

IES SAN SEBASTIÁN

I.E.S. San Sebastián

C.F.G.S. Desarrollo de Aplicaciones Web

2014/2015



Sara Alamillo Arroyo

Índice

0 ► Objetivos generales del proyecto.....	4
1 ► Planificación del proyecto.....	4
1.1 ► Estimación Económica Inicial.....	5
1.1.1 ► Definición de Actividades	5
1.1.2 ► Recursos humanos.....	7
1.2 ► Políticas de seguimiento y evaluación.....	8
1.3 ► Análisis de requisitos de la aplicación.....	9
1.3.1 ► Análisis del sistema actual. Justificación del proyecto....	9
1.3.2 ► Búsqueda de necesidades. NEEDFINDING.....	10
1.3.3 ► Lista de necesidades del sistema.....	11
1.3.4 ► Tablero de inspiración de diseños.....	12
1.3.5 ► Estrategia de diseño y puntos de vista.....	14
1.3.6 ► Requisitos físicos.....	16
1.3.7 ► Requisitos lógicos/funcionales.....	18
1.3.8 ► Diagramas de casos de uso.....	21
1.3.9 ► Requisitos no funcionales.....	23
1.3.10 ► Prototipo de la aplicación.....	24
1.3.11 ► Mapa de navegación de la aplicación.....	29
2 ► Modelo de datos.....	30
2.1 ► Diagrama Entidad/Relación.....	30
2.2 ► Esquema de la base de datos (MySQLWorkbench).....	31
3 ► Tecnologías utilizadas en el desarrollo del proyecto.....	34
3.1 ► Por qué usar tecnologías de Software Libre.....	34
3.2 ► Parte CLIENTE.....	34
3.2.1 ► HTML.....	34
3.2.2 ► CSS.....	35
3.2.3 ► Javascript.....	36

3.24 ▶ Bootstrap.....	36
3.25 ▶ jQuerry.....	38
3.3 ▶ Parte SERVIDOR.....	38
 3.3.1 ▶ Servidor Web Apache.....	38
 3.3.2 ▶ PHP.....	39
 3.3.3 ▶ MySQL.....	40
 3.3.4 ▶ CodeIgniter.....	41
 3.3.5 ▶ Control de Versiones. GitHub.....	42
4 ▶ Arquitectura de la aplicación. El Patrón MVC.....	43
5 ▶ Manual de usuario.....	45
 5.1 ▶ Login.....	45
 5.2 ▶ Pantalla de inicio.....	45
 5.3 ▶ Pantalla Viviendas.....	46
 5.4 ▶ Pantalla Hermanos.....	47
 5.5 ▶ Pantalla Remesas.....	50
 5.6 ▶ Pantalla Pagos.....	51
 5.7 ▶ Pantalla Agenda de Contactos.....	53
 5.8 ▶ Pantalla Sorteo de Medallas.....	55
6 ▶ Plan de Evaluación	56
 6.1 ▶ Realización de la prueba	56
 6.2 ▶ Instrucciones para la realización de la prueba	56
 6.3 ▶ Valoración y resultados de la prueba.....	58
Cuestionario de Evaluación.....	59
Cuestiones finales.....	62
7 ▶ Posibles mejoras o ampliaciones del proyecto.....	65
8 ▶ Conclusiones personales.....	65
▶ Bibliografía	67
▶ Fuentes de Internet	68

0 ► Objetivos generales del proyecto

La Hermandad de San Telmo, tal como su nombre indica, es una Hermandad que venera la figura de San Telmo y que se encuentra ubicada en la población de San Telmo, perteneciente al término municipal de Cortegana, Huelva.

En la actualidad la totalidad de la gestión de la Hermandad se realiza de manera manual mediante el uso de una serie de fichas de datos y resguardos definidos a tales fines en las que se recogen los distintos datos que resulten de interés conocer. Observando los datos recogidos en las fichas que la Hermandad utiliza, se deduce que existe una escasa información registrada de los hermanos por lo que será necesario ampliar la información registrada de manera que permita facilitar aspectos básicos como sería el de la comunicación Hermandad-hermano.

Dado lo tedioso de la gestión manual y la gran cantidad de documentos que supone manejar, la Hermandad ha decidido desarrollar una aplicación informática, que se pueda ejecutar en cualquier tipo de dispositivo móvil, que permita automatizar el manejo de información y las distintas tareas así como minimizar el número de documentos a manejar.

1 ► Planificación del proyecto

Desde el inicio del proyecto es importante poder planificar todo lo que se va a realizar para llevar una organización y poder determinar el tiempo y el coste que nos llevará realizarlo. En esta sección se detallarán las actividades, estimando los tiempos, los recursos que se necesitaran para realizarla y los costes.



Fig.0 Esquema del Proceso de Diseño Web

1.1 ► Estimación Económica Inicial

En este apartado explicaremos una estimación del coste económico necesario para la realización del proyecto. Para obtener el estimado realizamos varios pasos que explicaremos a continuación

1.1.1 ► Definición de Actividades

Las actividades son el resultado final de descomponer el trabajo a realizar en el proyecto en paquetes o fases que puedan ser controlados y gestionados. Al tratarse de un proyecto de desarrollo e implantación de un software web, podemos utilizar algunas de las metodologías más populares dentro del mundo de la ingeniería del software. Por ello se decidió usar el modelo clásico, que es dividir el proyecto en fases secuenciales.

Las fases que se definieron son: Análisis, Diseño, Implementación y Pruebas. Dado que el sistema será propiedad de la Hermandad de San Telmo, la fase de Mantenimiento dependerá de ella. El hecho de especificar este modo indica que, si seguimos un orden lógico y no aparecen errores ni contratiempos, las fases sucederán cronológicamente según se muestra la tabla siguiente:



Planificación Aplicación GestST

Tarea	Inicio	Fin
•Análisis		
→ Estudios previos. Needfinding	01/04/15	05/04/15
→ Requisitos físicos	06/04/15	06/04/15
→ Requisitos funcionales	06/04/15	15/04/15
→ Aprobación del cliente	16/04/15	16/04/15
•Diseño	17/04/15	06/05/15
→ Selección de tecnologías a utilizar	17/04/15	20/04/15
→ Formación tecnologías a utilizar	21/04/15	28/04/15
→ Definir prototipo de baja fidelidad del sitio	29/04/15	29/04/15
→ Estudio de alternativas de implementación	30/04/15	02/05/15
→ Definir prototipo de alta fidelidad del sitio	03/05/15	05/05/15
→ Aprobación del cliente	06/05/15	06/05/15
•Implementación	07/05/15	05/06/15
→ Página Inicio	07/05/15	09/05/15
→ Página Viviendas	10/05/15	11/05/15
→ Página Hermanos	12/05/15	18/05/15

→ Página Remesas	19/05/15	19/05/15
→ Página Pagos	20/05/15	21/05/15
→ Página Consultas	22/05/15	28/05/15
→ Página Agenda de Contactos	29/05/15	01/06/15
→ Página Sorteo de Medallas	02/06/15	04/06/15
→ Aprobación del cliente	05/06/15	05/06/15
• Evaluación y Pruebas	06/06/15	16/06/15
→ Preparación de la prueba	06/06/15	07/06/15
→ Realización de la prueba	08/06/15	08/06/15
→ Evaluación de los resultados de la prueba	08/06/15	08/06/15
→ Generar lista de cambios	09/06/15	09/06/15
→ Aplicar cambios a la aplicación	10/06/15	16/06/15
• Entrega de la aplicación	17/06/15	22/06/15
→ Documentar la aplicación	17/06/15	21/06/15
→ Presentación de la aplicación	22/06/15	22/06/15
→ Aprobación del cliente	22/06/15	22/06/15

►Diagrama de Gantt de la aplicación

El *diagrama de Gantt* es una herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado. La siguiente figura muestra el diagrama de Gantt que representa cuál sería la planificación inicial de desarrollo de nuestro proyecto:

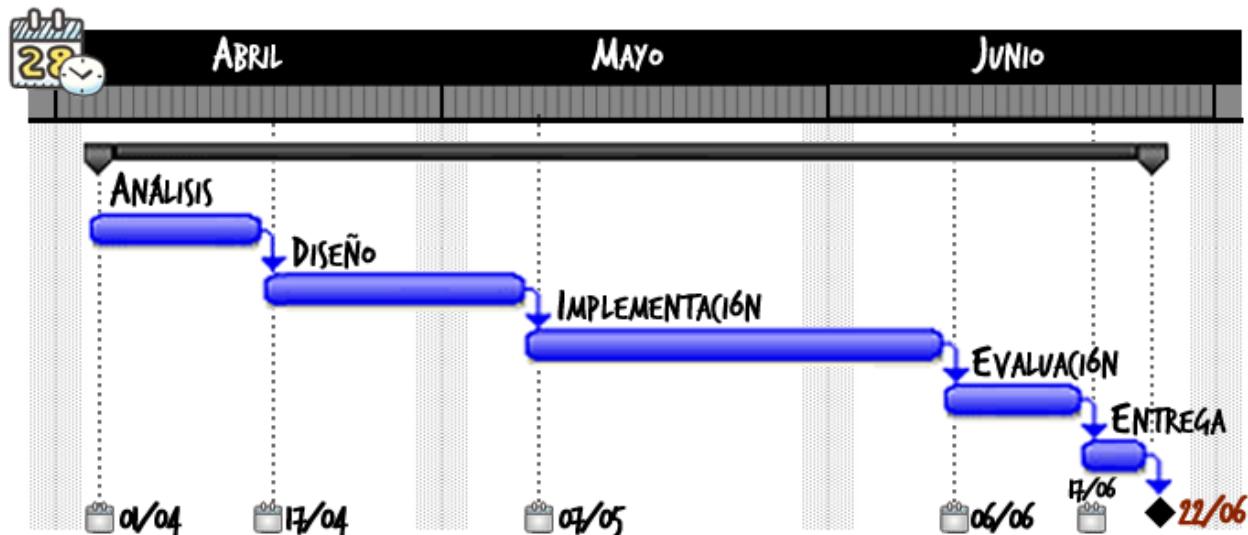


Figura 2. Diagrama de Gantt de la aplicación

112 Recursos humanos

Para este proyecto se identificaron los roles de las personas que típicamente intervienen en la realización de las distintas actividades como son el jefe de proyecto, analista, diseñador, programador y tester. Pero en nuestro caso, al ser un trabajo unipersonal, todas las tareas serían realizadas por mi persona por ello, para calcular el coste de un proyecto tendríamos en cuenta el tiempo aproximado dedicado a cada tarea, de manera que para cada rol multiplicaríamos el número de horas estimadas por un precio de coste razonable y ajustado al mercado. En nuestro caso, al haberlo implementado una sola persona, el cálculo lo basaremos en el tiempo requerido para implementar cada una de las tareas y en un precio de nuestra hora de trabajo que se aproximen a los de la actualidad.



Estimación de costes de la Aplicación GestST

Tarea	Horas	Precio	Coste
• Análisis	$16*6=96$	10€/h	960€
• Diseño	$19*6=114$	10€/h	1140€
• Implementación	126	10€/h	1260€
→ Página Inicio	$2*6=12$	10€/h	120€
→ Página Viviendas	$1*6=6$	10€/h	60€
→ Página Hermanos	$6*6=36$	10€/h	360€
→ Página Remesas	$1*6=6$	10€/h	60€
→ Página Pagos	$1*6=6$	10€/h	60€
→ Página Consultas	$6*6=36$	10€/h	360€
→ Página Agenda de Contactos	$2*6=12$	10€/h	120€
→ Página Sorteo de Medallas	$2*6=12$	10€/h	120€
• Evaluación y Pruebas	$10*6=60$	10€/h	600€
• Entrega de la aplicación	$5*6=30$	10€/h	300€
Coste total de la aplicación			3900€

1.2 ► Políticas de seguimiento y evaluación

- Antes de iniciar una tarea se deben tomar los requisitos que necesita para realizarla satisfactoriamente.
- Al terminar una tarea se deben de hacer las pruebas de funcionamiento pertinentes que demuestren el debido funcionamiento de la misma.
- Cada vez que se termine una tarea se hará un análisis de su estructura para saber si hay alguna mejora pertinente que podría ser más productiva.
- Cada tarea deberá ser elaborada en el tiempo establecido para la misma.
- Al terminar la aplicación se deberán hacer pruebas con personas para determinar la experiencia de usuario, si es satisfactoria o si tiene dificultades para encontrar acciones u operaciones realizando tareas.
- En el desarrollo del *backend* se deberá seguir el diagrama de clases que se elaboró en el proyecto.
- Se deberá revisar continuamente los requisitos de la aplicación para mejorar en medida de lo posible los posibles errores que hubiesen.
- Se harán constantemente pruebas de rendimiento de la aplicación para comprobar el funcionamiento del código y las consultas hacia la base de datos e intentar optimizar si fuese preciso.
- Se deberá realizar un diagrama de flujo para la interacción del usuario, con esto se pretende hacer un diseño productivo e intuitivo.
- Antes de iniciar el proyecto se deberá crear un repositorio local para poder trabajar en él y hacer una rama de trabajo.
- Al terminar una tarea y comprobar que no existen errores y su funcionamiento es adecuado se deberá hacer un *commit* en la rama de trabajo.
- Siempre que se cree un método dentro de una clase deberá ser comentado para saber qué es lo que hace el método.
- Siempre que se estén probando las vistas de la aplicación se deberá de ejecutar el depurador del navegador para saber si existen bugs por resolver.
- Bajo ningún concepto se deberá de cambiar la filosofía del framework de *Codeigniter*, que consta del patrón de diseño MVC, con esto se persigue mantener un código legible, y que pueda ser escalable con mayor facilidad.

- ▶ Al terminar una tarea y que su funcionalidad es correcta se deberá comprobar el código para verificar si hay opciones de mejorarlo y hacerlo más funcional o eliminar algún posible error.

1.3 ▶ Análisis de requisitos de la aplicación

1.3.1 ▶ Análisis del sistema actual. Justificación del proyecto

Con el fin de elaborar la lista de requisitos que deberá satisfacer mi aplicación, me reuní y entrevisté al secretario de la Hermandad quien llevaba varios años encargado de la gestión de la misma. Para ello, le pregunté a cerca de cómo realizaba las distintas tareas de gestión y qué dificultades encontraban para llevarlas a cabo.

Durante las entrevistas fui tomando nota de toda la información que podría resultar de interés para la fijar las necesidades que presentan los usuarios así como fotografié los distintos documentos utilizados para llevar la gestión los cuales se muestran a continuación.

La Hermandad de San Telmo, tal como su nombre indica, es una Hermandad que venera la figura de San Telmo y que se encuentra ubicada en la población de San Telmo, perteneciente al término municipal de Cortegana, Huelva.

Como en la actualidad la totalidad de la gestión de los distintos aspectos de la Hermandad, como son los relativos a gestión de hermanos, pagos de cuotas, etc, se realiza de manera manual mediante el uso de una serie de fichas de datos definidos a tales fines en las que se recogen los distintos datos que resulten de interés conocer, he decidido desarrollar una aplicación web, que permita controlar de manera fácil y centralizada todos los aspectos que conlleva la gestión de la Hermandad.

Observando los datos recogidos en las fichas que la Hermandad utiliza, se deduce que existe una escasa información registrada de los hermanos, apenas sus nombres completos dentro de lo que sería una ficha familiar que engloba a todos sus miembros, por lo que será necesario ampliar la información registrada de manera que se facilite aspectos básicos como sería el de la comunicación Hermandad-hermano.

Respecto al pago de las cuotas de hermanos, éstas se realizan de manera presencial: miembros de la Junta de Gobierno visitan las casas de los hermanos. En cada casa se podrá realizar no sólo los pagos de los miembros de la propia familia residente, sino también el de otros hermanos con cierta vinculación a ésta que tengan establecidos como lugar de pago dicha vivienda. Las cuotas podrán fraccionarse en dos pagos y se le entrega a cada hermano una pequeña cartulina acreditativa del pago realizado. Ante la creciente demanda de los hermanos, sobre todo los que residen de manera habitual fuera de la provincia de Huelva, la Hermandad se está planteando ofertar la posibilidad del pago mediante domiciliación bancaria. Asimismo existen hermanos sin residencia asignada que realizan el pago de sus cuotas a cualquier miembro de la hermandad en cualquier momento.

La Hermandad requiere de una serie de servicios como son relativos a aspectos de seguridad, sanitarios, recogida de residuos, tamboril, cohete, etc, de los que cada romería se hacen uso. Para

contactar con ellos se dispone de una carpeta de contactos con tarjetas de visitas de las personas asociadas a cada servicio. Será necesario, vista la gestión actual, crear una pequeña agenda de contactos de la hermandad que nos facilite el acceso y la gestión de cada

Dado lo tedioso de la gestión manual y la gran cantidad de documentos que supone manejar, la Hermandad ha decidido desarrollar una aplicación informática, que se pueda ejecutar en cualquier tipo de dispositivo móvil, que permita automatizar el manejo de información y las distintas tareas así como minimizar el número de documentos a manejar.

A partir de todo ello, tras analizar y organizar la información de las necesidades recogidas por el método de entrevistas a personas pertenecientes a la Junta de Gobierno de la Hermandad, he podido elaborar una lista con los principales objetivos que mi aplicación deberá alcanzar.

1.3.2 ► Búsqueda de necesidades. NEEDFINDING

Para tratar de completar mi conocimiento sobre el funcionamiento de la Hermandad entrevisté al vicepresidente de la misma, quien llevaba tres años participando de manera activa en las distintas tareas de gestión, al que le hice preguntas relativas a cómo las realizaba y qué dificultades encontraban para llevarla a cabo. En ella utilicé básicamente preguntas abiertas en las que pudieran explicar claramente sus respuestas y evité realizar preguntas que pudieran sugerir la respuesta así como realizar comentarios a las respuestas que pudieran influir en la respuesta del usuario.

Cabría señalar que tanto el secretario como el vicepresidente coincidieron en lo incómodo que resultaba tener que manejar tantos papeles para poder llevar a cabo la gestión de los distintos aspectos de la Hermandad.



DAVID

29 años

Vicepresidente de la Hermandad

David comenzó quejándose de la cantidad de papeles que tenía que manejar: fichas de hermanos, fichas de pagos, etc. muchas de ellas con correcciones y anotaciones múltiples. Por ello demanda una forma 'más actual' de poder gestionar toda esa información y, al ser posible, que permita el acceso a través del móvil, pues opina que hoy día todo el mundo dispone de ellos y lo suelen llevar siempre con ellos.

Asimismo cree que resultaría muy útil disponer de algún tipo de sistema que permita acceder de manera inmediata a la información de contacto de los distintos servicios que la Hermandad demanda para la realización de la Romería y demás actos que ésta organiza.

Por otro lado, David expuso un problema que año tras año suele plantearse: la gestión del sorteo de las medallas. Éste se realiza en base al número de hermano, lo que obliga a mantener 'huecos' entre los números de hermano generados por las bajas que se producen. Sería pues interesante que el número de hermano se regenerase automáticamente cuando se produzca una baja. Otro aspecto del sorteo a considerar sería el que para que un hermano pueda entrar en sorteo, además

de no haberle tocado con anterioridad la medalla, deberá estar al día en el pago de las cuotas o tener pendiente, a lo sumo, un año de pago, lo cual implica el tener que cotejar de manera manual cada uno de los pagos de los hermanos para poder definir aquellos que entran en sorteo, tarea muy laboriosa

Por último planteó la necesidad disponer de algún medio que permita mejorar la gestión de los pagos de las cuotas; que permita ir registrando los pagos realizados durante las propias visitas a los domicilios de los hermanos así como conocer la situación de los pagos de un determinado hermano o conseguir un listado con los hermanos que tengan un número determinado de cuotas pendientes.

1.3.3 ▶ Lista de necesidades del sistema

Tras realizar las entrevistas y analizar la información obtenida, tenemos como resultado la lista de las principales necesidades específicas del sistema que se corresponden con las oportunidades para la innovación en el diseño con las que podremos mejorar el desempeño de la actividad observada utilizando herramientas informáticas.

