desde el listado de actividades.

The screenshot shows a web application interface for a CRM system. At the top, there is a navigation bar with tabs: 'Contactos', 'Campañas', 'Actividades' (selected), 'Alertas', 'Soporte', and 'Usuarios'. A user profile 'Hola Administrador' is visible on the right. The main content area displays the details of a specific activity titled 'Llamada a Bárbara'. The details are organized into a table-like structure with the following fields:

Asunto	Llamada a Bárbara		
Inicio	26-08-2014 09:15	Finalización	-
Tipo	Llamada telefónica	Prioridad	Alta
Estado	Completada		
Contacto	Bárbara Barcelona		
Ticket	Factura duplicada		
Usuario	Administrador Administrador		
Descripción	Se le informa de la devolución del importe cobrado por segunda vez		
Resultado			

To the right of these details is a vertical menu with the following options: 'Listar mis actividades', 'Listar todas las actividades', 'Nueva actividad', 'Editar actividad', 'Eliminar actividad', and 'Nueva alerta para esta actividad'. Below the activity details, there is an 'Alertas' section with a table header: '#', 'Asunto', 'Fecha y hora', 'Notificaciones', and 'Visualizada'. The table body contains the text 'No hay alertas programadas'. At the bottom of the page, it shows 'Tiempo de respuesta: 0.0629 segundos' and '© Agustín Ruiz 2014'.

Figura 16: Vista de campaña

En la pantalla veremos todos los datos de la actividad, el contacto relacionado con ella, el usuario al que ha sido asignada, la campaña o ticket de soporte con la que está relacionada y las alertas vinculadas a ella. Podremos acceder a dichos registros pulsando directamente sobre el nombre o el asunto.

En el menú derecho tendremos un botón nuevo: *Nueva alerta para esta actividad*.

En ésta ventana se habilitan los botones *Editar actividad* y *Eliminar actividad*. En el primer caso se verá un formulario similar al de creación de una nueva actividad para la edición de la misma.

En el caso de eliminar la actividad se mostrará un mensaje modal para confirmar si se desea eliminar la actividad definitivamente.

6. GESTIÓN DE ALERTAS

Las alertas son eventos programados que se mostrarán al usuario sobre actividades, ya sean emergentes o envíos por correo electrónico.

Para entrar en éste módulo seleccionaremos *Alertas* del menú superior.

#	Asunto	Fecha y hora	Notificaciones	Visualizada
14	Evento para el futuro	17-09-2014 07:40		
7	Lorem ipsum dolor sit amet, consectetur adipiscing elit	26-08-2014 10:00		
13	Llamada telefónica con Bárbara Barcelona	13-08-2014 14:15		
10	Preparar documentación para la reunión de mañana	13-08-2014 13:48		
11	Para el lunes	12-08-2014 14:14		

1-5 de 5

Tiempo de respuesta: 0.0401 segundos
© Agustín Ruiz 2014

Figura 17: Listar alertas

Por defecto, se muestra un listado con las alertas propias del usuario. En la parte superior tenemos un campo de búsqueda genérico.

Haciendo clic sobre el encabezado de las columnas podemos elegir por qué criterio se ordenan los resultados. Para cambiar el orden de ascendente a descendente (y viceversa), deberemos volver a hacer clic en el encabezado.

Cuando se cumple la fecha y hora de una alerta aparece un icono en la barra superior con el número de alertas pendientes.

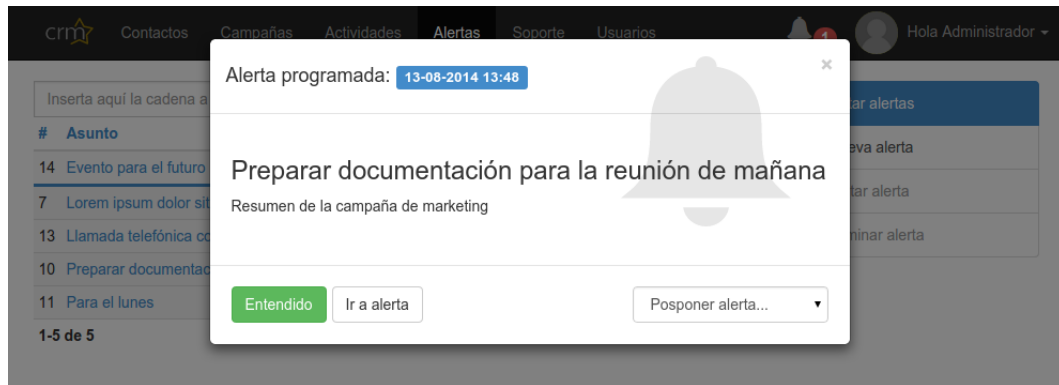


Figura 18: Ejemplo de notificación emergente

En la ventana que aparece podemos marcar la notificación como leída pulsando el botón *Entendido*, *Ir a alerta* o posponer la alerta un cierto tiempo.