La lista de necesidades de los usuarios sería la siguiente:

1. Los usuarios demandan una aplicación que se pueda ejecutar sobre dispositivos móviles, preferentemente sobre *smartphones* tanto sobre *android* como sobre *Apple iOS*.
2. La aplicación deberá distinguir dos tipos de perfiles básicos de usuario: uno que posea total acceso a las funcionalidades disponibles y otro que sólo pueda realizar consultas de datos.
3. Sería muy interesante el poder disponer de una lista de ‘Consultas frecuentes’, con las que se pueda recuperar una determinada información de manera directa, de forma que éstas pudiesen ser editadas o eliminadas.
4. Debido a que la información que se necesita recuperar puede ser muy variable, resultaría muy útil incluir un pequeño constructor de consultas desde el que se pueda personalizar las consultas y añadir nuevas consultas a la lista de consultas frecuentes.
5. Los resultados de las consultas podrán mostrarse por pantalla y/o ser impresos.
6. La Hermandad necesita que la aplicación incluya una funcionalidad que permita registrar los datos personales de los hermanos para su utilización en su gestión.
7. Es fundamental que nuestro sistema facilite y automatice el proceso de pago de cuotas para agilizar numerosas tareas que éste conlleva, y que permita ir registrando los pagos realizados durante las propias visitas a los domicilios de los hermanos.
8. Será necesario conocer la situación de los pagos de un determinado hermano o conseguir un listado con los hermanos que tengan un número determinado de cuotas pendientes.

9. Debido a que en la gestión de pagos pueden producirse situaciones especiales como serían la del abono parcial de un pago, modificación del lugar de pago, etc..., sería recomendable que nuestra aplicación permitiese incluir notas/aclaraciones relativas a la propia realización del pago.
10. Nuestra aplicación deberá contemplar la posibilidad de los hermanos puedan domiciliar el pago de sus cuotas.
11. Será necesario disponer de algún tipo de sistema de agenda de contactos que permita acceder de manera inmediata a la información de contacto de los distintos servicios que la Hermandad demanda para la realización de la Romería y demás actos que ésta organiza.
12. Dado que la organización del sorteo de las medallas es una de las tareas más tediosas que la Hermandad realiza, sería muy necesario el tratar de automatizar todo el proceso que tiene asociado, conocer los hermanos que no les han tocado las medalla con anterioridad, y que tengan pendiente, a lo sumo, el pago de una cuota anual, y generar una lista con los números de hermanos que pueden entrar en sorteo.
13. El sorteo de medallas se realiza extrayendo de una bolsa una serie de papeletas que contienen los números de los hermanos que entran en sorteo, por lo que será necesario crearlas previamente. Sería necesario el poder generar, a partir de la lista de números de hermanos que entran en sorteo, un documento de texto (.pdf o .doc) que contenga dichos números distribuidos u organizado de forma que sólo se tuviesen que recortar para realizar el sorteo.
14. Los hermanos necesitan que se les avise de alguna manera y con cierta antelación de los distintos actos organizados por la Hermandad: correo electrónico, whatsapp, etc. Por ello, se tratará de registrar toda la información relativa a teléfono y redes sociales de cada hermano.
15. Sería muy útil que nuestra aplicación implementase la automatización de la elaboración de correspondencia (sobre+membrete de la carta) dentro del proceso de envío de correspondencia a la totalidad de hermanos a un conjunto determinado de ellos que podamos definir según necesidad.

134 ► Tablero de inspiración de diseños

Las palabras que me sugiere la idea del diseño que tengo para la aplicación serían:

- Completa
- Funcional
- Disponibilidad
- Organizada
- Atractiva
- Especializada
- Usable
- Especializada
- Disponible
- Accesible

Para el diseño de la aplicación he elegido la plantilla web **Kappe** que posee elementos estructurales e implementan funcionalidades similares a las de nuestro sistema a desarrollar, además de tener una forma de mostrar los distintos elementos y opciones disponibles bastante atractiva y dinámica. Esta plantilla la utilizaré de momento en modo de prueba, y se adquirirá si el producto finalizado cuenta con el visto bueno de la hermandad, con lo que haríamos ya un uso legal de la misma.

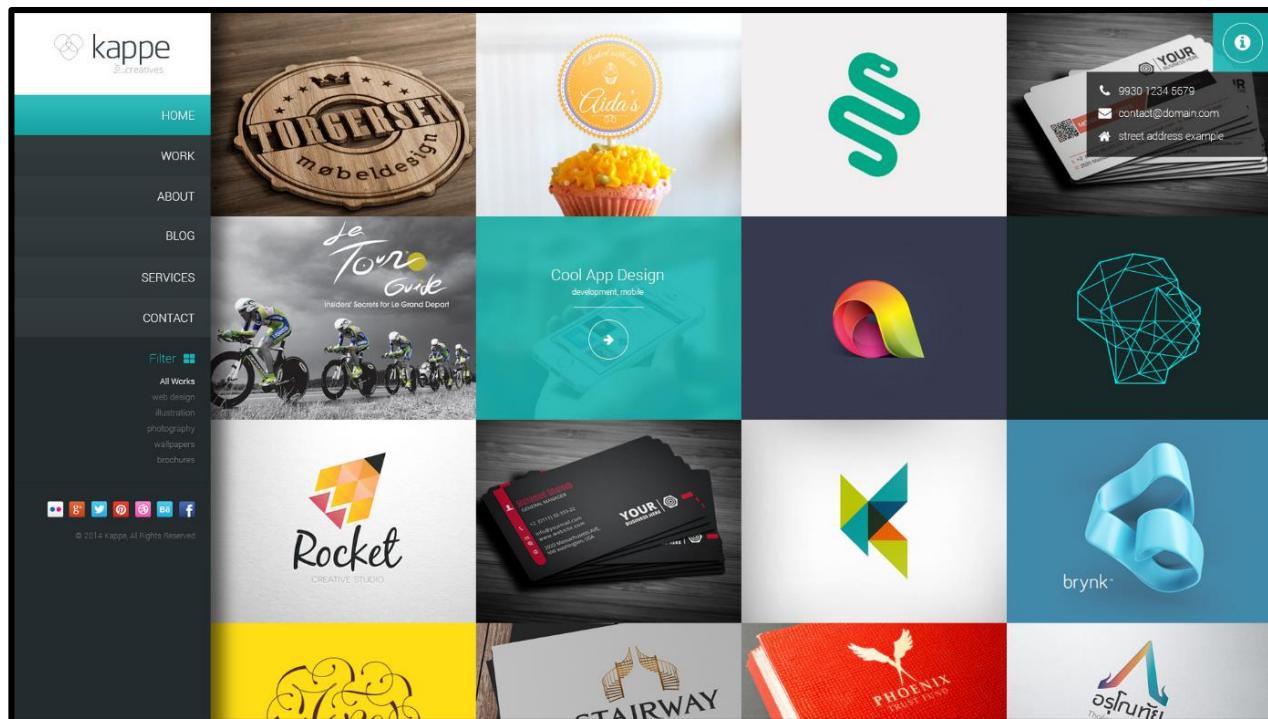


Figura 3. Plantilla Kappe. Página de Inicio

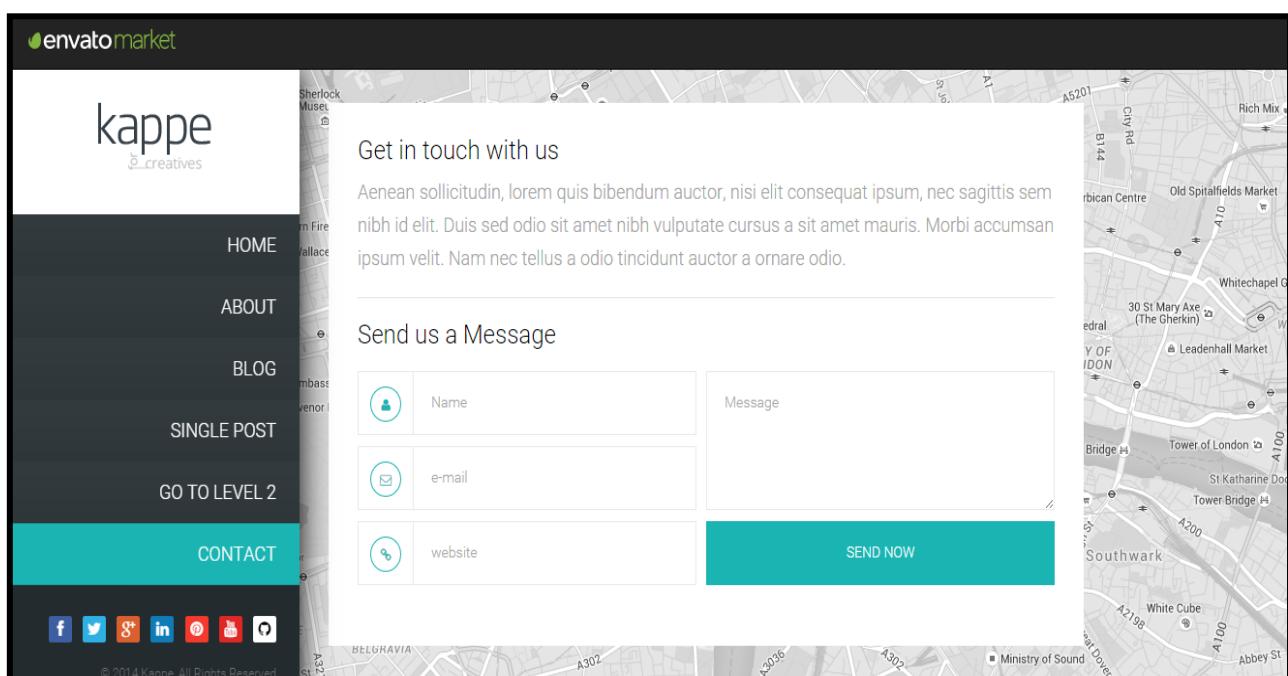

 A screenshot of the Kappe website's contact page. At the top, there is a header with the Envato Market logo. Below it, the main content area features a sidebar with links: HOME, ABOUT, BLOG, SINGLE POST, GO TO LEVEL 2, and CONTACT. The main content area has a heading "Get in touch with us" followed by a paragraph of placeholder text. To the right of the text is a map of London. Below the map is a form titled "Send us a Message" with fields for Name, e-mail, and website, along with a "SEND NOW" button. The footer contains a copyright notice: "© 2014 Kappe. All Rights Reserved" and social media icons for Facebook, Twitter, Google+, LinkedIn, Pinterest, and YouTube.

Figura 4. Plantilla Kappe. Página de Contacto

1.35 ► Estrategia de diseño y puntos de vista

- Las siguientes situaciones serían representativas de tareas que realiza la hermandad que podrían realizarse de manera más cómoda y eficiente con el uso de nuestra aplicación.

► STORYBOARDING

A Sorteo de medallas

La gestión del sorteo de las medallas de manera manual constituye una tarea muy laboriosa. En él, para que un hermano pueda entrar en sorteo, además de no haberle tocado con anterioridad la medalla, deberá estar al día en el pago de las cuotas o tener pendiente, a lo sumo, un año de pago, lo cual implica el tener que cotejar de manera manual cada uno de los pagos y si al hermano correspondiente le ha tocado o no ya la medalla, información que está recogida en el libro de registro de hermanos. Sólo una vez cotejada dichos datos podremos definir el conjunto de Hermano que entran en sorteo.

B ¡Adiós a los papeles!

La gestión de los hermanos de una Hermandad conlleva el manejo de diversos documentos (*fichas*) y datos asociados a cada uno de ellos que será necesario registrar para poder acceder y filtrar cuando y como sea necesario.

La realización de esta tarea de manera manual, mediante fichas de datos impresas en papel que se van actualizando convenientemente, plantea posibles problemas como el de la *pérdida de datos* si perdemos la ficha, la *poca disponibilidad* de la información, pues necesitaríamos del papel para tener acceso a ellos, o la necesidad de realizar tediosas tareas para determinar los hermanos que entran en sorteo.

Para que el usuario pueda evitar el tener que manejar tantos documentos y tener accesibles los datos en todo momento resultaría de mucha utilidad una aplicación móvil, preferentemente sobre *smartphones* por ser los más utilizados, que fuese personalizable para cada usuario, que permita controlar de manera fácil y centralizada todos los aspectos que conlleva la realización de una dieta.

El usuario deberá poder acceder a los datos de los hermanos registrados clasificados de distintas maneras, manejar una agenda de contactos de la Hermandad de manera rápida y eficiente así como permitir que se impriman o se exporten a algún formato estándar como PDF.

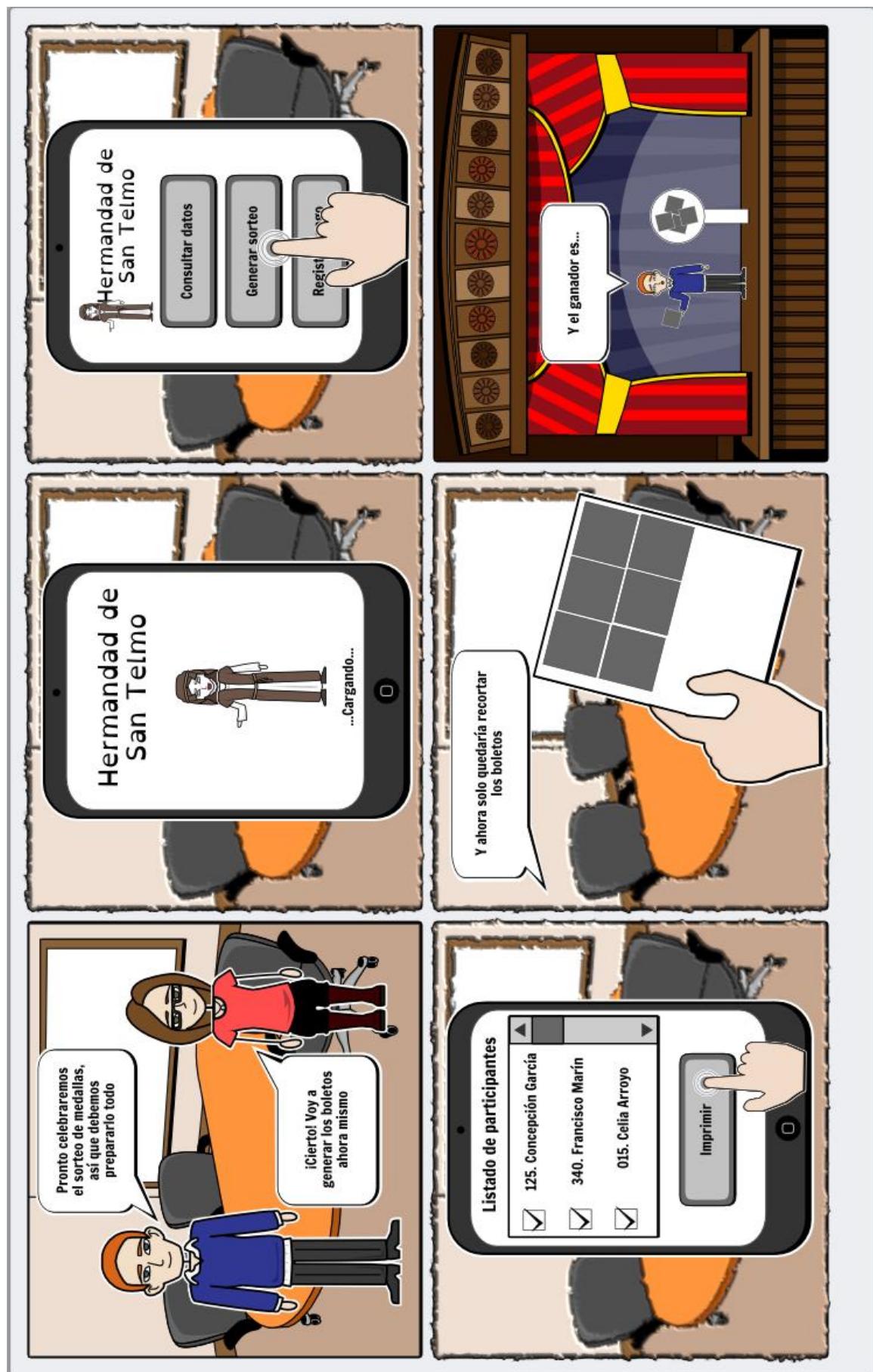


Figura 5. Storyboard: “Sorteo de medallas”

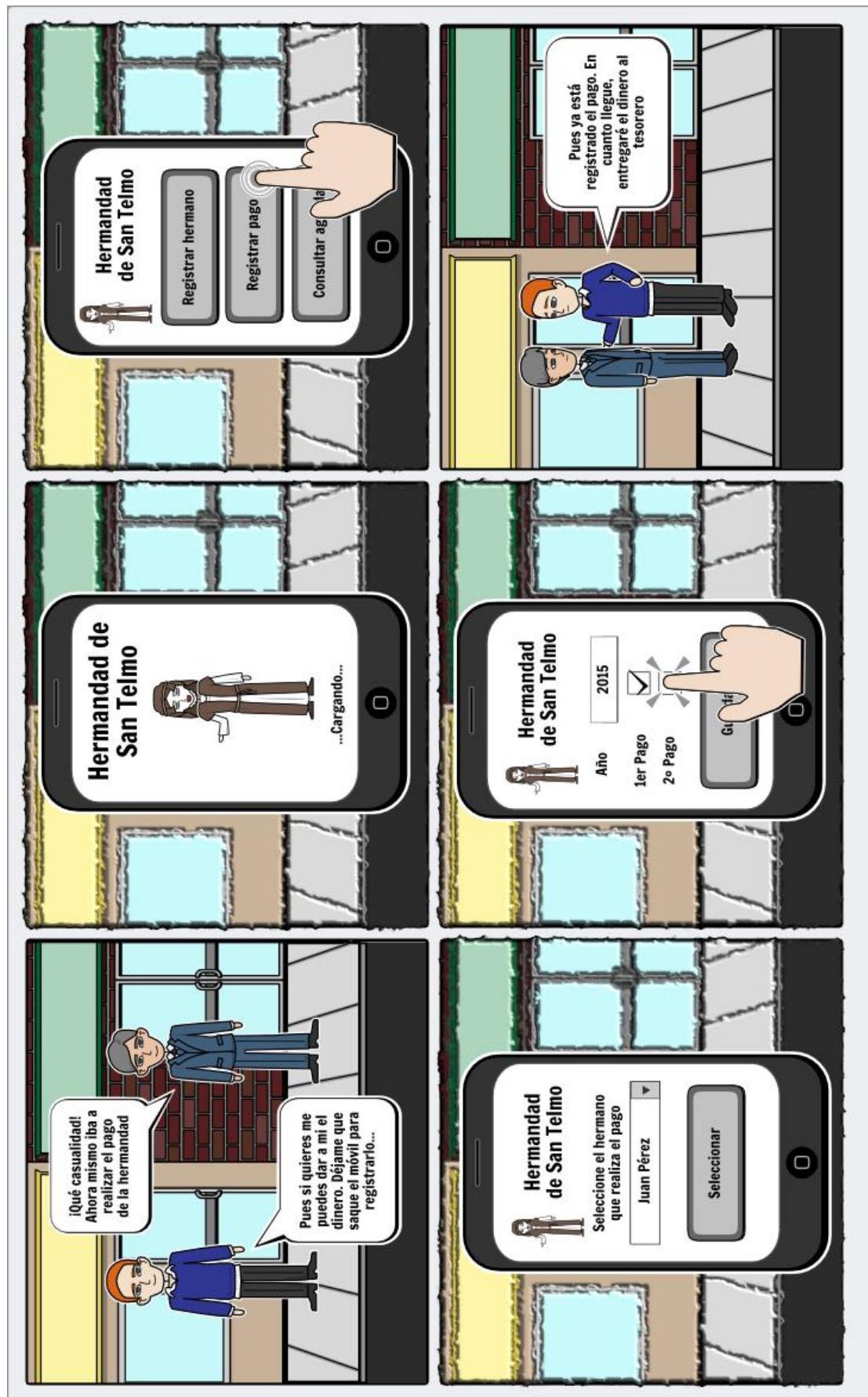


Figura 6. Storyboard: "Adiós a los papeles"

1.3.6 ► Requisitos físicos

La utilización de la aplicación podrá ser *fija* o *móvil*. La *fija*, será la que se realice sobre el equipo de escritorio con acceso a internet que posee la Hermandad en la actualidad, en la que nuestra aplicación podrá trabajar en modo local como sistema de respaldo ante situaciones de imposibilidad de acceso al sitio que aloja la aplicación o, normalmente, accediendo a la web de la aplicación. Las tareas que se realizarán mayoritariamente en este caso serían las relacionadas con la gestión de información: hermanos, contactos, etc. Por otro lado, tendríamos el uso móvil de la aplicación que básicamente estará asociado a la tarea de cobro de las cuotas, en la que será necesario ir visitando las distintas viviendas de la localidad.