6.1. Crear alerta

Hacemos clic sobre la entrada *Nueva alerta* del menú lateral.

Figura 19: Formulario nueva alerta

Los campos marcados con un asterisco son obligados. Los campos del formulario a cumplimentar son los siguientes:

- **Asunto:** Asunto de la alerta.
- **Fecha/Hora:** Fecha y hora de la notificación de la alerta.
- **Descripción:** Descripción de la alerta.

- **Enviar email:** Activar o no el envío por correo electrónico de la alerta al correo electrónico del usuario (no disponible en el prototipo).
- **Ventana emergente:** Mostrar la alerta emergente mientras se usa la aplicación o dejar que sea una alerta silenciosa que sólo se muestre el icono en el menú superior.
- **Actividad:** Seleccionar la actividad relacionada con la alerta.

Para finalizar, pulsaremos el botón *Crear alerta* para guardar el nuevo registro o *Cancelar* para salir.

6.2. Seleccionar alerta

Para seleccionar una alerta haremos clic sobre su nombre desde el listado de alertas.

The screenshot shows a web application interface for managing alerts. At the top is a dark navigation bar with the CRM logo and menu items: Contactos, Campañas, Actividades, **Alertas**, Soporte, and Usuarios. On the right of the navigation bar, it says 'Hola Administrador' next to a user icon. The main content area has a title 'Llamada telefónica con Bárbara Barcelona'. Below the title is a form with the following fields: 'Asunto' (Llamada telefónica con Bárbara Barcelona), 'Fecha/Hora' (13-08-2014 14:15), 'Descripción' (empty), 'Enviar email' (ari00029@red.ujaen.es), and 'Ventana emergente' (checked, with a 'Sí' label). Below these fields is the 'Actividad' section, showing a blue circle icon and the text 'Otra (Llamada telefónica con Bárbara Barcelona)'. At the bottom of the form, it says 'Alerta visualizada' with a green 'Sí' label and a blue button labeled 'Marcar NO visualizada'. To the right of the form is a vertical sidebar with four buttons: 'Listar alertas', 'Nueva alerta', 'Editar alerta', and 'Eliminar alerta'. At the very bottom of the page, it shows 'Tiempo de respuesta: 0.1131 segundos' and '© Agustín Ruiz 2014'.

Figura 20: Vista de alerta

En la pantalla veremos todos los datos de la alerta y la actividad asociada a la misma, a la que podremos acceder pulsando sobre su asunto.

También se mostrará un botón para marcar la alerta visualizada si la fecha y hora de ésta ya ha expirado.

En ésta ventana se habilitan los botones *Editar alerta* y *Eliminar alerta*. En el primer caso se verá un formulario similar al de creación de una nueva alerta para la edición de la misma.

En el caso de eliminar la alerta se mostrará un mensaje modal para confirmar si se desea eliminar la alerta definitivamente.

7. GESTIÓN DE USUARIOS

Los usuarios son las cuentas con las que los empleados de la organización pueden acceder al sistema.

Para el acceso al módulo de contactos entramos en el apartado *Usuarios* del menú superior. Por defecto, encontraremos un listado con todos los usuario.

Inserta aquí la cadena a buscar...

#	Apellidos, Nombre	Nick	Email	Perfil	Estado
1	Administrador, Administrador	admin	arl00029@red.ujen.es	Superusuario	Activo
13	Ruiz Linares, Agustín	aruiz	agustinruizlinares@gmail.com	Superusuario	Activo
14	Uno, Uno	uno	uno@uno.com	Coordinación	Despedido
15	Dos, Dos	dos	dos@dos.com	Comercial	Excedencia
16	Tres, Tres	tres	tres@tres.com	Técnico	Activo
17	Cuatro, Cuatro	cuatro	cuatro@cuatro.com	Coordinación	Activo
18	Cinco, Cinco	cinco	cinco@cinco.com	Comercial	Activo
19	Seis, Seis	seis	seis@seis.com	Técnico	Activo

1-8 de 17

1 2 3 »

Lista usuarios

Nuevo usuario

Editar usuario

Eliminar usuario

Listar perfiles de usuario

Tiempo de respuesta: 0.2989 segundos

© Agustín Ruiz 2014

Figura 21: Listar usuarios

En la parte superior tenemos un campo de búsqueda genérico.

Haciendo clic sobre el encabezado de las columnas podemos elegir por qué criterio se ordenan los resultados. Para cambiar el orden de ascendente a descendente (y viceversa), deberemos volver a hacer clic en el encabezado.

7.1. Crear un nuevo usuario

Hacemos clic sobre la entrada *Nuevo usuario* del menú lateral.