Por ello, para poder hacer uso de nuestra aplicación en ambos modos será necesario disponer de un equipo sobremesa en el que se instalará la aplicación junto al servidor web y demás recursos software necesarios para que la aplicación se pueda ejecutar correctamente, así como de algún dispositivo móvil, preferentemente una *tablet*, con acceso a internet desde la cual se pueda acceder a la aplicación que estará disponible en algún sitio web de hostings que la Hermandad haya contratado, siendo, obviamente necesario, mantener sincronizadas de algún modo la información que se vaya actualizando en la base de datos.

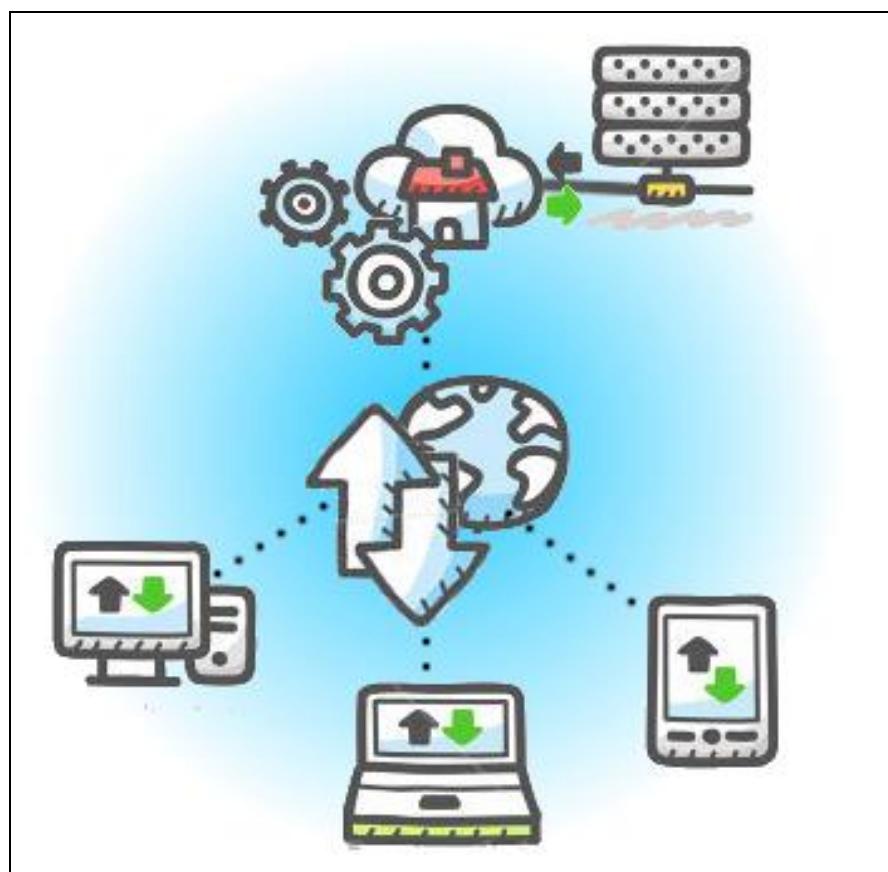


Figura 7. Configuración física de acceso a la aplicación

1.37 ▶ Requisitos lógicos/funcionales

Los requisitos funcionales de una aplicación definen los comportamientos del sistema y constituyen las especificaciones que definen las distintas funcionalidades que implementa un sistema software. Los requerimientos funcionales pueden ser: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone, un sistema debe cumplir y que son revisados y aprobados por el cliente.. Los requerimientos de comportamiento para cada requerimiento funcional serán los que se muestran en los casos de uso.

► Tipos de usuarios

Los *tipos de usuario* que existen para nuestra aplicación son dos:

- **Administrador.** Usuario que tendrá todos los privilegios y, por tanto, acceso a todas las funcionalidades que implementa la aplicación.



- **Usuario.** Usuario que no podrá actualizar la base de datos y sólo tendrá acceso de lectura a la información de la aplicación.



► Bloques funcionales

Los *bloques funcionales* que se incluyen en la aplicación estarán asociados, cada uno de ellos, a las distintas categorías de datos que se en ella se maneja. En nuestro caso podremos definir los siguientes bloques:

- **Viviendas.** Gestión del registro de las distintas viviendas donde los hermanos están asociados para la realización de los pagos de sus cuotas o recibir comunicaciones.
- **Hermanos.** Gestión de los Hermanos de la Hermandad.
- **Remesas.** Gestión de las distintas remesas o pagos que se definen.
- **Consultas.** Permite recuperar información de la base de datos según un criterio de búsqueda preestablecido.
- **Pagos.** Gestión del abono de los hermanos de los distintos pagos asociados a las distintas remesas.
- **Agenda de Contactos.** Gestión de la información de contacto de personas y empresas que prestan o pueden prestar algún tipo de servicio a la hermandad.
- **Sorteo de medallas.** Generación lista de papeletas del sorteo.

Vamos a listar a continuación los distintos bloques funcionales incluidos en la aplicación junto a sus tareas y la accesibilidad de los usuarios a dichas tareas.



Gestión de Viviendas

Nº	TAREA	USUARIO		OBSERVACIONES
		ADMIN	USER	
1	Añadir Vivienda	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2	Editar Vivienda	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	Listar Viviendas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	



Gestión de Hermanos

Nº	TAREA	USUARIO		OBSERVACIONES
		ADMIN	USER	
1	Añadir Hermano	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2	Editar Hermano	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	Eliminar Hermano	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	Listar Hermano	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	



Gestión de Remesas

Nº	TAREA	USUARIO		OBSERVACIONES
		ADMIN	USER	
1	Añadir Remesa	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
2	Editar Remesa	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	Eliminar Remesa	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	Listar Remesa	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	



Gestión de Consultas

Nº	TAREA	USUARIO		OBSERVACIONES
		ADMIN	USER	
1	Añadir Consulta	<input checked="" type="checkbox"/>	✗	
2	Editar Consulta	<input checked="" type="checkbox"/>	✗	
3	Eliminar Consulta	<input checked="" type="checkbox"/>	✗	
4	Listar Consultas	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	



Gestión de Pagos

Nº	TAREA	USUARIO		OBSERVACIONES
		ADMIN	USER	
1	Añadir Pago	<input checked="" type="checkbox"/>	✗	
2	Editar Pago	<input checked="" type="checkbox"/>	✗	
3	Eliminar Pago	<input checked="" type="checkbox"/>	✗	
4	Listar Hnos con impagos	<input checked="" type="checkbox"/>	✗	
5	Consultar Pagos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	



Gestión de Agenda de Contactos

Nº	TAREA	USUARIO		OBSERVACIONES
		ADMIN	USER	
1	Añadir Contacto	<input checked="" type="checkbox"/>	✗	
2	Editar Contacto	<input checked="" type="checkbox"/>	✗	
3	Eliminar Contacto	<input checked="" type="checkbox"/>	✗	
4	Listar Contacto	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	



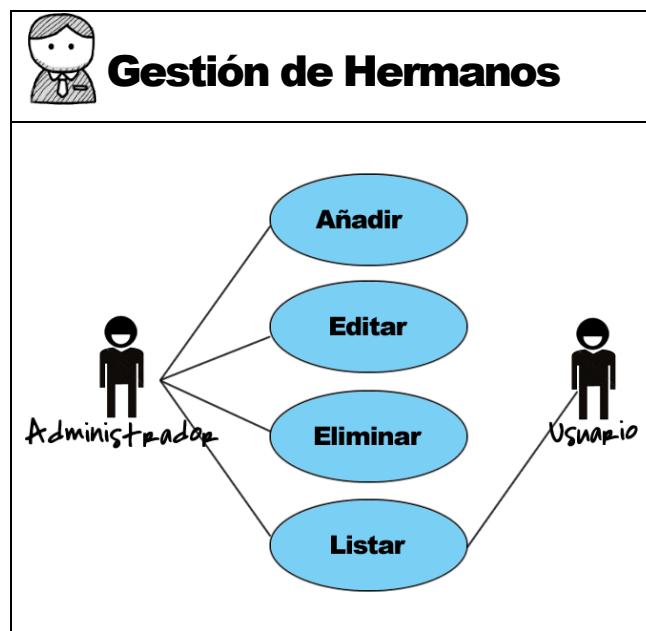
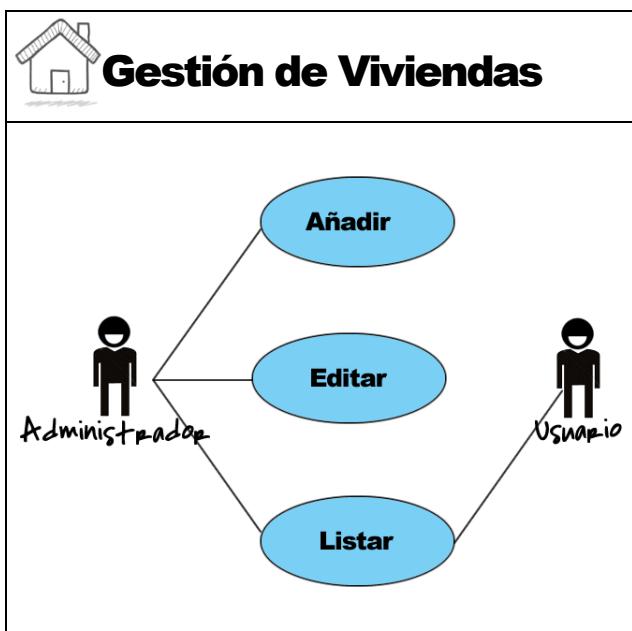
Gestión de Sorteo de Medallas

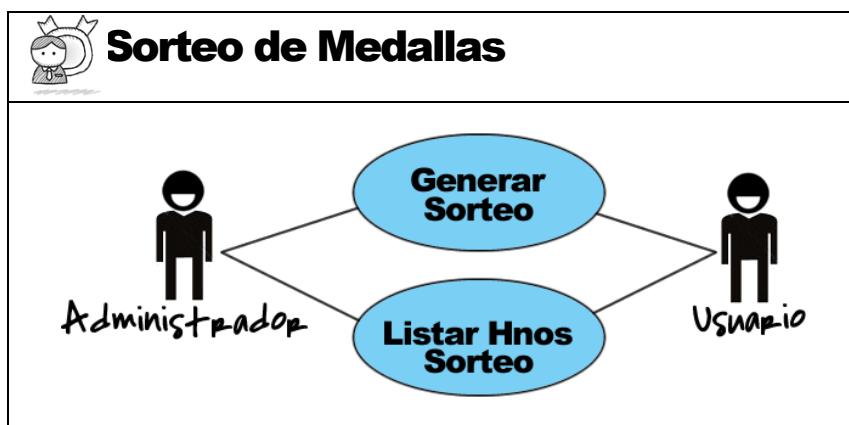
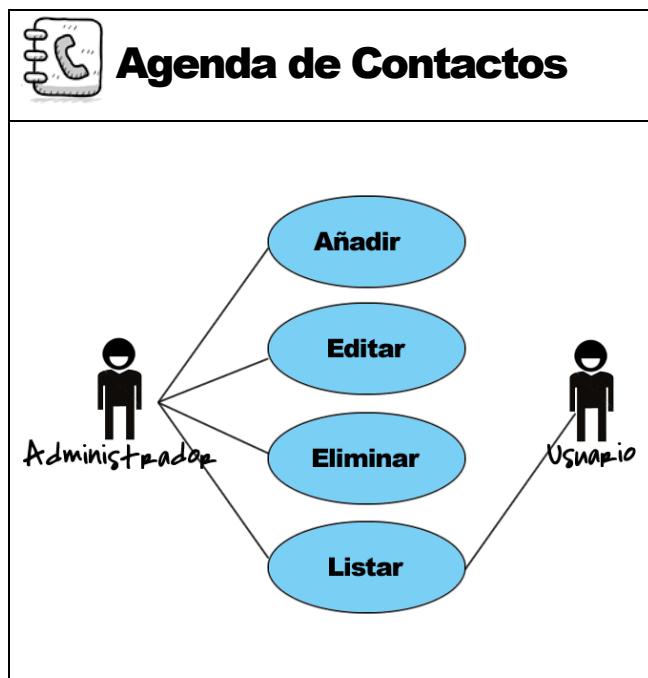
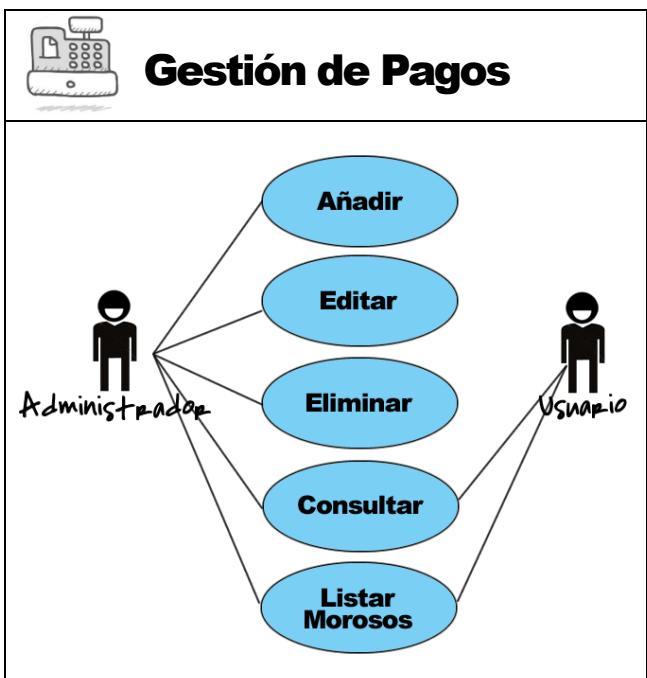
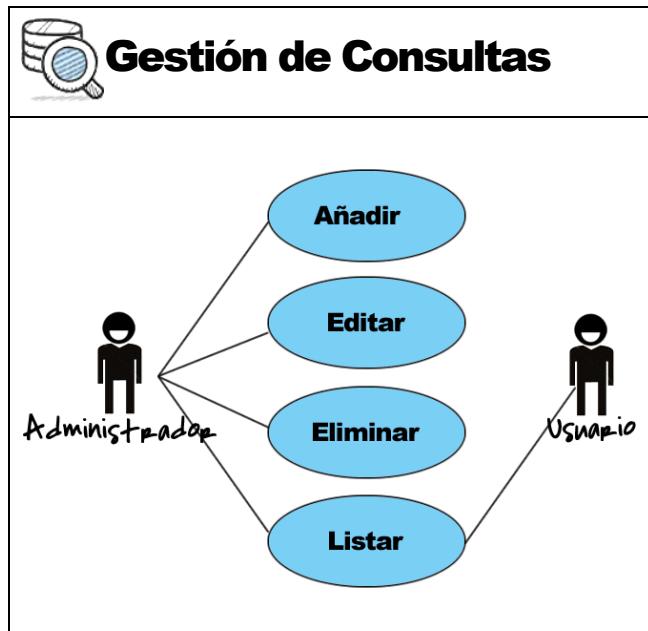
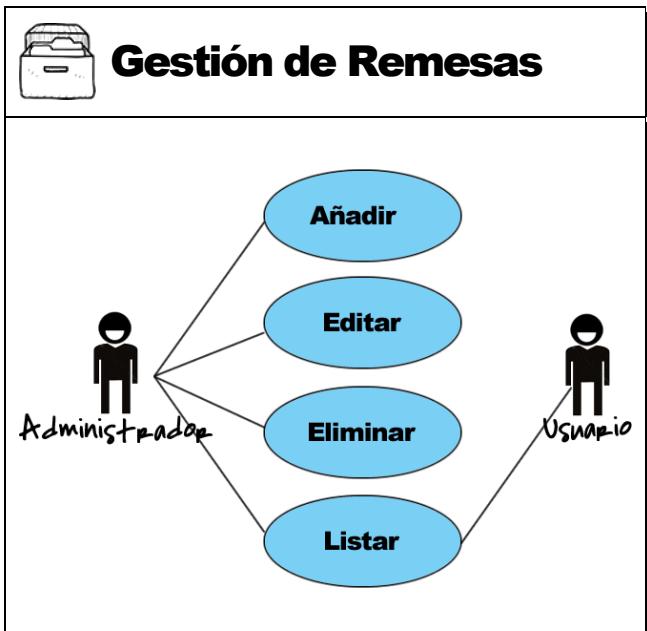
Nº	TAREA	USUARIO		OBSERVACIONES
		ADMIN	USER	
1	Generar Sorteo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	Listar Hermanos Sorteo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

1.3.8 ▶ Diagramas de casos de uso

Los *diagramas de casos de uso* se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo, siendo los más comunes los empleados para la captura de requisitos funcionales.

En el contexto de ingeniería del software, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas o, dicho de otra forma, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema.





1.39 ► Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que nuestro sistema debe poseer. El conjunto de estos requisitos los podemos dividir en diferentes áreas:

→ **Disponibilidad**

Se realizará un sistema Web, con lo que los usuarios se podrán conectar desde cualquier parte del mundo. Este sistema deberá de poder funcionar desde los navegadores más utilizados por los usuarios.

→ **Seguridad.**

La información estará protegida tanto para visualizarlos como para actualizarlos. Cada usuario contará con un perfil, que le permitirá realizar sólo las operaciones asociadas a dicho perfil.

→ **Usabilidad.**

Se utilizará un diseño de las páginas del sitio web intuitivo, con una navegación clara, que permita al usuario realizar sus tareas de manera productiva y satisfactoria.

→ **Rendimiento.**

Será una aplicación Web condicionada por diversos factores como la velocidad de acceso a Internet aunque, dado el pequeño volumen de datos a manejar, a priori éste no será un aspecto problemático.

→ **Coste.**

Las herramientas y tecnologías utilizadas para el desarrollo del sistema son gratuitas y la plantilla utilizada se tendrá a modo de prueba temporal.

→ **Mantenibilidad.**

Se utilizará una arquitectura *Model-View-Controller* para poder separar el interfaz del usuario, controladores con la lógica del negocio y módulos que transparentemente se conectan a la base de datos. Esto facilitará el análisis de nuevas funcionalidades, los cambios en un futuro y las pruebas.

13.10 ► Prototipo de la aplicación



Pantalla de inicio

www.gestst.org

Viviendas Hermanos Remesas Consultas Pagos Agenda Sorteo ...

Viviendas
Hermanos
Remesas
Consultas
Pagos
Agenda
Sorteo
XXXXX



Gestión de Viviendas

www.gestst.org

Viviendas Hermanos Remesas Consultas Pagos Agenda Sorteo ...

Vivienda

Barriada	Barriada
Línea	Línea
Número	Número

Observaciones

Lore ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



Gestión de Remesas

www.gestst.org

Viviendas Hermanos Remesas Consultas Pagos Agenda Sorteo ...

Remesa

Año

Descripción

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.







Gestión de Pagos

www.gestst.org

Viviendas Hermanos Remesas Consultas Pagos Agenda Sorteo ...

Pago

Año Remesa

Cuota 1	<input type="text" value="14/06/2015"/>	
Cuota 1	<input type="text" value="14/06/2015"/>	

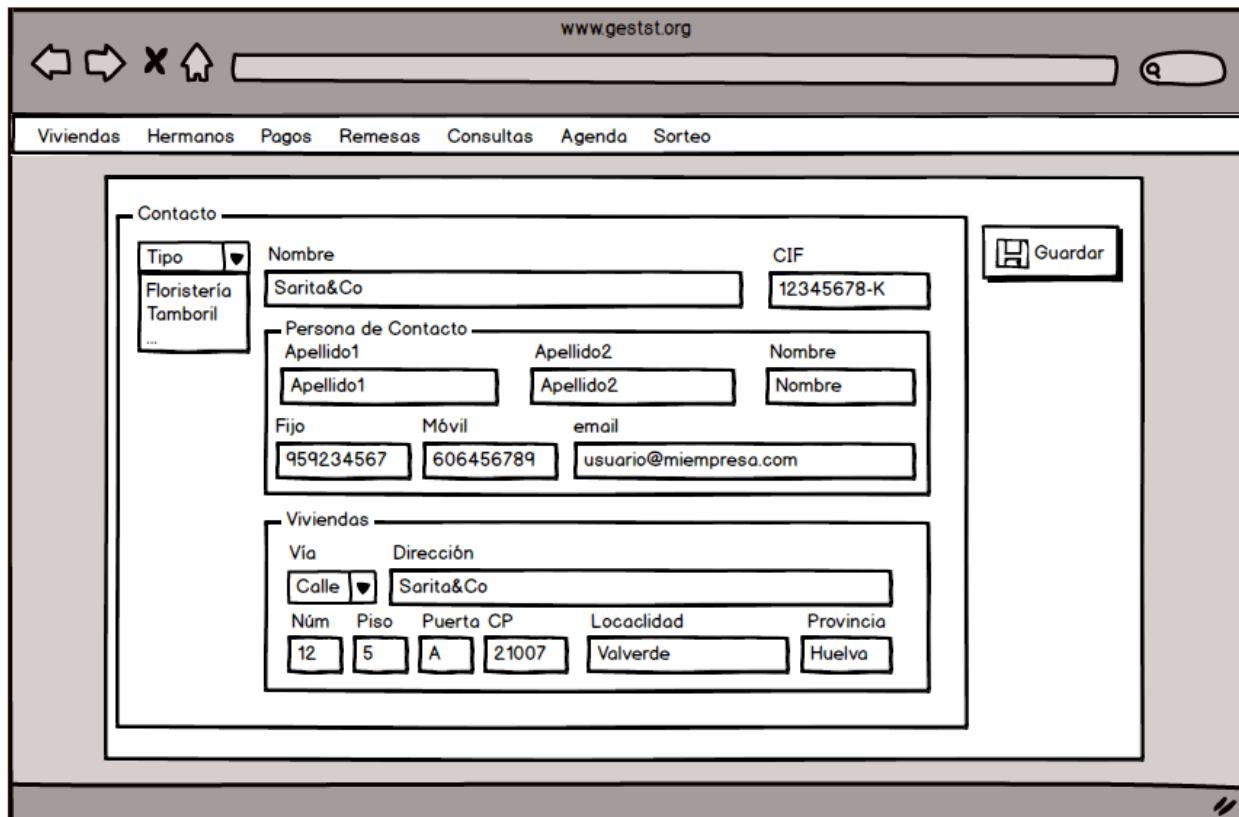






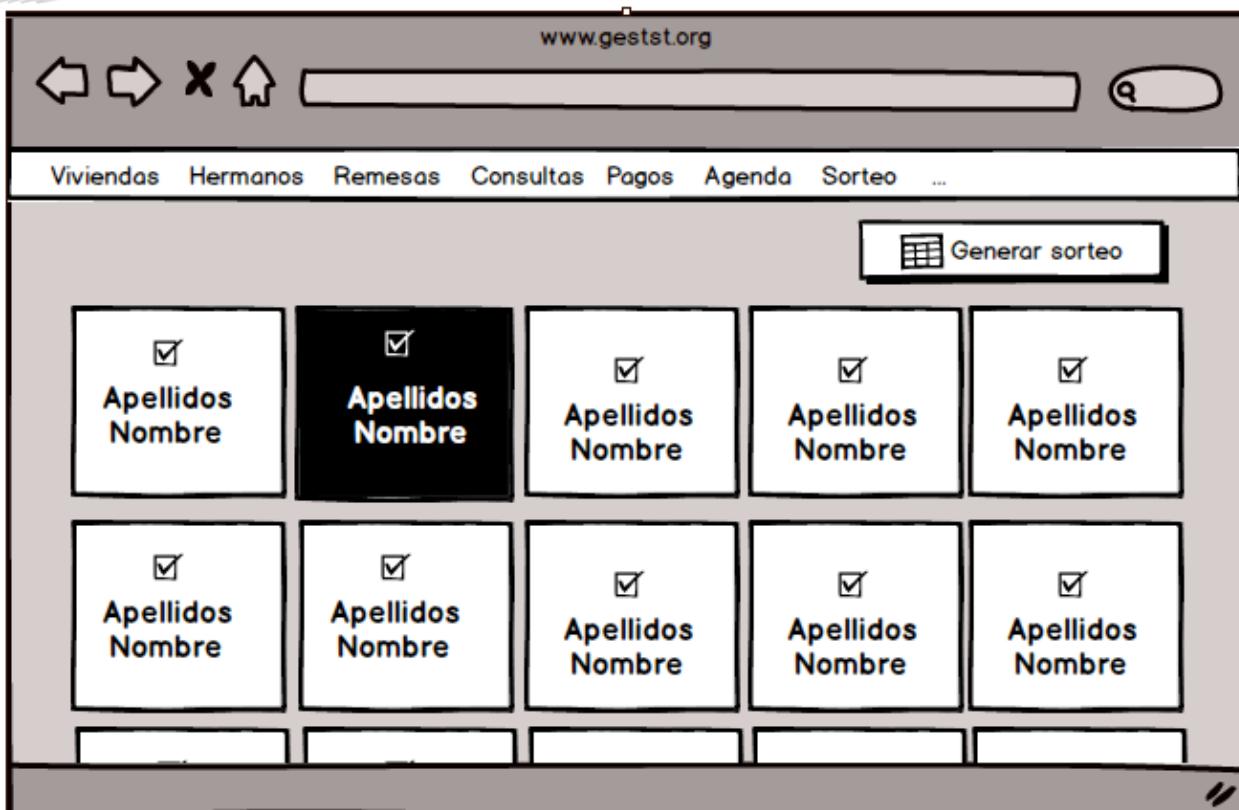
Gestión de Agenda de Contactos

www.gestst.org




Gestión de Sorteo de Medallas

www.gestst.org

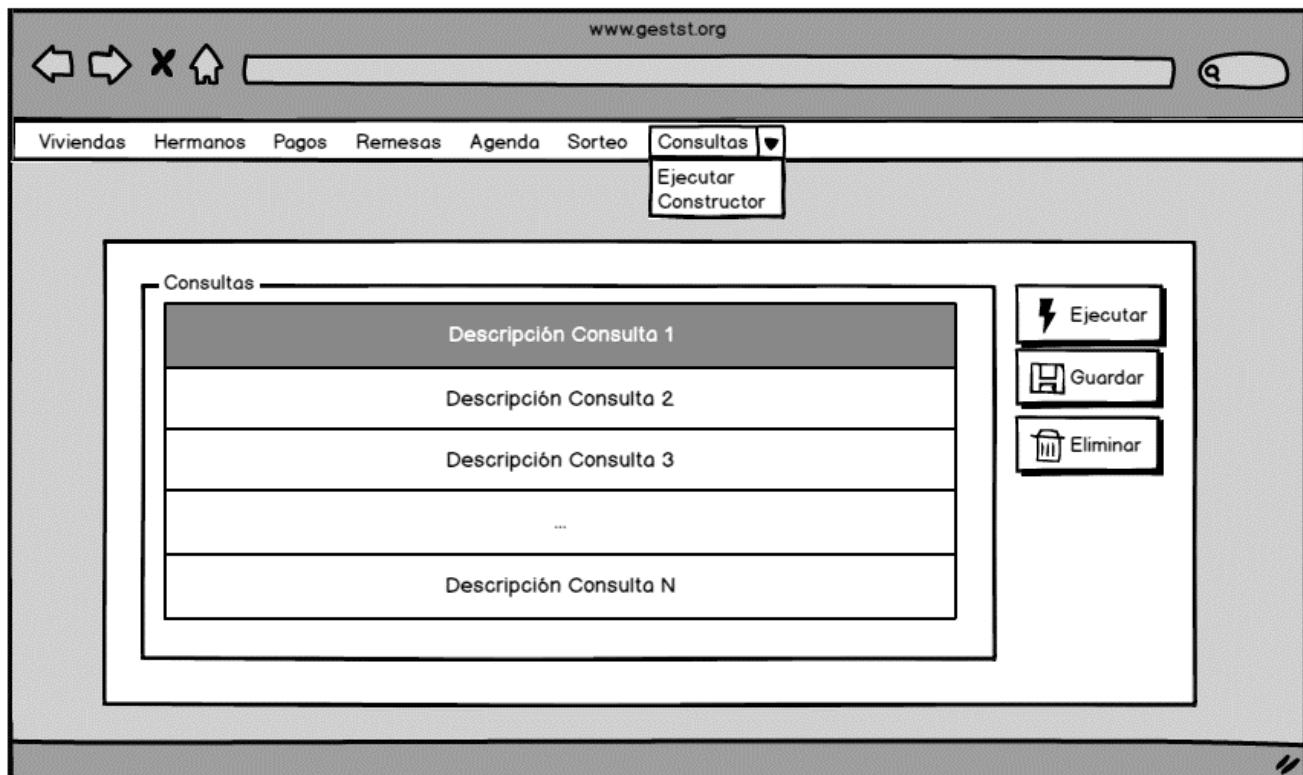




Gestión de Consultas

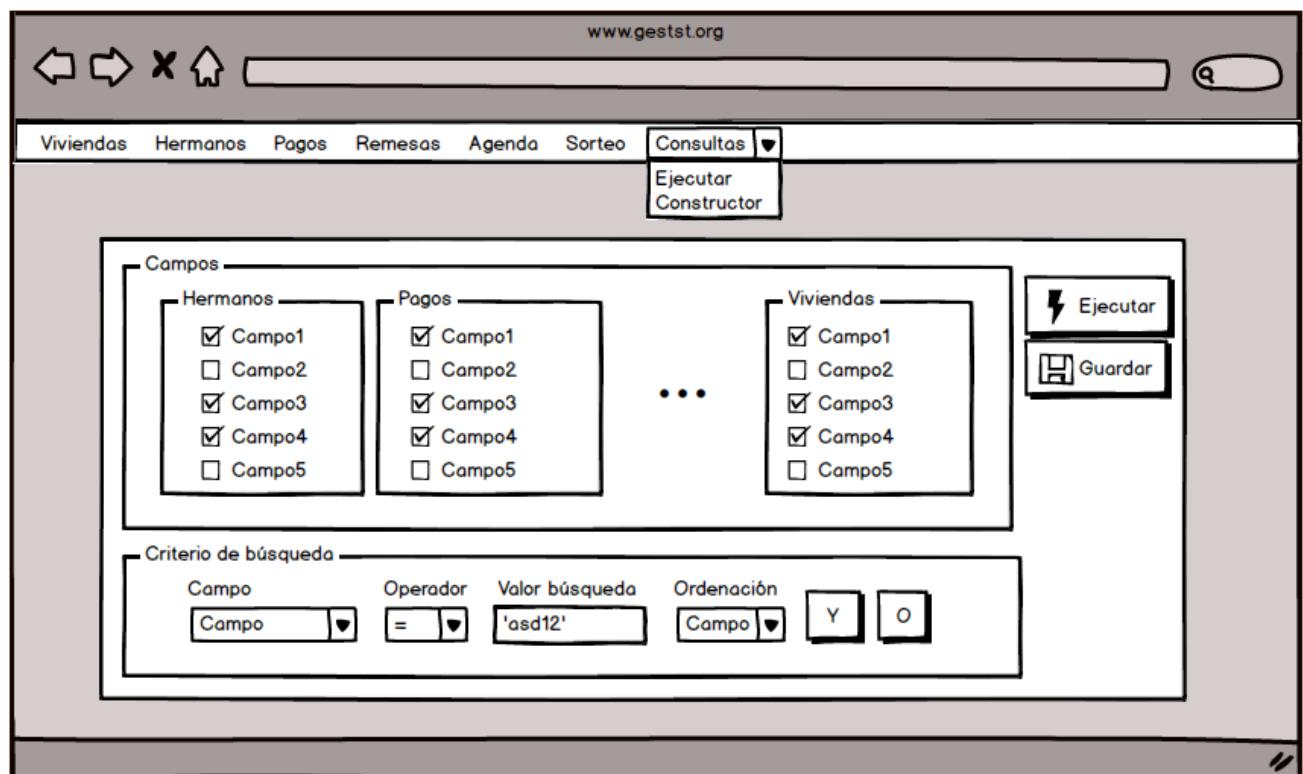
▶ Consultas almacenadas

www.gestst.org



▶ Constructor de Consultas

www.gestst.org





Gestión de Hermanos

Datos Personales		Num. Hno.	Familia			
	Tratamiento*:	Señor <input type="button" value="▼"/>				
	Nombre*:	<input type="text"/>				
	Primer apellido*:	<input type="text"/>				
	Segundo apellido:	<input type="text"/>				
	Fecha de nacimiento*:	1 <input type="button" value="▼"/> Enero <input type="button" value="▼"/>	<input type="text"/>			
	Documento de identidad*:	NIF <input type="button" value="▼"/>	<input type="text"/>			
	Tipo vía*:	Calle <input type="button" value="▼"/>				
	Dirección*:	<input type="text"/>				
	Número:	<input type="text"/>				
	Piso:	<input type="text"/>				
	Puerta:	<input type="text"/>				
	Código Postal*:	<input type="text"/>				
	Ciudad*:	<input type="text"/>				
	Provincia*:	Alava <input type="button" value="▼"/>				
	Teléfonos de contacto*:	<input type="text"/>	<input type="text"/>			
	Móvil	<input type="text"/>	<input type="text"/>			
	Email*:	<input type="text"/>				
Redes Sociales						
	Twitter	<input type="text"/>	<input type="text"/>			
	Facebook	<input type="text"/>	<input type="text"/>			
	Instagram	<input type="text"/>	<input type="text"/>			
	Entidad	<input type="text"/>				
	ES 13 2090 0000 29 0350000083					
	CÓDIGO PAÍS	DIGITO CONTROL	ENTIDAD	OFICINA	DIGITO CONTROL	Nº DE CUENTA
	"IBAN"					

13.11 ▶ Mapa de navegación de la aplicación

Nuestra aplicación va a definir un sitio Web que consta de un conjunto de páginas web que conforman el sistema completo de pantallas que muestran contenidos o que permiten realizar las tareas que nuestra web implementa.

El siguiente diagrama muestra el *Mapa de Navegación* de la aplicación que proporciona una representación esquemática de la estructura del sitio, indicando los enlaces entre las páginas y las interrelaciones que existen entre ellas. Será pues una vista Web que muestra cómo los usuarios del sitio lo navegarán, representado en un diagrama de *árbol jerárquico* en el que cada nivel del diagrama muestra el número de clicks necesarios para acceder a una determinada página del sitio desde otra cualquiera. Este tipo de diagrama a menudo es mencionado "un *mapa de sitio*".

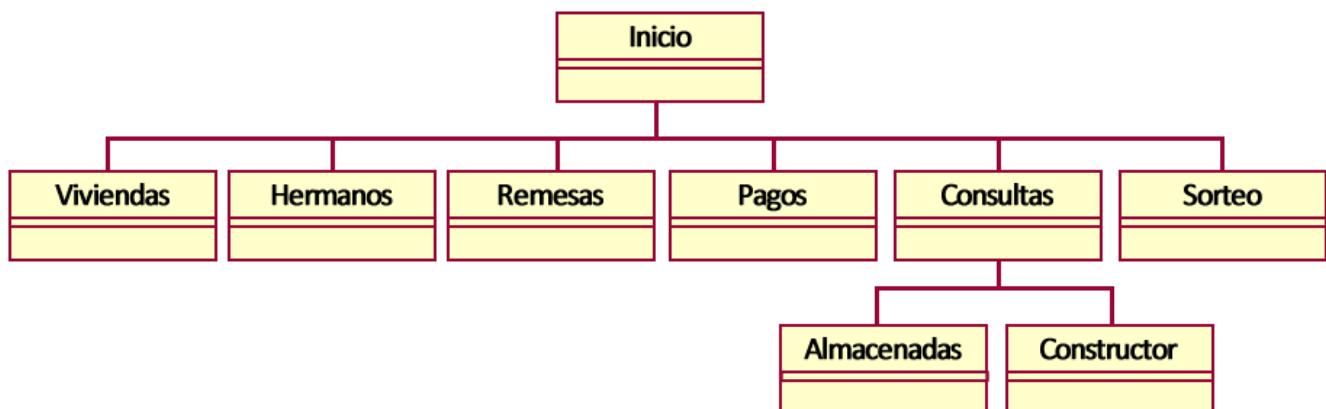


Figura 8. Mapa de Navegación de la aplicación

A continuación se muestran los mapas de navegación para los distintos actores que utilizan la aplicación junto a los permisos de acceso asociados a cada uno de ellos:

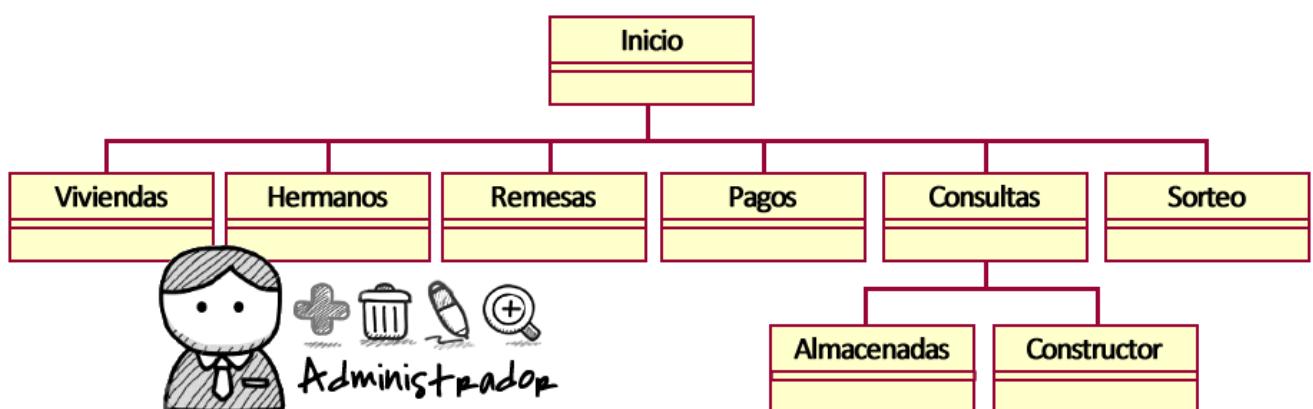


Figura 9. Mapa de Navegación de la aplicación para el actor Administrador

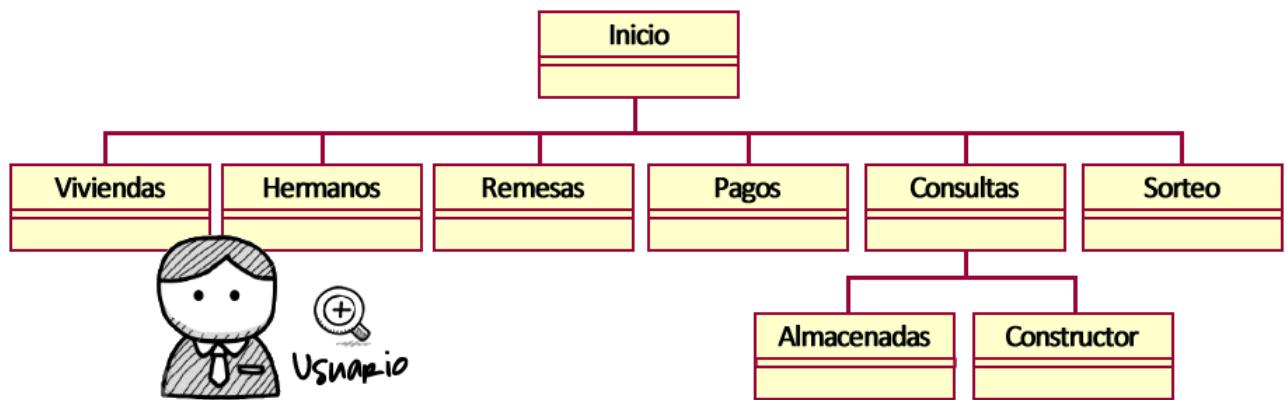


Figura 10. Mapa de Navegación de la aplicación para el actor Administrador

2 ► Modelo de datos

El modelo de datos es un recurso orientado a hablar de una Base de Datos que permite describir:

- Las estructuras de datos de la base: El tipo de los datos que hay en la base y la forma en que se relacionan.
- Las restricciones de integridad: Un conjunto de condiciones que deben cumplir los datos para reflejar correctamente la realidad deseada.
- Operaciones de manipulación de los datos: típicamente, operaciones de agregado, borrado, modificación y recuperación de los datos de la base.