Datos generales

Nombre*

Apellidos*

NIF

Identificador*

Perfil de usuario*

Estado de usuario*

Correo electrónico*

Telf. Oficina

Telf. Móvil

FAX

Otros datos

Los campos marcados con un asterisco (*) son obligatorios

Tiempo de respuesta: 0.1450 segundos
© Agustín Ruiz 2014

Figura 22: Formulario de nuevo usuario

Los campos marcados con un asterisco son obligados. Los campos del formulario a cumplimentar son los siguientes:

- **Nombre:** Nombre del usuario.
- **Apellidos:** Apellidos del usuario.
- **NIF:** NIF de la persona o empresa.
- **Indicador:** Nombre de usuario con el que el empleado accederá al sistema.
- **Perfil de usuario:** Tipo de usuario que se está creando. Los posibles valores son *Técnico*, *Comercial*, *Coordinador* y *Superadministrador*.
- **Estado del usuario:** Los valores posibles son *Activo*, *Excedencia* o *Despedido*. Si el usuario está en excedencia o despedido no podrá acceder al sistema.
- **Correo electrónico:** Correo electrónico del usuario. A ésta cuenta se mandará las notificaciones y contraseñas generadas.
- **Telf. Oficina:** Teléfono de oficina del usuario.

- **Telf. Móvil:** Teléfono móvil del usuario.
- **FAX:** Teléfono FAX del usuario.
- **Otros datos:** Datos adicionales acerca del usuario.

Para finalizar, pulsaremos el botón *Crear usuario* para guardar el nuevo registro o *Cancelar* para salir.

Al crear un nuevo usuario se generará una contraseña aleatoria que se enviará al correo electrónico indicado.

7.2. Seleccionar usuario

Para seleccionar un usuario haremos clic sobre su nombre desde el listado de usuarios.

The screenshot displays a web application interface for user management. At the top, a navigation bar includes links for 'Contactos', 'Campañas', 'Actividades', 'Alertas', 'Soporte', and 'Usuarios' (which is highlighted). A user profile icon and the text 'Hola Administrador' are on the right. The main content area shows the details for a user named 'Agustín Ruiz Linares'. It includes a summary section with fields for Nick (aruz), Contraseña (with a 'Reiniciar contraseña' link), Perfil de usuario (Superusuario), and Estado (Activo). Below this is a 'Datos generales' section with a table containing fields like Nombre (Agustín), Apellidos (Ruiz Linares), NIF, Correo (agustinruizlinares@gmail.com), Telf. Oficina, Telf. Móvil, FAX, and Otros datos. To the right of this section is a vertical menu with options: 'Listar usuarios', 'Nuevo usuario', 'Editar usuario', 'Eliminar usuario', and 'Listar perfiles de usuario'. Further down, the 'Campañas asignadas' section shows a table with one entry: 'Campaña activa' (Ventas, Activa). The 'Actividades' section shows a table with one entry: 'Actividad de Agustín' (Correo electrónico, No iniciada). At the bottom, it shows the response time '0.1427 segundos' and the copyright '© Agustín Ruiz 2014'.

Figura 23: Visualización de usuarios

En la pantalla veremos todos los datos del usuario, las campañas asignadas y las actividades realizadas. Podremos acceder a dichos registros pulsando directamente sobre el nombre o el asunto.

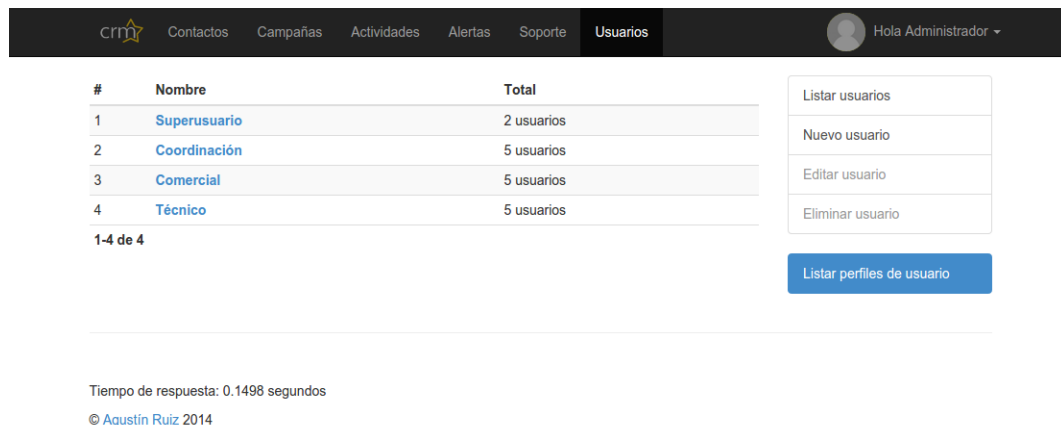
Si tenemos los permisos adecuados, también podremos generar una nueva contraseña para el usuario que se le enviará a la dirección de correo indicada.