21 ► Diagrama Entidad/Relación

En el modelo de datos, los *diagramas entidad/relación* son un método del que disponemos para diseñar estos esquemas que posteriormente debemos de implementar en un gestor de BBDD (bases de datos). Estos diagramas están formados por distintos elementos que ayudan a entender los datos y cómo se relacionan entre ellos. La siguiente figura muestra el diagrama Entidad/Relación inicial de la aplicación:

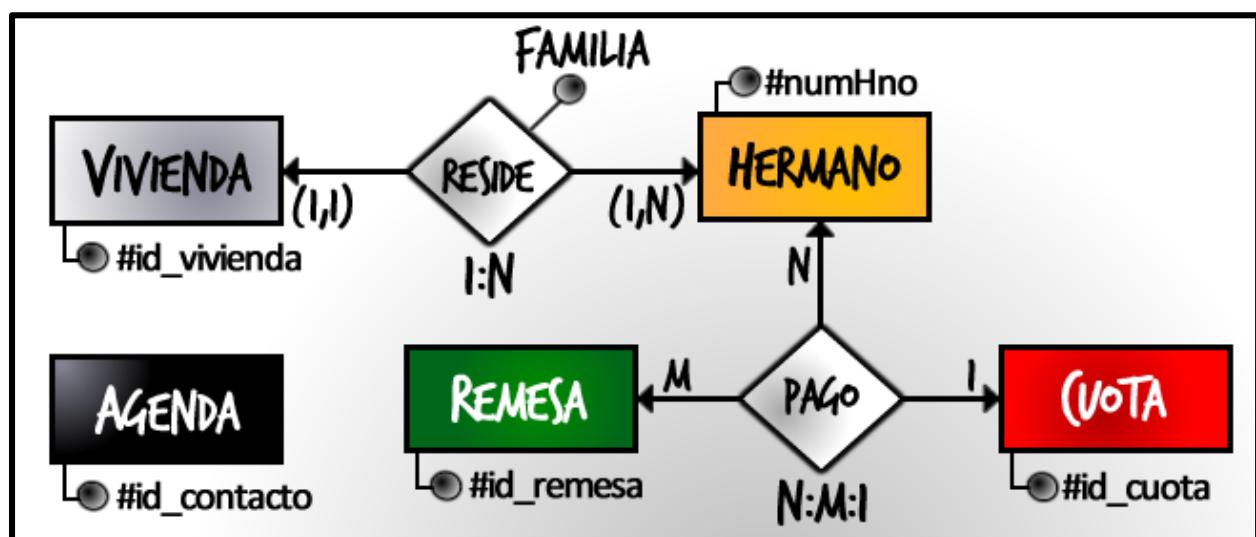


Figura 11. Diagrama Entidad/Relación de la aplicación

22 Esquema de la base de datos (MySQLWorkbench)

Las **bases de datos** son parte fundamental de toda aplicación ya que nos permiten almacenar y usar de forma rápida y eficiente cantidades ingentes de datos con cierta facilidad. Para la definición del modelo de datos de mi aplicación he utilizado el modelo de **bases de datos relacionales** y para su implementación he utilizado la aplicación MySQLWorkbench. A continuación se muestran las tablas que incluye la base de datos y las relaciones que entre ellas se establecen.

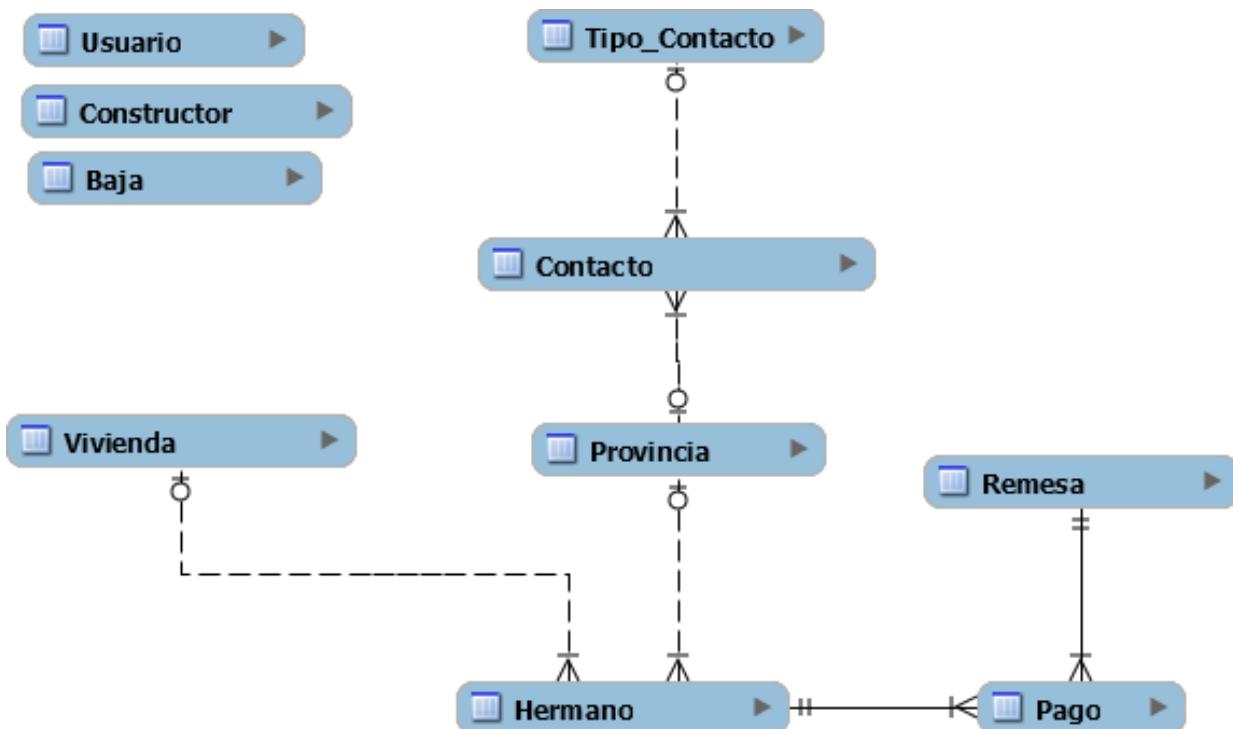
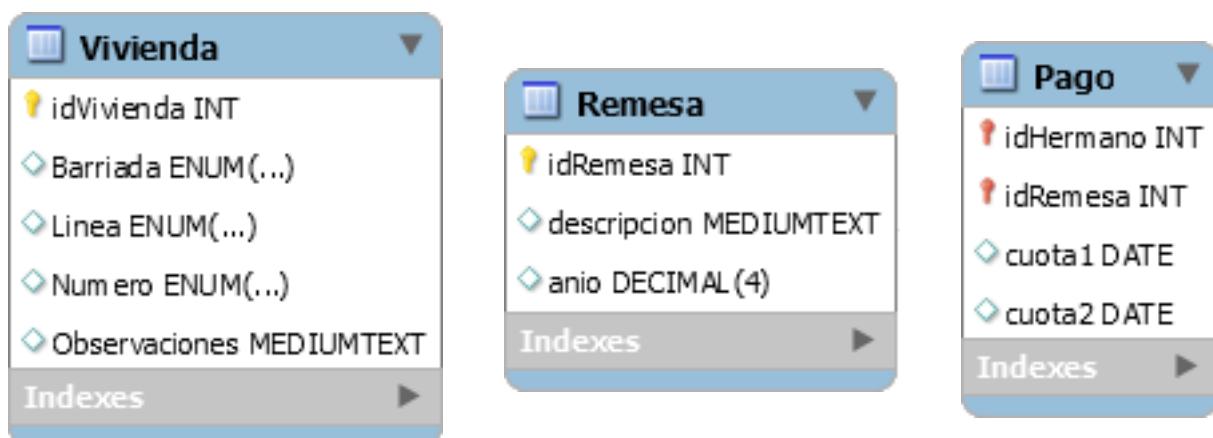


Figura 12. Esquema de la base de datos de la aplicación

Descripción de las Tablas

A continuación se detallan cada una de las tablas y los campos que cada una de ellas contiene así como los campos que constituyen la clave primaria de cada una.



Hermano	
!	idHermano INT
◇	vivienda INT
◇	tratamiento ENUM(...)
◇	nombre VARCHAR(50)
◇	apellido1 VARCHAR(50)
◇	apellido2 VARCHAR(50)
◇	fecha_nacimiento DATE
◇	dni VARCHAR(20)
◇	tipo_via ENUM(...)
◇	direccion VARCHAR(50)
◇	numero DECIMAL(4)
◇	piso DECIMAL(2)
◇	puerta CHAR(1)
◇	codigo_postal CHAR(5)
◇	poblacion VARCHAR(50)
◇	movil VARCHAR(15)
◇	fijo VARCHAR(15)
◇	email VARCHAR(100)
◇	twitter VARCHAR(100)
◇	facebook VARCHAR(100)
◇	instagram VARCHAR(100)
◇	tipo ENUM(...)
◇	cuenta_corriente CHAR(24)
◇	familia ENUM(...)
◇	provincia INT
◇	medalla TINYINT(1)
Indexes	

Contacto	
!	idContacto INT
◇	nombre_empresa VARCHAR(100)
◇	tratamiento ENUM(...)
◇	nombre VARCHAR(50)
◇	apellido1 VARCHAR(50)
◇	apellido2 VARCHAR(50)
◇	cif VARCHAR(20)
◇	tipo_via ENUM(...)
◇	direccion VARCHAR(50)
◇	numero DECIMAL(4)
◇	piso DECIMAL(2)
◇	puerta CHAR(1)
◇	codigo_postal CHAR(5)
◇	poblacion VARCHAR(50)
◇	movil VARCHAR(15)
◇	fijo VARCHAR(15)
◇	email VARCHAR(100)
◇	twitter VARCHAR(100)
◇	facebook VARCHAR(100)
◇	instagram VARCHAR(100)
◇	tipo INT
◇	provincia INT
Indexes	

Tipo_Contacto	
!	idTipo_Contacto INT
◇	tipo VARCHAR(100)
Indexes	

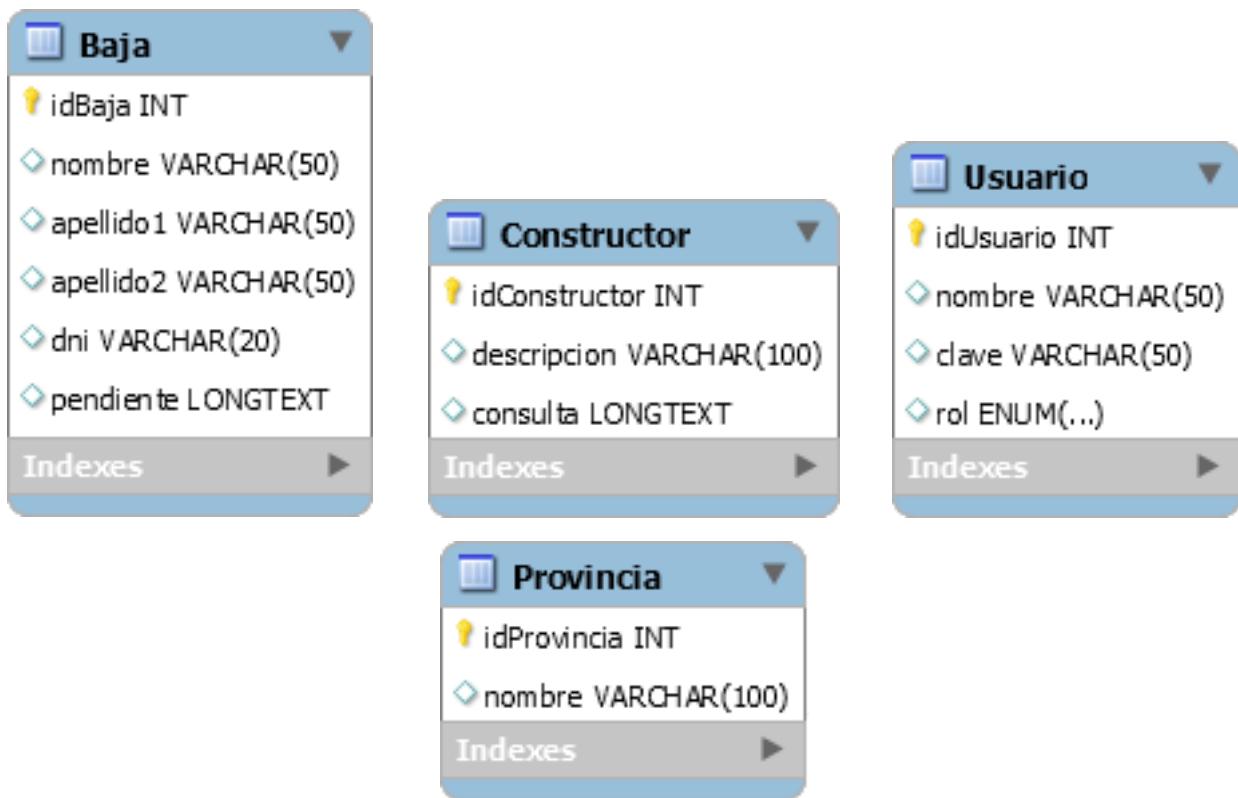
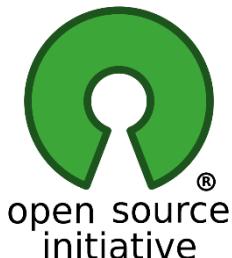


Figura 13. Descripción de las tablas de la base de datos de la aplicación

3► Tecnologías utilizadas en el desarrollo del proyecto

3.1► Por qué usar tecnologías de Software Libre



El movimiento *Open Source* comenzó como una oposición consciente a la mentalidad de la propiedad corporativa. Sus comienzos pueden encontrarse en el **MIT** (*Massachusetts Institute of Technology*), en los años 70, cuando Richard Stallman comenzó un movimiento para compartir código entre todos los programadores, y señaló la necesidad de una comunidad de desarrolladores en constante colaboración.

Pronto salieron a relucir las ventajas de este movimiento, el software libre cuesta poco o nada de usar y es especialmente atractivo para ONGs, universidades y organizaciones cuyos presupuestos son variables. Por otra parte el software open source puede ser modificado para su adaptación a cada necesidad específica, es más robusto (probado), que el software comercial, más fiable y seguro. Este tipo de tecnologías tienen cada vez más aceptación en la industria de internet española. Si bien es cierto que solamente un “21% de las grandes empresas apuesta por el software libre, constituye un 33% entre las pymes españolas” -según el Estudio sobre el estado del arte del Software de Fuentes Abiertas en la empresa española. 2015.

Entre las principales ventajas encontradas entre las empresas que apostaron por el software libre se encuentran la reducción de costes por licencias y coste total de propiedad, seguidas por el acceso al código fuente y una elevada valoración de los estándares y procesos de desarrollo abierto.

Aunque algunas empresas no encuentran en las Administraciones el ejemplo Open Source que se da en otros estados más avanzados que el nuestro, otras afirman que las empresas y administraciones empiezan a ser conscientes de las ventajas que ofrece un modelo de desarrollo empresarial sostenible como el ofrecido por las tecnologías de software libre basado en la cooperación, innovación y transferencia de conocimientos entre otros valores.

3.2► Parte CLIENTE

3.2.1► HTML



HTML, siglas de *HyperText Markup Language* («lenguaje de marcas de hipertexto»), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, entre otros. Es un estándar a cargo de la W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación.

El lenguaje HTML basa su filosofía de desarrollo en la referenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, entre otros.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante

texto. De este modo, la página web contiene sólo texto que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

Sin embargo, a lo largo de sus diferentes versiones, se han incorporado y suprimido diversas características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (PC de escritorio, portátiles, teléfonos inteligentes, tabletas, etc.). Sin embargo, para interpretar correctamente una nueva versión de HTML, los desarrolladores de navegadores web deben incorporar estos cambios y el usuario debe ser capaz de usar la nueva versión del navegador con los cambios incorporados. Normalmente los cambios son aplicados mediante parches de actualización automática (Firefox, Chrome) u ofreciendo una nueva versión del navegador con todos los cambios incorporados, en un sitio web de descarga oficial (Internet Explorer). Un navegador no actualizado no será capaz de interpretar correctamente una página web escrita en una versión de HTML superior a la que pueda interpretar, lo que obliga muchas veces a los desarrolladores a aplicar técnicas y cambios que permitan corregir problemas de visualización e incluso de interpretación de código HTML. Así mismo, las páginas escritas en una versión anterior de HTML deberían ser actualizadas o reescritas, lo que no siempre se cumple. Es por ello que ciertos navegadores aún mantienen la capacidad de interpretar páginas web de versiones HTML anteriores. Por estas razones, aún existen diferencias entre distintos navegadores y versiones al interpretar una misma página web.

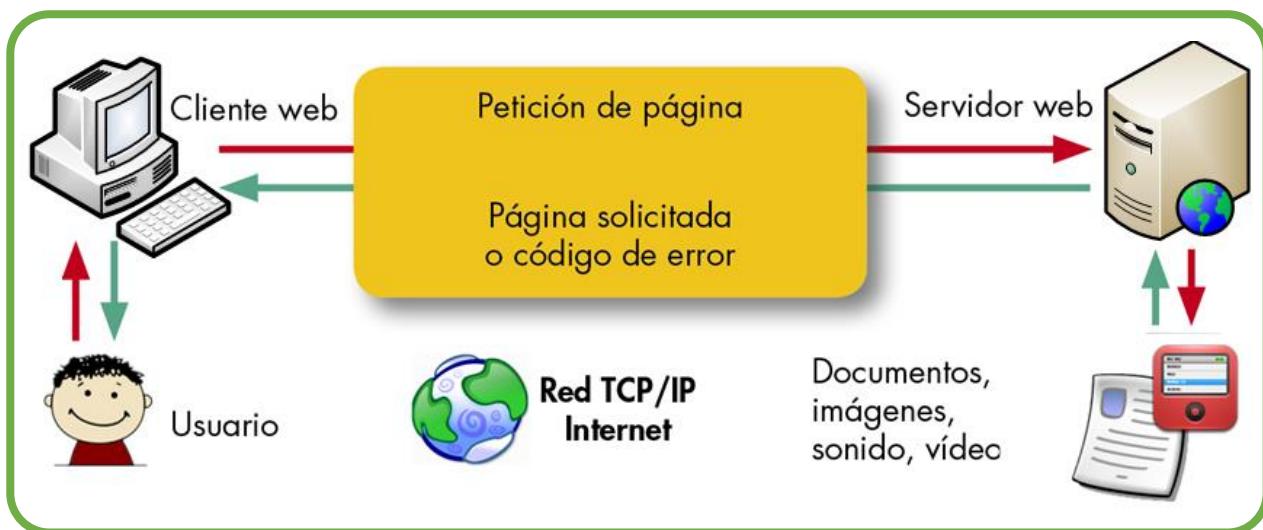


Figura 14. Esquema básico de una transferencia web

3.2.2 ▶ CSS



Las siglas **CSS** pertenecen al término *Hojas de estilo en cascada* (siglas en inglés de *Cascading Style Sheets*). CSS permite definir características como el aspecto con el que se va a presentar el documento en el navegador, como se va a imprimir o incluso como sería pronunciado por un dispositivo de lectura. Es usado tanto en documentos HTML como en documentos XML. Las

hojas de estilo permiten un mayor control sobre el aspecto de los documentos a la vez que ofrecen la posibilidad de separar el contenido de la presentación, siendo posible controlar el aspecto de varias páginas a la vez. El código CSS puede ser incrustado en el código HTML, sin embargo resulta más conveniente escribirlo en un archivo aparte, ya que, de esta manera, no hace falta cambiar el estilo de un elemento en todas las páginas en que aparece, sino que bastaría con modificar su definición para actualizar los cambios automáticamente.

Una hoja de estilo CSS está formada por una serie de reglas aplicadas a uno o más documentos HTML o XML. Una regla tiene 2 partes: un selector y su declaración. A su vez una declaración está compuesta por una propiedad y su valor asociado. Un ejemplo de regla CSS sería el siguiente:

```
hr {color:black;}
```

3.2.3 ► Javascript



Los comienzos de *Javascript* se remontan a 1995 cuando *Brendan Eich* un trabajador de Netscape creó el lenguaje *LiveScript*. Este lenguaje ofrecía una respuesta a la necesidad de incorporar un lenguaje de programación que se ejecutara en el navegador, para solucionar los problemas derivados del aumento de la complejidad de las aplicaciones, y la lentitud de la velocidad de navegación. Posteriormente se desarrollaría este lenguaje bajo el nombre de *JavaScript*.

JavaScript es un estándar basado en ECMA-262, una especificación para lenguajes script mantenido por *ECMA International*, una organización sin ánimo de lucro cuyo objetivo es la redacción de normas y estándares internacionales.

El lenguaje *JavaScript* ha tenido una gran evolución en los últimos años pasando de ser utilizado solamente en algunas utilidades de la parte cliente, hasta convertirse en un lenguaje popular y potente que cada año crece en funcionalidades. Javascript ha contribuido además al éxito de aplicaciones importantes como Google Maps, Gmail o Google Docs.

JavaScript unido a la interfaz de peticiones *XMLHttpRequest* (lo que se conoce como *AJAX*), ha permitido que eventos de usuario dentro del navegador tengan como consecuencias peticiones de información a servidores web. Entre los inconvenientes de *JavaScript* se encuentra su lentitud para ser ejecutado en los navegadores. El sistema operativo Google Chrome pretende dar solución a este problema permitiendo una ejecución más rápida de *JavaScript* y *AJAX*.

3.2.4 ► Bootstrap

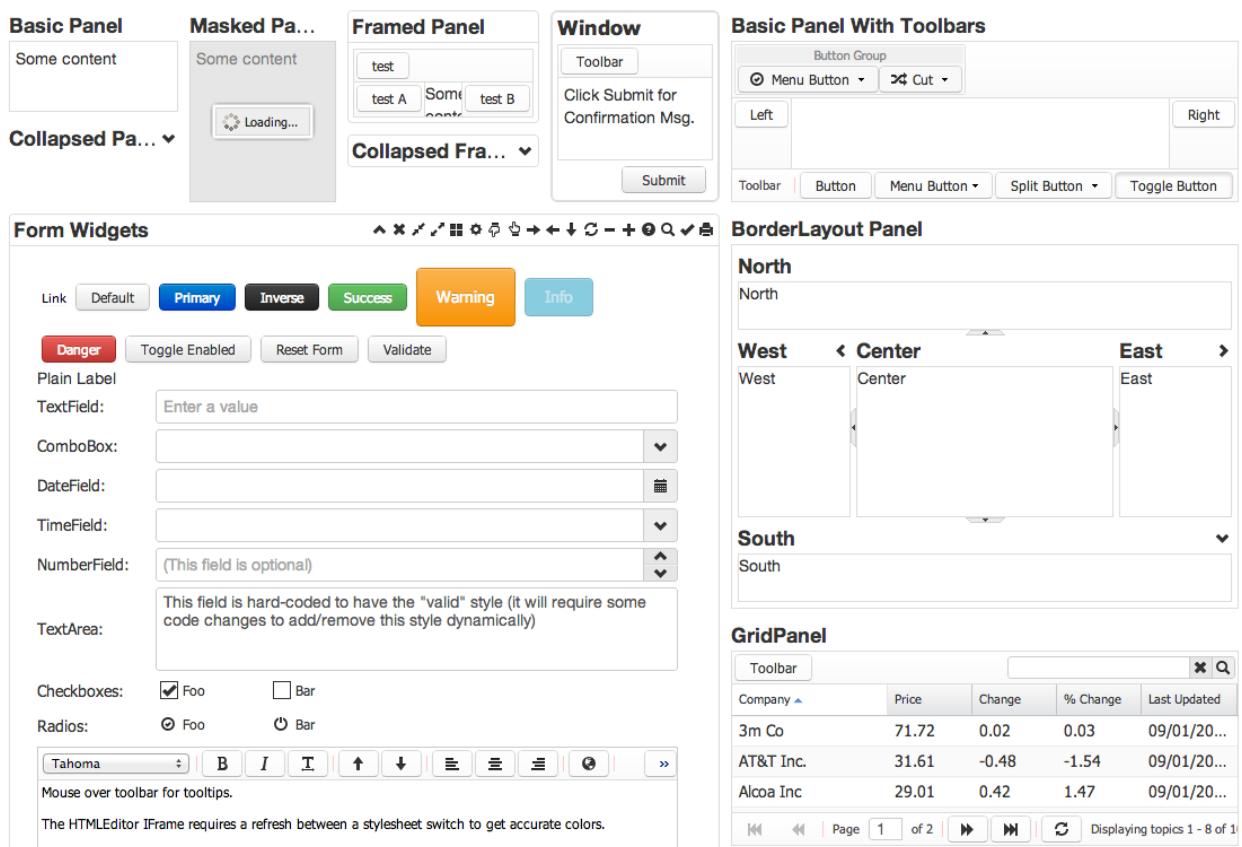


Twitter Bootstrap es un framework o conjunto de herramientas de software libre para diseño de sitios y aplicaciones web *responsive*. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de *JavaScript* opcionales adicionales.

Bootstrap 3 se ha creado desde cero pensando en los móviles de manera que en lugar de incluir algunos estilos opcionales para móviles todo ello ya está incluido en el propio Bootstrap. Para ello ha definido un sistema de doce rejillas que permitirá distribuir y adaptar los contenidos de un sitio web para cualquier tipo de dispositivos y navegadores.

Bootstrap tiene un soporte relativamente incompleto para HTML5 y CSS 3, pero es compatible con la mayoría de los navegadores web. La información básica de compatibilidad de sitios web o aplicaciones está disponible para todos los dispositivos y navegadores. Existe un concepto de compatibilidad parcial que hace disponible la información básica. Bootstrap es modular y consiste esencialmente en una serie de hojas de estilo LESS que implementan la variedad de componentes de la herramienta. Los desarrolladores pueden adaptar el mismo archivo de Bootstrap, seleccionando los componentes que deseen usar en su proyecto. Los ajustes son posibles en una medida limitada a través de una hoja de estilo de configuración central. Los cambios más profundos son posibles mediante las declaraciones LESS. El uso del lenguaje de hojas de estilo LESS permite el uso de variables, funciones y operadores, selectores anidados, así como clases mixin.

B Bootstrap



The screenshot displays a comprehensive overview of Bootstrap's UI components. It includes:

- Basic Panel**: Shows a simple panel with content and a collapsed panel dropdown.
- Masked Panel**: Shows a panel with a masked input field displaying "Loading...".
- Framed Panel**: Shows a panel with a frame containing multiple smaller panels labeled "test", "test A", "Some content", and "test B".
- Window**: Shows a window with a toolbar, a message "Click Submit for Confirmation Msg.", and a "Submit" button.
- Basic Panel With Toolbars**: Shows a panel with a toolbar containing a "Button Group" with "Menu Button" and "Cut" items, and other buttons like "Left", "Toolbar", "Button", "Menu Button", "Split Button", and "Toggle Button".
- Form Widgets**: A large section showing various form input types: Link, Default, Primary, Inverse, Success, Warning, Info, Danger, Toggle Enabled, Reset Form, Validate. It also shows Plain Label, TextField, ComboBox, DateField, TimeField, NumberField, and TextArea inputs.
- BorderLayout Panel**: A panel divided into North, South, East, West, and Center sections.
- GridPanel**: A grid-based panel showing a table of stock market data with columns for Company, Price, Change, % Change, and Last Updated. Data rows include 3m Co, AT&T Inc., and Alcoa Inc.

Figura 15. Componentes disponibles en la UI de Bootstrap

325 ► jQuery



jQuery es software libre y de código abierto que se ha convertido en la actualidad en uno de los complementos más esenciales para el desarrollo web, usado en millones de sitios en toda la web, ya que nos facilita en gran medida el desarrollo de aplicaciones enriquecidas del lado del cliente, en Javascript, compatibles con todos los navegadores.

jQuery no es un lenguaje, sino una serie de funciones y métodos de Javascript de manera que podremos referirnos a jQuery como un framework o incluso como una API de funciones, útiles en la mayoría de proyectos web. Por tanto, Javascript es el lenguaje y jQuery es una librería que podremos utilizar opcionalmente para facilitar el desarrollo en Javascript.

Antes de llegar jQuery los desarrolladores estaban obligados a discriminar entre los diversos navegadores, para ejecutar aquel código Javascript que funcionaba en cada uno. Con la llegada de jQuery tendríamos la ventaja de que ya no necesitamos preocuparnos del navegador que se utilice pues será la propia librería la que hará el trabajo "sucio" y ejecutará el código que sea compatible con el software del cliente que está accediendo a nuestra web. Para ello usaremos las funciones que jQuery nos proporciona, dentro de un grandísimo abanico de funcionalidades que además se extiende por medio de miles de plugins que ofrece la comunidad para implementar cualquier tipo de comportamiento.

La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX.

Cabría asimismo citar las siguientes características que jquery nos aporta: selección de elementos DOM, control de eventos, manipulación de la hoja de estilos CSS, conjunto de efectos y animaciones, obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes o la de suministrar la compatibilidad con la mayoría de los navegadores disponibles en la actualidad.

33 ► Parte SERVIDOR

3.31 ► Servidor Web Apache



El **Servidor HTTP Apache** es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual. El servidor Apache se desarrolla dentro del proyecto HTTP Server (`httpd`) de la *Apache Software Foundation*.

El proyecto Apache surgió para crear un recurso de libre acceso, robusto y de calidad comercial de un servidor Web HTTP. Este proyecto fue llevado a cabo por voluntarios de todo el mundo. En 1995 una primera versión del servidor resultó ser un gran éxito y en 1999 miembros del grupo Apache formaron el “*Apache Software Foundation*” que proporcionaría apoyo financiero, legal

y organizacional para asegurar el futuro del proyecto, e impulsaría gran cantidad de proyectos de código libre.

El servidor Apache es modular, multiplataforma, de código libre, extensible y resulta muy fácil conseguir ayuda y soporte. Estas características lo convirtieron enseguida en una de las opciones más populares. Desde 1996 fue el más usado llegando al 70% de los sitios en 2005. Apache se muestra como un servidor muy competitivo, rápido, estable y con más ventajas que otros servidores, incluso en entornos con millones de visitas al día. La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

3.3.2 ▶ PHP



PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

PHP fue creado por *Rasmus Lerdorf* en 1995, como un simple conjunto de *scripts de Perl* para el control de acceso a currículums on-line. El lenguaje fue creciendo en funcionalidad hasta permitir a los usuarios desarrollar sencillas aplicaciones Web dinámicas, llegando en 1998 a publicarse la tercera versión de PHP que ya era capaz de competir con productos similares como ASP o JSP.

PHP se considera uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos hasta el día de hoy, lo que ha atraído el interés de múltiples sitios con gran demanda de tráfico, como Facebook, para optar por el mismo como tecnología de servidor.

La unión entre su potencia y su simplicidad hacen de PHP un lenguaje muy popular, soportado por la mayoría de los proveedores de Internet y usado en millones de dominios.

La base de datos más adecuada para trabajar con PHP es *mySQL* ya que *mySQL* y *PHP* han sido desarrollados teniéndose en cuenta el uno al otro, ambos son *Open Source* y tienen una amplia comunidad de apoyo, su eficiencia y simplicidad permiten un procesamiento más rápido.

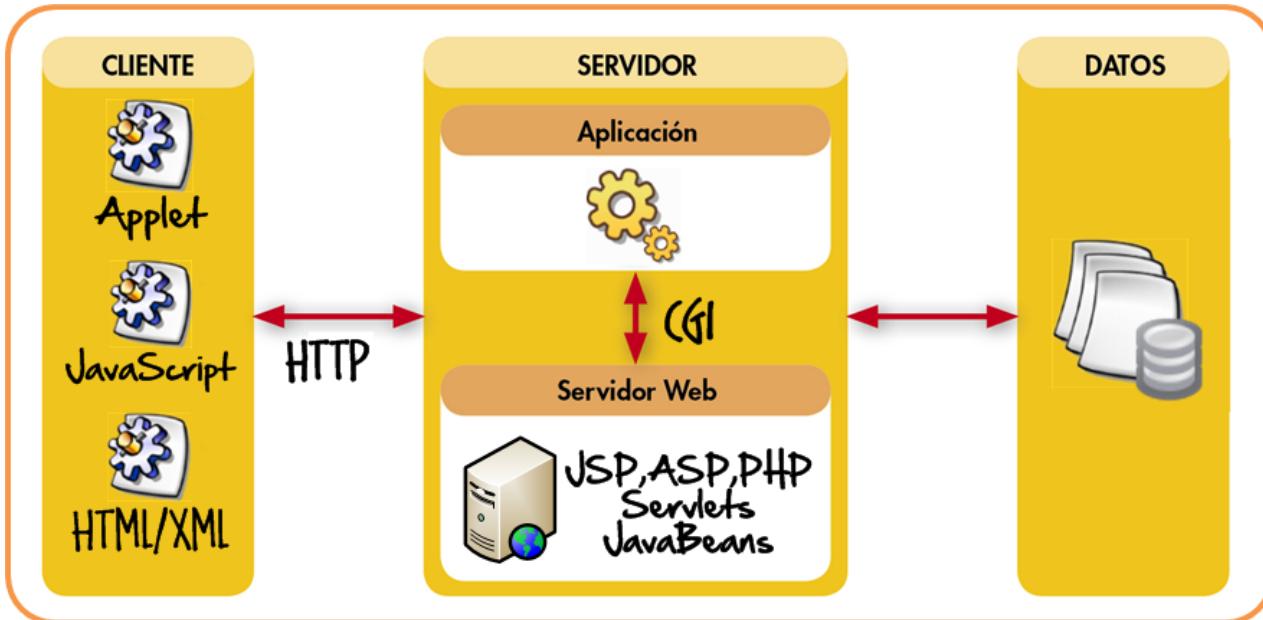


Figura 16. Esquema básico de funcionamiento de un servidor web

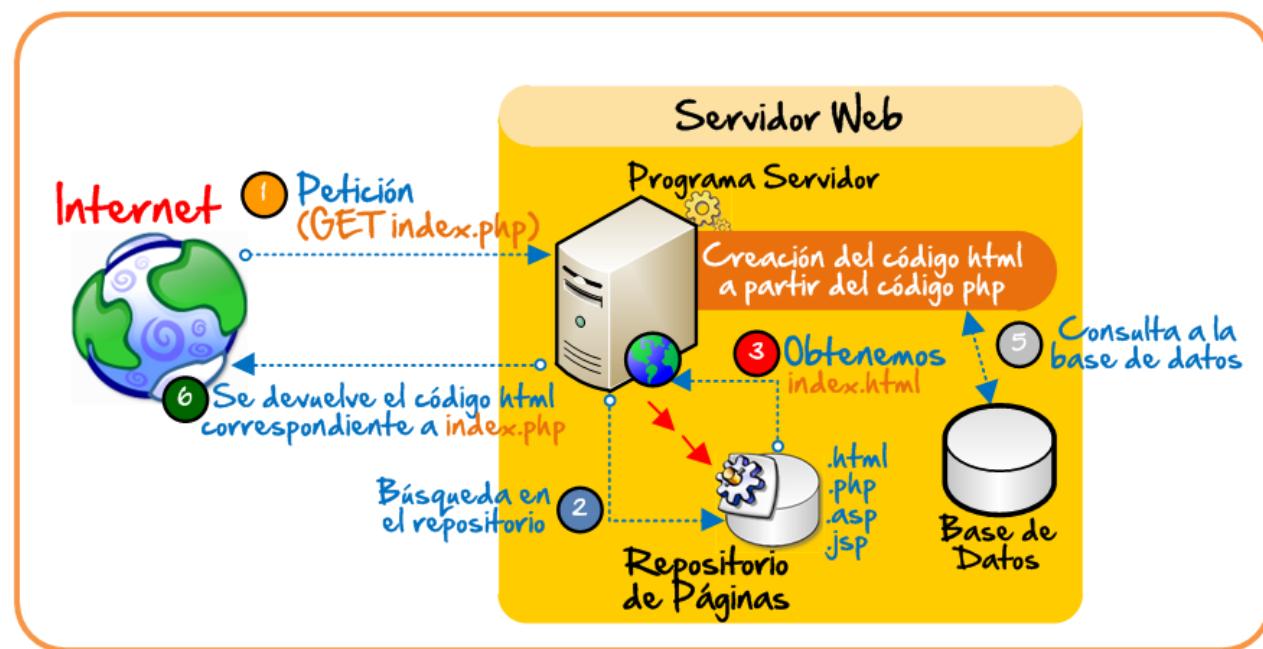


Figura 17. Esquema básico de funcionamiento de un servidor web con páginas dinámicas

3.3.3 MySQL



MySQL fue desarrollada en 1990 basada en *mSQL*, una simple y pequeña base de datos, a partir de la que se crearía una base de datos más robusta cuya evolución se convertiría en MySQL.

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario que cuenta en la actualidad con más de seis millones de instalaciones, que se distribuye bajo un sistema de doble licencia. Se puede obtener gratuitamente bajo licencia GPL y en el caso en el que se quiera distribuir una aplicación que no cumpla los términos GPL pero que incluya MySQL existe también una licencia comercial.

Ofrece una gran adaptabilidad permitiendo gran variedad de tipos de datos, y lo que es menos común, la posibilidad de elegir el tipo de tabla en el que guardar los registros, incluso permitiendo diferentes tipos de tablas en una misma base de datos. El motor de bases de datos *MyISAM* es el proporcionado por defecto y es capaz de manejar eficientemente grandes cantidades de datos, sin embargo existen muchos más tipos cada uno con sus ventajas.

MySQL Es una base de datos rápida, fácil de configurar y de aprender a usar, con amplia documentación de apoyo, acceso al código fuente y buena portabilidad. La última versión de MySQL, la versión 5.1 es comparable a cualquier otra base de datos de pago como Oracle, Informix o SQL Server.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código. Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet.

MySQL es usado en la actualidad por numerosos sitios web grandes y populares, como son Wikipedia, Google, Twitter, Flickr o YouTube.



3.3.4 ▶ CodeIgniter

CodeIgniter es un framework PHP para la creación rápida de aplicaciones web, y por tanto, un programa o aplicación web desarrollada en PHP para la creación de cualquier tipo de aplicación web bajo PHP. Es un producto de código libre, libre de uso para cualquier aplicación.

Como cualquier otro framework, CodeIgniter contiene una serie de librerías que sirven para el desarrollo de aplicaciones web y además propone una manera de desarrollarlas que debemos seguir para obtener provecho de la aplicación, marcando una manera específica de codificar las páginas web y clasificar sus diferentes scripts, que sirve para que el código esté organizado y sea más fácil de crear y mantener. CodeIgniter implementa el proceso de desarrollo *llamado Model View Controller (MVC)*, que es un estándar de programación de aplicaciones, utilizado tanto para hacer sitios web como programas tradicionales. Este sistema tiene sus características, que veremos en artículos siguientes.

CodeIgniter contiene muchas ayudas para la creación de aplicaciones PHP avanzadas, que hacen que el proceso de desarrollo más rápido. A la vez, define una arquitectura de desarrollo que hará que programemos de una manera más ordenada y contiene diversas herramientas que ayudan a hacer aplicaciones más versátiles y seguras.

3.35 ▶ Control de Versiones. GitHub



El *control de versiones* permite gestionar y controlar los diversos cambios que se realizan sobre los elementos de la aplicación o de alguna de las opciones de configuración de la misma. Una *versión*, *revisión* o *edición* de un producto, es el estado en el que se encuentra el mismo en un momento dado de su desarrollo o modificación.

En el desarrollo de la aplicación he utilizado como herramienta de control de versiones **GitHub**. *Git* es uno de los sistemas de control de versiones más populares entre los desarrolladores, y gran parte culpa de su popularidad la tiene *GitHub*, un excelente servicio de alojamiento de repositorios de software con este sistema, que ofrece hoy en día un conjunto de características muy útiles para el trabajo en equipo, no en vano, es el servicio elegido por proyectos de software libre como jQuery, reddit, Sparkle, curl, Ruby on Rails, node.js, ClickToFlash, Erlang/OTP, CakePHP, Redis, y otros muchos. Además, algunas de las grandes empresas de Internet, como Facebook, alojan ahí sus desarrollos públicos, tales como el SDK, librerías, ejemplos, etc.

Al alojar el código, podemos decidir si éste será público o su visibilidad estará limitada a un grupo de usuarios determinado que serán los únicos que tendrán permiso para acceder al mismo y subir cambios. Asimismo, *GitHub* posee un *visor de código* mediante el cual, a través del navegador, podremos consultar en un instante el contenido de un determinado fichero, con su resaltado de sintaxis correspondiente para el lenguaje en el que esté escrito. Por supuesto, se puede navegar por cualquiera de las versiones del mismo, de modo que podemos ver el contenido de un fichero antes de que se dieran varias actualizaciones de cambios. Seleccionar un determinado *commit*, *tag* o rama es cuestión de pocos clicks.

4 Arquitectura de la aplicación. El Patrón MVC (Modelo Vista Controlador)

El *Modelo Vista Controlador* (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos (Modelo, Vista y Controlador). El Patrón MVC se ve frecuentemente en aplicaciones Web, donde la Vista es la página HTML y el código que provee de datos dinámicos a la página; el Modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; el Controlador es el responsable de recibir los eventos de entrada desde la Vista. Un modelo puede tener diversas vistas, cada una con su correspondiente controlador.