En ésta ventana se habilitan los botones *Editar usuario* y *Eliminar usuario*. En el primer caso se verá un formulario similar al de creación de un nuevo usuario para la edición del mismo.

En el caso de eliminar el usuario se mostrará un mensaje modal para confirmar si se desea eliminar el usuario definitivamente.

7.3. Perfiles

Si pulsamos en el botón Listar perfiles de usuario podremos ver un listado de los perfiles de usuario.



The screenshot shows a web application interface for managing users. At the top is a navigation bar with the 'CRM' logo and several menu items: 'Contactos', 'Campañas', 'Actividades', 'Alertas', 'Soporte', and 'Usuarios' (which is highlighted). On the right side of the navigation bar, there is a user profile icon and the text 'Hola Administrador'. Below the navigation bar is a table with three columns: '#', 'Nombre', and 'Total'. The table contains four rows of data. To the right of the table is a sidebar with four buttons: 'Listar usuarios', 'Nuevo usuario', 'Editar usuario', and 'Eliminar usuario'. Below these buttons is a blue button labeled 'Listar perfiles de usuario'. At the bottom of the page, there is a footer with the text 'Tiempo de respuesta: 0.1498 segundos' and '© Agustín Ruiz 2014'.

#	Nombre	Total
1	Superusuario	2 usuarios
2	Coordinación	5 usuarios
3	Comercial	5 usuarios
4	Técnico	5 usuarios

1-4 de 4

Tiempo de respuesta: 0.1498 segundos
© Agustín Ruiz 2014

Figura 24: Listado de perfiles de usuario

Para ver un perfil haremos clic sobre el nombre del mismo. En la ventana veremos la política del perfil y los usuarios que tiene asignado.

CRM

Contactos

Campañas

Actividades

Alertas

Soporte

Usuarios

Hola Administrador

Coordinación

Nombre del perfil **Coordinación**

Política

Permisos contactos

Listar

✓ SI

Crear

✓ SI

Editar

✓ SI

Eliminar

✓ SI

Permisos campañas

Listar todas las campañas

✓ SI

Listar campañas propias

✓ SI

Crear

✓ SI

Editar

✓ SI

Eliminar

✓ SI

Permisos actividades

Listar todas las actividades

✓ SI

Crear

✓ SI

Editar

✓ SI

Eliminar

✓ SI

Listar actividades propias

✓ SI

Crear propias

✓ SI

Editar propias

✓ SI

Eliminar propias

✓ SI

Permisos para soporte

Listar todos

NO

Listar tickets propios

NO

Crear

NO

Crear tickets propios

NO

Editar

NO

Editar tickets propios

NO

Eliminar

NO

Eliminar propios

NO

Permisos usuarios

Listar

✓ SI

Crear

NO

Editar

NO

Eliminar

NO

Permisos perfiles de usuario

Listar

✓ SI

Listar usuarios

Nuevo usuario

Editar usuario

Eliminar usuario

Listar perfiles de usuario

Usuarios

#	Nombre	Identificador	Email
14	Uno Uno	uno	uno@uno.com
17	Cuatro Cuatro	cuatro	cuatro@cuatro.com
20	Siete Siete	siete	siete@siete.com
23	Diez Diez	diez	diez@diez.com
26	Trece Trece	trece	trece@trece.com

5 usuarios

Tiempo de respuesta: 0.1223 segundos

© Agustín Ruiz 2014

Figura 25: Visualización de perfil de usuario

Anexo IV: Descripción de contenidos suministrados

En el CD suministrador se encuentran los siguientes contenidos:

- **Memoria.pdf**: Memoria del proyecto.
- **crmStar**: Directorio con el trabajo desarrollado.

Anexo V: Acceso a la plataforma de pruebas

Para probar el prototipo en funcionamiento se ha instalado la aplicación en un servidor accesible a través de Internet en la siguiente dirección:

<http://www.agustruiz.es/crm>

Se ha provisto la plataforma de datos de ejemplo con los que poder probar las distintas funcionalidades de la herramienta. Se ha habilitado distintos perfiles de usuario para pruebas. Las credenciales de acceso son las siguientes:

<i>Perfil</i>	<i>Usuario</i>	<i>Contraseña</i>
Superusuario	admin	123456
Coordinador	coordinador	123456
Comercial	comercial1	123456
Comercial	comercial2	123456
Técnico	tecnico1	123456
Técnico	tecnico2	123456

Por motivos de acceso a cualquier usuario, se ruega no modificar ninguna contraseña del sistema.

Si el lector de ésta memoria se encontrase algún problema, en el footer de todas las páginas de la plataforma se encuentra los datos de contacto del webmaster para comunicárselo y se solucione el problema en la mayor brevedad de tiempo.