Los elementos que constituyen el *MVC* son los siguientes:

- 1) **Modelo:** Es la representación de la información en el sistema. Trabaja junto a la *Vista* para mostrar la información al usuario y es accedido por el *Controlador* para añadir, eliminar, consultar o actualizar datos, llevando un registro de las vistas y controladores del sistema. En él se programa todo aquello relacionado con las bases de datos y las reglas del negocio (la funcionalidad del sistema). Un ejemplo de regla podría ser: “*Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor*”.
- 2) **Vista:** asociada a su controlador correspondiente, presenta al *Modelo* en un formato adecuado para que el usuario pueda interactuar él mediante elementos como textos, formularios de entrada, etc. En ella se programa la parte visual del software (interfaz de usuario), generalmente, mediante HTML, CSS y JavaScript.
- 3) **Controlador:** Implementa la lógica del programa. Controla todo lo que puede realizar nuestra aplicación recibiendo, tratando y dando respuesta a los eventos enviados por el usuario o por la propia aplicación. Interactúa tanto con el *Modelo* como con la *Vista*, pidiendo al *Modelo* los datos y mostrándolos en la *Vista*.

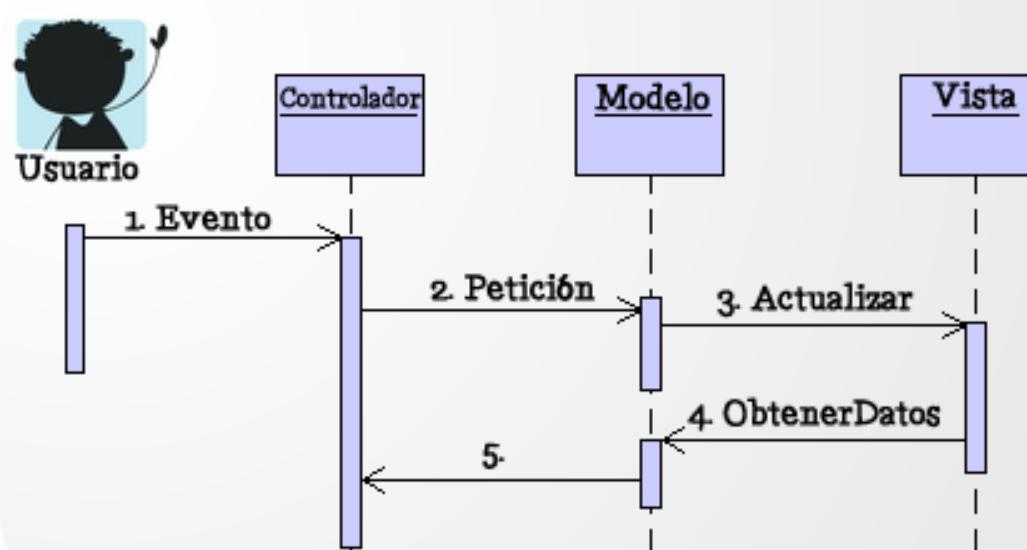


Figura 18. Diagrama de Secuencia del Esquema MVC

La secuencia de funcionamiento que sigue el modelo MVC, tal como muestra el diagrama de secuencia de la figura anterior sería:

1. El usuario introduce el evento.
2. El Controlador recibe el evento y lo traduce en una petición al Modelo (aunque también puede llamar directamente a la vista).
3. El modelo (si es necesario) llama a la vista para su actualización.
4. Para cumplir con la actualización la Vista puede solicitar datos al Modelo.
5. El Controlador recibe el control.

La popularidad de este diseño se debe más que todo a que es mucho más fácil organizar el proceso de desarrollo de aplicaciones grandes.

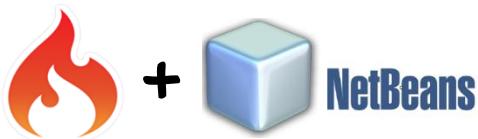
Entre las ventajas que tiene el uso del MVC cabrá señalar las siguientes:

- Clara separación entre interfaz, lógica de negocio y de presentación, que además provoca parte de las ventajas siguientes.
- Sencillez para crear distintas representaciones de los mismos datos.
- Facilidad para la realización de pruebas unitarias de los componentes, así como de aplicar desarrollo guiado por pruebas (TDD).
- Reutilización de los componentes.
- Simplicidad en el mantenimiento de los sistemas.
- Facilidad para desarrollar prototipos rápidos.
- Los desarrollos suelen ser más escalables.

Entre las principales desventajas que presenta señalar estas otras:

- Tener que ceñirse a una estructura predefinida, lo que a veces puede incrementar la complejidad del sistema. Hay problemas que son más difíciles de resolver respetando el patrón MVC.
- La curva de aprendizaje para los nuevos desarrolladores se estima mayor que la de modelos más simples como Webforms.
- La distribución de componentes obliga a crear y mantener un mayor número de ficheros.

La implementación de la aplicación mediante este modelo la he realizado utilizando el framework *CodeIgniter* y el editor de código *NetBeans*



5►Manual de usuario

5.1►Login

El objetivo de esta pantalla es la de crear un sistema de autenticación capaz de identificar a los usuarios que deseen acceder a la aplicación por medio de un nombre de usuario y una contraseña. En función del resultado de la validación podremos acceder a la aplicación con el rol de *Administrador*, con el rol de *Usuario* o bien rechazar el acceso por no ser válidos los datos introducidos.



Figura 19. Pantalla de Login

5.2►Pantalla de inicio

Esta pantalla sería a la que accederá todo usuario que se autentique correctamente en la aplicación y constituye todo un menú de navegación de la aplicación desde el cual se puede acceder a los distintos bloques funcionales que en ella se incluyen.

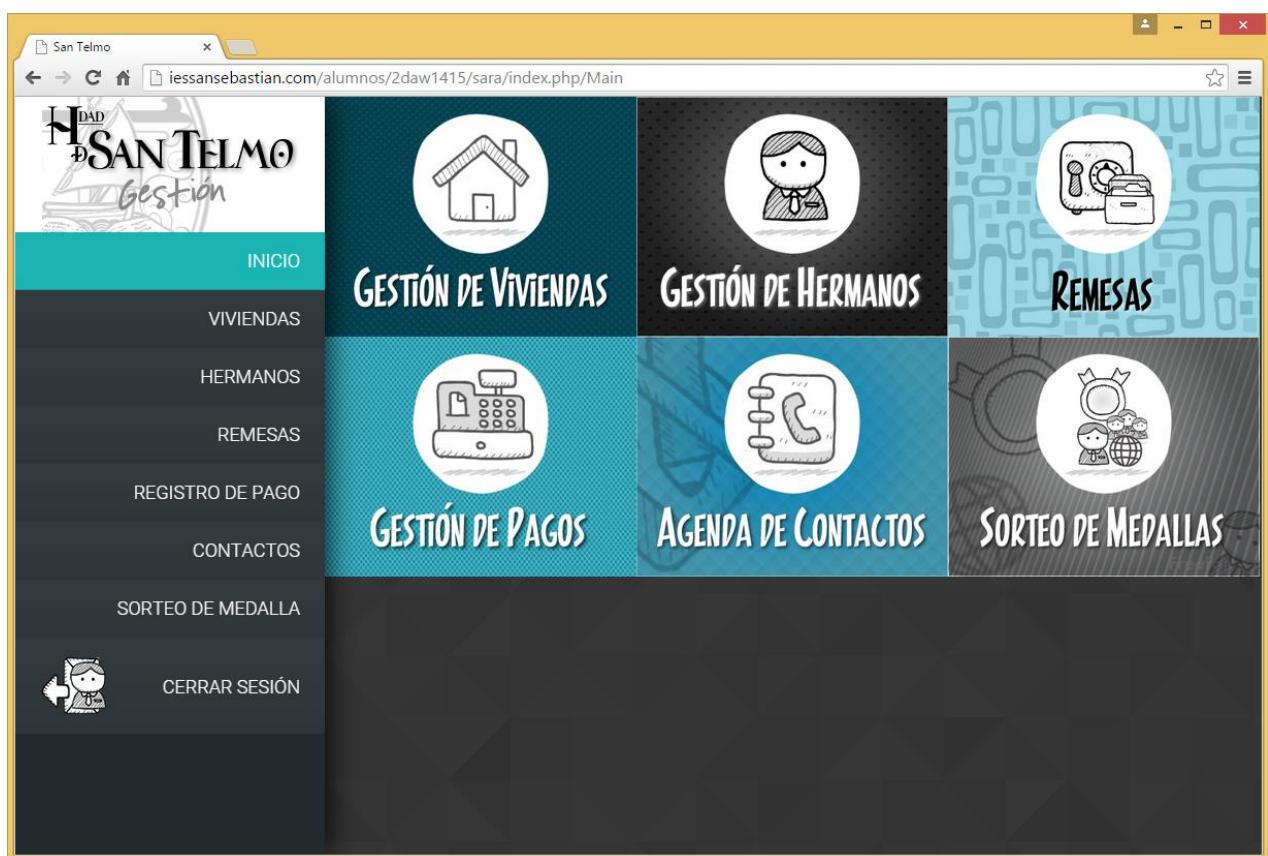


Figura 20. Pantalla de Inicio de la aplicación GestST

Cabría señalar que su diseño, realizado con Bootstrap, es *responsive*, gracias a lo cual nuestra aplicación tendrá la capacidad reorganizar elementos de diseño y contenidos para adaptarse a los distintos dispositivos desde los que se acceda.

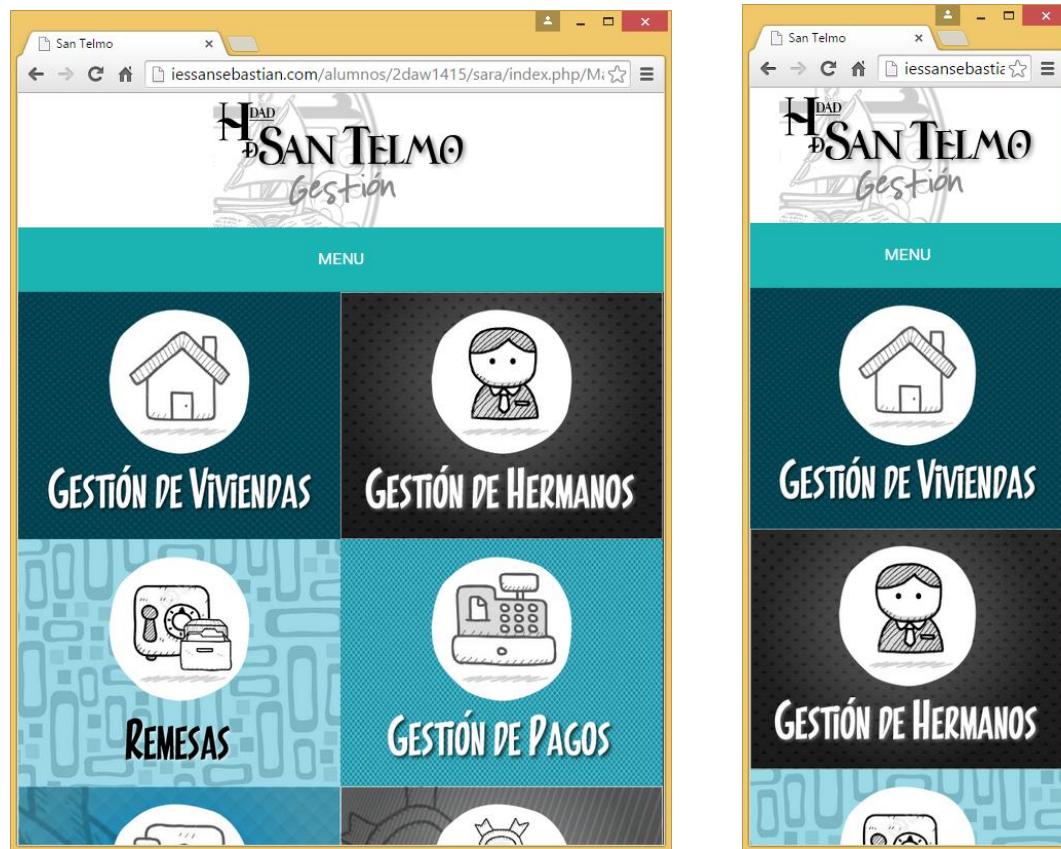


Figura 21. Pantalla de Inicio de la aplicación GestST vista en distintas áreas de visualización

5.3 ▶ Pantalla Viviendas

Esta pantalla sería la que permitirá registrar las distintas viviendas de la localidad que tengan asociadas algún pago. La información de las mismas es relativa a la ubicación de cada una de ellas dentro del núcleo poblacional: barriada, línea y número, datos que permitirán identificar únicamente a cada una de ellas. Asimismo se registra información adicional sobre la familia o los hermanos que en ella residen. Inicialmente, esta página mostrará el listado de las viviendas ya registradas.

Listado de viviendas					
Barriada	Línea	Número	Observaciones	Acciones	
Vieja	1	1	Observaciones id1		
Nueva	2	2	Observaciones id2		
Kiosko	3	3	Observaciones id3		
Vieja	4	4	Observaciones id4		

Figura 22. Pantalla de Viviendas

Dar de alta una vivienda

Barriada	Nueva
Línea	1
Número	5

Familia Romero/Caro

AÑADIR



Figura 23. Pantalla de registro de nuevas Viviendas

Se ha añadido la vivienda correctamente

Listado de viviendas

Barriada	Línea	Número	Observaciones	Acciones
Vieja	1	1	Observaciones id1	
...				
Nueva	1	5	Familia Romero/Caro	

Figura 24. Pantalla de confirmación del registro de una nueva Vivienda

5.4 ▶ Pantalla Hermanos

Esta pantalla sería la que permitirá gestionar los distintos hermanos de la Hermandad. Cabría señalar que en función del rol con el que se haya autenticado el usuario en la aplicación éste podrá acceder y editar la información, si es usuario *Administrador*, o sólo acceder a ella, si es usuario *Usuario*. Cada perfil tendrá su propia vista con sus funciones correspondientes.

Inicialmente esta página mostrará una lista breve con los hermanos registrados hasta el momento así como un conjunto de botones de acción adjunto a cada hermano mediante los cuales podremos realizar cualquiera de las operaciones disponibles en el rol activo.

El color de los elementos de la lista puede ser verde o rojo en función de si el hermano correspondiente está al día en el pago de las cuotas o no respectivamente.

Asimismo, desde cada hermano de la lista, con el perfil Administrador, podremos acceder de manera directa a su información de pago para visualizar las distintas remesas pagadas o pendientes de pago.

Listados de hermanos dados de alta



#	Nombre	Primer apellido	Segundo apellido	Familia	¿Tiene medalla?	Acciones
1	Sara	Alamillo	Arroyo	2	No	   
2	Celia	Alamillo	Arroyo	2	No	   
3	Maria Inmaculada	Alamillo	Arroyo	2	No	   
4	Manuel Alfonso	Romero	Romero	2	No	   
5	Ana	Barrio	Yáñez	2	No	   

Figura 25. Pantalla principal de Gestión de Hermanos

Vista detalla de un hermano

Datos personales		Vivienda	Barrada: Nueva Número: 2 Línea: 2
Tratamiento	Señora	Familia	2
Nombre	Sara	¿Tiene medalla?	No
Primer apellido	Alamillo	Dirección	
Segundo apellido	Arroyo	Tipo de vía	Plaza
DNI	dos	Dirección	dos
Contacto		Número	2
Móvil	dos	Piso	2
Fijo	dos	Puerta	2
Email	dos	Código postal	22222
Redes Sociales		Población	dos
Twitter	dos	Provincia	Albacete
Facebook	dos	Formas de pago	
Instagram	dos	Tipo	Domiciliado
		Cuenta corriente	



Figura 26. Pantalla de Vista-detalle de Hermano

Listado de pagos de Sara Alamillo Arroyo

Año	Descripción	Primer pago	Segundo pago
2015	remesa1	2015-06-09	NO PAGADO
2015	remesa2	NO PAGADO	NO PAGADO

1 2 » Última



Figura 27. Pantalla de Pagos asociados a un Hermano

Modificar los datos de un hermano

Familia	2
Vivienda	Barriada: Nueva - Línea: 2 - Número: 2
Medalla	Si
Tratamiento	Señora
Nombre	Sara
Primer apellido	Alamillo
Segundo apellido	Arroyo
DNI	dos
Residencia habitual	
Tipo de vía	Plaza
Dirección	dos
Número	2
Piso	2
Puerta	2
Código postal	22222
Población	dos
Provincia	Alava
Redes sociales	
Twitter	dos
Facebook	dos
Instagram	dos
MODIFICAR	

Datos personales

Contacto

Pago de cuotas



Figura 28. Pantalla de Edición de un Hermano

55 ▶ Pantalla Remesas

Esta pantalla sería la permitiría registrar las distintas remesas organizadas por la Hermandad a las que se les asociarán las distintas cuotas de pago.

Listado de remesas		
Año	Descripción	Acciones
2015	remesa1	 
2015	remesa2	 

Figura 29. Pantalla de Edición de un Hermano

Añadir una nueva remesa	
Descripción	Año
<input type="text"/>	<input type="text"/>
	AÑADIR

Figura 30. Pantalla de Edición de un Hermano

<div style="background-color: #f0e68c; padding: 5px; margin-bottom: 10px;"> No se puede eliminar una remesa con cuotas asociadas </div>		
Listado de remesas		
Año	Descripción	Acciones
2015	remesa1	 
2015	remesa2	 

Figura 31. Pantalla de Edición de un Hermano

56 ▶ Pantalla Pagos

Esta pantalla sería la permitiría registrar los pagos de las cuotas que los hermanos hacen de las distintas remesas organizadas por la Hermandad.

Registro de pagos

Nombre completo	Hermano
-----------------	---------

Registro de pagos

Nombre completo	Sara Alamillo Arroyo
Año	Año

Registro de pagos

Nombre completo	Sara Alamillo Arroyo
Año	2015
Remesa	Remesa

Registro de pagos

Nombre completo	Sara Alamillo Arroyo
Año	2015
Remesa	remesa2

Plazos

Cuota 1	dd/mm/aaaa
Cuota 2	dd/mm/aaaa

REGISTRAR

Plazos

Cuota 1	dd/mm/aaaa
Cuota 2	<input style="width: 150px; height: 20px; border: 1px solid #ccc; padding: 2px;" type="text" value="junio de 2015"/> ▼ ▲

REGISTRAR



Figura 32. Secuencia de pantallas para el registro de pagos

57►Pantalla Agenda de Contactos

Esta pantalla sería a través de la cual se podrá registrar información sobre empresas y/o personas que prestan algún tipo de servicio a la Hermandad a modo de agenda de contactos. Inicialmente mostrará la lista con los contactos ya existentes.

Agenda de contactos



Empresa	Nombre	Primer apellido	Segundo apellido	Teléfono móvil	Teléfono fijo	Acciones
dos	dos	dos	dos	dos	dos	  

Figura 33. Pantalla inicial de la Agenda de Contactos

Vista detalla de un contacto

Datos personales		Tipo de contacto	
Empresa	dos	Dirección	
Tratamiento	Señora	Tipo de vía	Calle
Nombre	dos	Dirección	dos
Primer apellido	dos	Número	2
Segundo apellido	dos	Piso	2
CIF	dos	Puerta	2
Contacto		Código postal	00000
Móvil	dos	Población	dos
Fijo	dos	Provincia	Albacete
Email		Redes sociales	
		Twitter	dos
		Facebook	dos
		Instagram	dos



Figura 34. Pantalla ver detalles de la Agenda de Contactos

Modificar un contacto de la agenda

Tipo de contacto	Dos	Domicilio	
Datos personales		Tipo de vía	Calle
Empresa	dos	Dirección	dos
Tratamiento	Señora	Número	2
Nombre	dos	Piso	2
Primer apellido	dos	Puerta	2
Segundo apellido	dos	Código postal	00000
CIF	dos	Población	dos
Contacto		Provincia	Albacete
Teléfono móvil	dos	Redes sociales	
Teléfono fijo	dos	Twitter	dos

Figura 35. Pantalla de edición de un contacto

Eliminar un contacto de la agenda

¿Está seguro que quiere eliminar el siguiente registro y todos los datos asociados?

Campo	Valor
Número de contacto	2
Nombre de la empresa	dos
Tratamiento	Señora
Nombre	dos
Primer apellido	dos
Segundo apellido	dos
CIF	dos
Tipo de vía	Calle
Dirección	dos
...	
Tipo de pago	2
Provincia	2

SI **NO**

Figura 36. Pantalla de eliminación de un contacto

58 ▶ Pantalla Sorteo de Medallas

Esta pantalla sería a través de la cual se podrá generar las distintas papeletas que entrarán en sorteo. Inicialmente mostrará la lista con todos los hermanos de manera que podremos deseleccionar aquellos que por cualquier incumplimiento o circunstancia particular debiera obviarse. Una vez seleccionados los hermanos participantes se procedería a generar las papeletas del sorteo en un documento pdf.

Participantes del sorteo de medallas

GENERAR SORTEO

En el siguiente listado están recogidos todos los hermanos que podrán ser participantes en el sorteo de medallas. Quedan excluidos los hermanos que ya han sido premiados con una medalla anteriormente.

Para excluir a cualquier otro hermano, basta con desmarcar la casilla correspondiente a dicho hermano.

<input checked="" type="checkbox"/> 2. dos dos dos	<input checked="" type="checkbox"/> 3. Sara Alamillo Arroyo	<input checked="" type="checkbox"/> 4. Celia Alamillo Arroyo	<input checked="" type="checkbox"/> 5. María Inmaculada Alamillo Arroyo
<input checked="" type="checkbox"/> 6. Manuel Alfonso Romero Romero	<input checked="" type="checkbox"/> 7. Ana Barrio Yáñez	<input checked="" type="checkbox"/> 8. Alfonso Romero	<input checked="" type="checkbox"/> 9. Alfonso Romero
<input checked="" type="checkbox"/> 10. Alfonso Romero	<input checked="" type="checkbox"/> 11. Alfonso Romero	<input checked="" type="checkbox"/> 12. Alfonso Romero	<input checked="" type="checkbox"/> 13. Alfonso Romero
<input checked="" type="checkbox"/> 14. Alfonso Romero	<input checked="" type="checkbox"/> 15. Alfonso Romero	<input checked="" type="checkbox"/> 16. Alfonso Romero	<input checked="" type="checkbox"/> 17. Alfonso Romero
<input checked="" type="checkbox"/> 18. Alfonso Romero	<input checked="" type="checkbox"/> 19. Alfonso Romero	<input checked="" type="checkbox"/> 20. Alfonso Romero	<input checked="" type="checkbox"/> 21. Alfonso Romero

▼

2	dos dos dos
3	Sara Alamillo Arroyo
4	Celia Alamillo Arroyo
5	Maria Inmaculada Alamillo Arroyo
6	Manuel Alfonso Romero Romero
7	Ana Barrio Yáñez

Figura 37. Pantallas de generación del Sorteo de Medallas

6 ► Plan de Evaluación

6.1 ► Realización de la prueba

- **Objetivo.** El objetivo básico de la evaluación de la usabilidad de una aplicación será el de identificar posibles errores potenciales de diseño para así tratar de mejorar la eficiencia, productividad y el nivel de satisfacción del usuario final de la misma.
- **Fecha, hora y lugar del test:** La prueba tendrá lugar el día 15 de Junio entre las 17:00 y las 19:00 horas en el Taller de Informática nº4 perteneciente al Curso: 2º DAW. La duración de la realización de la prueba no excederá los 20 minutos, siendo el tiempo total estimado de 1:30' horas.
- **Participantes.** Los usuarios que participarán en la evaluación de la aplicación serán el tutor del proyecto y alumnos voluntarios pertenecientes al curso 2º DAW. Todos ellos están familiarizados con el uso de internet y smartphones así como con el diseño de aplicaciones web, por lo que se ajustan al perfil del potencial usuario definido para nuestra aplicación y con lo que nos aseguramos que los resultados que obtengamos serán estadísticamente significativos.

6.2 ► Instrucciones para la realización de la prueba

- Se explicará cuál es el motivo de la realización de esta prueba y el objetivo que se pretende alcanzar con ella.
 - ▶ **Motivación:** Realizar la evaluación con usuarios reales de la aplicación desarrollada.
 - ▶ **Objetivo:** Se va a evaluar la utilización de las distintas funcionalidades disponibles en la aplicación a fin de poder determinar si existen errores en el diseño de su interfaz que imposibilite o dificulte al usuario la realización de una tarea determinada, así como obtener una referencia del grado de usabilidad de la misma.
- Se presentará la **GestST** como una aplicación *responsive* diseñada para ayudar a los miembros de la hermandad en todas las tareas relativas a la gestión de la misma.
- Se enumerarán las distintas funcionalidades que permite gestionar la aplicación:
 - ▶ Gestión de Viviendas asociadas al pago de cuotas.
 - ▶ Gestión de Hermanos (Altas, bajas, modificaciones y consultas)
 - ▶ Gestión de Pagos (Altas, bajas, modificaciones y consultas)
 - ▶ Gestión de Remesas.
 - ▶ Gestión de Agenda de Contactos.
 - ▶ Elaboración del Sorteo de Medallas.
- Se les entregará un cuestionario en el que se especifican las distintas tareas que deberán realizar y la información que deberá suministrar de cada una de ellas una vez la vayan completando. En él se pedirá que valoren el nivel de dificultad asociado a cada una de ellas en una escala de 1 a 10, siendo '1': *muy difícil/confuso* y '10' *fácil de realizar/entender*. Asimismo se les indicará que si tienen problemas en la realización de alguna tarea y el tiempo dedicado excede los 3 minutos el participante



deberá pasar a la siguiente tarea marcando dicha tarea como *no completada* e indicando en el cuestionario el problema que se encontró.

- ▶ Por último, antes de comenzar la realización de la prueba, se recalcará que el objetivo de ésta no es valorar las habilidades de los participantes sino el de evaluar la usabilidad de la aplicación, tratando de identificar posibles errores potenciales de diseño para así intentar de mejorar la eficiencia, productividad y el nivel de satisfacción del usuario final de la misma.

En concreto, nuestro objetivo será determinar si un usuario que conozca la existencia de nuestra aplicación y se anime a probarla puede utilizar de manera efectiva las distintas funcionalidades que ésta ofrece.

Se le advertirá de las limitaciones inherentes a un prototipo y se les leerá y comentará los objetivos que persigue la prueba para que los participantes los tengan presentes durante la evaluación. Esta lista de objetivos se adjunta al cuestionario.

→ Objetivos del test

- ▶ *Identificar los errores del sistema de navegación de la aplicación* que puedan implicar la imposibilidad de localizar funciones, la necesidad de un excesivo número de clics para completar una tarea o el no seguir el flujo de pantalla recomendado.
 - ▶ *Identificar los errores de presentación* que puedan dificultar la localización de la información deseada en las pantallas de la aplicación y el poder interactuar adecuadamente, a causa de errores en la selección de elementos de la interfaz debidos a ambigüedades del etiquetado.
 - ▶ *Identificar los problemas en la secuencia de pasos necesarios para completar las distintas tareas* que puedan estar provocados por el uso indebido de la barra de navegación, el menú de inicio o de algún campo de entrada.
 - ▶ Los datos obtenidos del test se utilizarán para *determinar si la interfaz podría ser considerada como eficaz y eficiente* o, por el contrario, es necesaria la realización de modificaciones que mejoren la usabilidad y el rendimiento de la aplicación.
 - ▶ *Establecer un perfil de cada usuario participante* en test que recoja su nivel de rendimiento para considerarlo en futuras evaluaciones.
 - ▶ *Obtener el grado de satisfacción* de los usuarios y *el nivel de aplicabilidad* en la vida real que tendría la aplicación.
-
- ▶ Tras todo lo anterior daría comienzo la prueba que, tal como se indicó anteriormente, tendría una duración de *20 minutos*.
 - ▶ Una vez finalizada la realización del cuestionario se organizará un pequeño debate en el que se discutirá grupalmente las impresiones de la aplicación, pudiendo profundizar sobre aquellos aspectos de la aplicación que hayan resultado más interesantes o complejos.
 - ¿Qué es lo que más y menos le ha gustado de la aplicación?



- ¿Cree que puede resultar útil? ¿Recomendaría su uso? ¿Por qué?
 - Respecto al sistema de navegación de la aplicación ¿cree que está bien diseñado? ¿ha encontrado alguna dificultad? ¿Recomendaría realizar algún cambio? En caso afirmativo, ¿cuál?
 - ¿Cree que la organización general de la aplicación es lógica y adecuada? ¿Recomendaría realizar algún cambio? En caso afirmativo, ¿cuál?
 - ¿Qué tarea/tareas le han resultado más complejas o difícil de realizar? Dicha complejidad ¿es debida a un mal diseño de la interfaz correspondiente o a una lógica de funcionamiento confusa o poco adecuada?
 - ¿Qué cambiaría de la aplicación? ¿Por qué? ¿Cómo realizaría dicho cambio?
- Finalmente se responderá cualquier pregunta que puedan plantear los participantes y se les dará a conocer los planes de desarrollo futuros respecto al desarrollo de la aplicación así como información de contacto y seguimiento para todos aquellos que estén interesados en la aplicación. Asimismo, se les pedirá a los participantes que completen un cuestionario demográfico que nos permita conocer el perfil de los mismos.

6.3 ▶ Valoración y resultados de la prueba

Los aspectos a considerar para realizar la valoración de la prueba serían los siguientes:

- **Tasa de éxito de terminación de las tareas planteadas.** Una tarea se considerará completada cuando el participante alcance alguno de los criterios de parada o situaciones finales correspondientes.
- **Tasas de error.** Se recogerán tanto los errores críticos, que serían los que no han sido resueltos durante el proceso de realización de una tarea o los que producen resultados incorrectos, como los errores no críticos, de los que se recuperó el participante durante la realización de la tarea.
- **Tiempo de finalización de cada tarea.** Se medirá el tiempo de realización de las distintas tareas, calculados desde que el participante comienza su ejecución hasta que se alcance el criterio de parada.

Los resultados que pretendemos conseguir en esta prueba de usabilidad de nuestra aplicación serían los siguientes:

- Que el 100% de los participantes puedan completar con éxito la totalidad de las tareas planteadas sin errores críticos.
- Que el 80% de los participantes puedan completar las tareas sin cometer ningún tipo de errores (ni críticos ni no críticos).



QUESTIONARIO DE EVALUACIÓN

TAREA #1

Completada Sí <input type="checkbox"/> No <input type="checkbox"/>	Acceda a la aplicación con los roles de Administrador y Usuario. Verifique la disponibilidad de las distintas funciones según el rol correspondiente.	
⌚ Tiempo		Nivel de dificultad (1..5)
Problemas detectados		
Cambios /Sugerencias		

TAREA #2

Completada Sí <input type="checkbox"/> No <input type="checkbox"/>	Con el rol de Administrador, registre una vivienda, visualice las viviendas registradas, elija la vivienda introducida, editela y verifique que los datos se han actualizado correctamente.	
⌚ Tiempo		Nivel de dificultad (1..5)
Problemas detectados		
Cambios /Sugerencias		

**TAREA #3**

Completada Sí <input type="checkbox"/> No <input type="checkbox"/>	Con el rol de administrador, registre los datos de un hermano asociado a una determinada vivienda. Liste los hermanos, edite los datos de hermano introducido y verifique que los datos se actualizaron correctamente.	
Tiempo		Nivel de dificultad (1..5)
Problemas detectados		
Cambios /Sugerencias		

TAREA #4

Completada Sí <input type="checkbox"/> No <input type="checkbox"/>	Con el rol de administrador, registre los datos de un determinado contacto en la agenda. Liste los contactos, edite los datos de contacto introducido y verifique que los datos se actualizaron correctamente.	
Tiempo		Nivel de dificultad (1..5)
Problemas detectados		
Cambios /Sugerencias		



TAREA #5

Completada Sí <input type="checkbox"/> No <input type="checkbox"/>	Con el rol de administrador, registre los datos del pago de la cuota de una determinada remesa por parte de un hermano. Liste los hermanos, acceda a los datos de pagos asociados al hermano correspondiente y verifique que los datos del pago se actualizaron correctamente.		
⌚ Tiempo			Nivel de dificultad (1..5)
Problemas detectados			
Cambios /Sugerencias			

TAREA #6

Completada Sí <input type="checkbox"/> No <input type="checkbox"/>	Con el rol de administrador, registre los datos de una remesa. Liste las remesas existentes, edite los datos de la introducida y verifique que los datos se actualizaron correctamente.		
⌚ Tiempo			Nivel de dificultad (1..5)
Problemas detectados			
Cambios /Sugerencias			

TAREA #7

Completada Sí <input type="checkbox"/> No <input type="checkbox"/>	Genere las papeletas del sorteo de medallas excluyendo de él algunos de los hermanos		
⌚ Tiempo		Nivel de dificultad (1..5)	
Problemas detectados			
Cambios /Sugerencias			

● (UESTIONES FINALES)

1. ¿Qué es lo que más y menos le ha gustado de la aplicación? ¿Por qué?

2. ¿Cree que puede resultar útil? ¿Recomendaría su uso? ¿Por qué?

3. Respecto al sistema de navegación de la aplicación ¿cree que está bien diseñado? ¿Ha encontrado alguna dificultad? ¿Recomendaría realizar algún cambio? En caso afirmativo, ¿cuál?

4. ¿Cree que la organización general de la aplicación es lógica y adecuada? ¿Creería conveniente realizar algún cambio? En caso afirmativo, ¿cuál?

5. ¿Qué tarea/tareas le han resultado más complejas o difícil de realizar? Dicha complejidad ¿es debida a un mal diseño de la interfaz correspondiente o a una lógica de funcionamiento confusa o poco adecuada? ¿Cómo la implementaría usted?

6. En general, ¿qué cambiaría de la aplicación? ¿Por qué? ¿Cómo realizaría dicho cambio?



7. En general y tomando como referencia otras aplicaciones que haya utilizado, ¿cree que la aplicación es intuitiva y fácil de usar para cualquier tipo de usuario?

.....
.....
.....

7 ► Posibles mejoras o ampliaciones del proyecto

Entre las posibles mejoras/ampliaciones que podremos aplicar a la aplicación serían las siguientes.

1. Sería muy interesante el poder disponer de una lista de ‘Consultas frecuentes’, con las que se pueda recuperar una determinada información de manera directa, de forma que éstas pudiesen ser editadas o eliminadas.
2. Debido a que la información que se necesita recuperar puede ser muy variable, resultaría muy útil incluir un pequeño constructor de consultas desde el que se pueda personalizar las consultas y añadir nuevas consultas a la lista de consultas frecuentes.
3. Los resultados de las consultas podrán mostrarse por pantalla y/o ser impresos.
4. Será necesario conocer la situación de los pagos de un determinado hermano o conseguir un listado con los hermanos que tengan un número determinado de cuotas pendientes.
5. Debido a que en la gestión de pagos pueden producirse situaciones especiales como serían la del abono parcial de un pago, modificación del lugar de pago, etc..., sería recomendable que nuestra aplicación permitiese incluir notas/aclaraciones relativas a la propia realización del pago.
6. Los hermanos necesitan que se les avise de alguna manera y con cierta antelación de los distintos actos organizados por la Hermandad: correo electrónico, whatsapp, etc. Por ello, se tratará de registrar toda la información relativa a teléfono y redes sociales de cada hermano.
7. Sería muy útil que nuestra aplicación implementase la automatización de la elaboración de correspondencia (sobre+membrete de la carta) dentro del proceso de envío de correspondencia a la totalidad de hermanos a un conjunto determinado de ellos que podamos definir según necesidad.

8 ► Conclusiones personales

El contacto con el cliente es horrible. El cambio de requisitos y funcionalidad varía diariamente, con lo cual no se puede llevar a cabo una buena planificación y organización con el proyecto. Además, los conocimientos sobre el sector del cliente empeoran mucho más la situación puesto que está mucho más inmerso en el proyecto y quiere ser partícipe de cada decisión tomada.

En cuanto a la programación, no difiere mucho de la realización de cualquier práctica del ciclo puesto que básicamente comenzamos con un volumen de dudas, el cual no sabemos ni cómo abarcar, cuando encontramos un camino por el cual comenzar a programar, aparecen innumerables problemas y errores que nos devuelven al punto de partida, y, cuando por fin parecen que dejamos atrás las dudas, problemas y demás, hay algo que no nos convence y



volvemos al punto de partida. Pues realizar una aplicación para otra persona es básicamente eso, solo que en el último punto, el cliente se encarga de que tengas que volver a empezar.

Siendo más específica, Alfonso es profesor de Diseño de Interfaces Web y, por tanto, para él, una de las cosas más importantes es la interfaz de la aplicación. Y esto quizás es lo que me ha conllevado más tiempo. Hemos utilizado una plantilla para la aplicación, pero quizás esto ha dificultado un poco más el desarrollo puesto que, para adaptarla a nuestro gusto, había que repasar el código hasta encontrar el sitio exacto donde teníamos que poner lo que queríamos y esto, al final, no son más que horas y horas invertidas.

En conclusión, no ha sido una experiencia tan mala. La relación con el cliente ha sido bastante “especial” pero, al mismo tiempo, ha sido interesante combinar todo lo aprendido en el ciclo y ser capaz (o al menos intentarlo) de solventar los problemas que surgen.

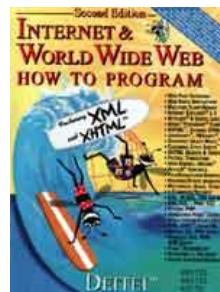


Bibliografía



Creación de aplicaciones Web con PHP 4

Tobias Ratschiller y Till Gerken,
Pearson Educación, Madrid 2001



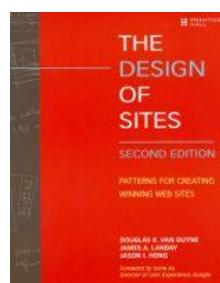
Internet and World Wide Web How to Program

H.M. Deitel, P.J. Deitel y T.R. Nieto,
Prentice Hall Second Edition
Upper Saddle New Jersey USA 2002



No me Hagas Pensar

Steve Krug, Segunda Edición,
Pearson Educación, Madrid España
2006



The Design of Sites

Douglas K. Van Duyne, James A. Landay
and Jason I. Hong
Second Edition, Prentice Hall, USA 2007



Ajax Web 2.0 para profesionales

Maximiliano Firtman, Alfaomega
Grupo Editor, Primera Edición,
México D.F. 2008



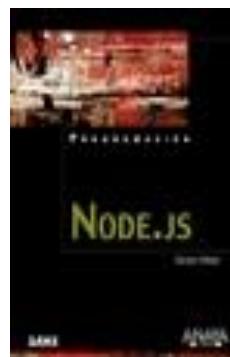
Diseño web con CSS

Ralph Schulz, Alfaomega Grupo
Editor, Primera Edición, México D.F.
febrero 2009



Arrancar con HTML5. Curso de programación

Emmanuel Herrera Ríos
Primera Edición, Alfaomega Grupo
Editor, México D.F. octubre 2011

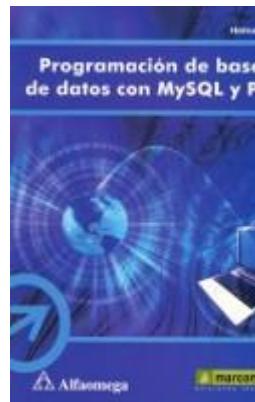


NODE.js

George Ombo
Anaya Multimedia, 2013



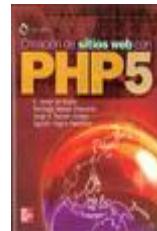
CURSO DE PROGRAMACION WEB: CON HTML5, CSS, JAVASCRIPT, PHP 5/6 Y MYSQL
Scott McCracken, Inforbooks
Ediciones, 2011



Programación de bases de datos con MySQL y PHP
Helma Spona
Primera Edición en Español Alfaomega
Grupo Editor, México D.F. junio 2010



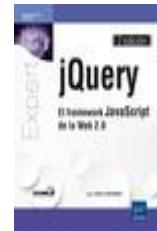
PHP Y MYSQL práctico: para DISEÑADORES y PROGRAMADORES WEB
Ellie Quigley
Anaya Multimedia, 2007



Creación de Sitios Web con PHP 5.1.
Javier Rubio
McGraw-Hill /
Interamericana de
España, S.A., 2005



JAVASCRIPT y JQUERY
David Sawyer
McFarland
Anaya
Multimedia, 2012



JQUERY (2ª ED.): El FRAMEWORK JAVASCRIPT de la Web 2.0
Luc Van Lancker
Eni, 2014



Fuentes de Internet



Manual framework Codeigniter
http://escodeigner.com/guia_usuario/



Guía Jquery UI
<http://jqueryui.com/>



Guía Bootstrap
<http://getbootstrap.com/>



Netbeans
<https://netbeans.org/>



Filezilla
<https://filezilla-project.org/>



MySQL Workbench
<https://www.mysql.com/products/workbench/>



PHP 5 Tutorial - W3Schools
www.w3schools.com/php



Desarrollo Web
<http://www.desarrolloweb.com/php/>



GitHub
<https://github.com/>



Carátula CD





Carátula Interior CD

