Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de una plataforma móvil para la visita turística de las Fallas de Valencia

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Franco Julián García Montero

Tutor: Pedro José Valderas Aranda

2016-2017

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

Agradecimientos

A Pedro José Valderas, por haberme ayudado y orientado a la hora de realizar este proyecto.

A mis padres y a mi novia, por apoyarme en los momentos más duros durante la carrera y por aguantarme durante todo este tiempo.

A mis amigos y compañeros de la universidad, por compartir todos los buenos momentos juntos a lo largo de estos cuatro años. En especial a Adrián Fernández Cid, Alberto Pérez Tolosa y Danny Yang Lee.

Resumen

La temática del presente proyecto es el análisis, diseño e implementación de una plataforma móvil para indicar las distintas ubicaciones durante las fiestas conocidas como Las Fallas, tales como las posiciones de los monumentos falleros, zonas más populares durante estas fiestas e información de distintos eventos. Su uso aporta la ventaja de proporcionar al usuario información de estas fiestas mediante el uso de la geolocalización.

Palabras clave: geolocalización, JqueryMobile, PHP, HTML, aplicación.

Abstract

The subject matter of this project is the analysis, design and implementation of a mobile plataform to indicate differents positions during holidays known as Las Fallas, such as positions of monuments falleros, popular zones in this holidays and information of different events. It's use has the advantage of providing the user with information about these holidays through the use of geolocation.

Keywords: geolocation, JqueryMobile, PHP, HTML, aplication.

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

Resumen

La temàtica del present projecte es l'anàlisis, disseny i implementació d'una plataforma mòbil per a indicar les diferents ubicacions durant les festes conegudes com Les Falles, com ara les posicions dels monuments fallers, zones mès populars durant aquestes festes i informació de diferents esdeveniments. El seu ús aporta l'avantage de proporcionar a l'usuari informació de aquestes festes mitjançant l'us de la geolocalització.

Paraules clau: geolocalització, JqueryMobile, PHP, HTML, aplicació.

TABLA DE CONTENIDOS

1. INTRODUCCIÓN 10
1.1. Motivación y justificación 10
1.2. Objetivos del proyecto 10
1.3. Estructura de la memoria 11
2. ESTADO DEL ARTE 13
3. METODOLOGÍA 15
4. ARQUITECTURA DEL SISTEMA 16
5. TECNOLOGÍAS UTLIZADAS 18
5.1. *Aptana Studio 3* 18
5.2. XAMPP 18
5.3. *PHP* 18
5.4. *JavaScript y jQueryMobile* 19
5.5. *MySQL* 19
5.6. *HTML* 19
5.7. *CSS* 20
5.8. *Visual Studio Code* 20
5.9. *Justinmind* 20
5.10. *Google maps Javascript* 21
5.11. *Google form* 21
5.12. *Datepicker widget* 21
6. ANÁLISIS DE NECESIDADES 22
7. DISEÑO 27
7.1. Mockup 27
7.2. Diseño de la base de datos 31
8. IMPLEMENTACIÓN 33
8.1. Servicios de *google maps* 33
8.2. Almacenamiento de los monumentos 33
8.2.1. saveObra.php 36
8.3. Almacenamiento de coordenadas 37
8.4. Almacenamiento de votaciones 39

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

8.5. Obtener votaciones 41
8.6. Obtener usuarios estandartes 42
8.7. Obtener coordenadas estandartes 43
8.8. Funcionamiento del calendario 44
9. CONCLUSIONES 46
BIBLIOGRAFÍA 48
ANEXO A: MANUAL DE USUARIO 50
APLICACIÓN ESTANDARTE 50
APLICACIÓN CLIENTE O TURISTA 51

TABLA DE ILUSTRACIONES

Ilustración 1: Fallas Valencia 2017 ..... *13*
Ilustración 2: Live Fallas *14*
Ilustración 3: Esquema metodología DCU *15*
Ilustración 4: Esquema estructura del proyecto *17*
Ilustración 5: Resultado primera cuestión *22*
Ilustración 6: Resultado segunda cuestión *22*
Ilustración 7: Resultado tercera cuestión *23*
Ilustración 8: Resultado cuarta cuestión *23*
Ilustración 9: Resultado quinta cuestión *23*
Ilustración 10: Resultado sexta cuestión *24*
Ilustración 11: Resultado séptima cuestión *24*
Ilustración 12: Resultado octava cuestión *25*
Ilustración 13: Resultado novena cuestión *25*
Ilustración 14: Diseño vista índice y vista monumentos *27*
Ilustración 15: Diseño vista clasificaciones *28*
Ilustración 16: Diseño vista información *28*
Ilustración 17: Diseño vista zonas populares y vista calendario *29*
Ilustración 18: Diseño vista buscador de estandartes *29*
Ilustración 19: Diseño vista inicio de sesión y compartir ubicación *30*
Ilustración 20: Diseño base de datos *31*
Ilustración 21: Página Login estandarte *50*
Ilustración 22: Página compartir ubicación *51*
Ilustración 23: Página índice *52*
Ilustración 24: Página monumentos *52*
Ilustración 25: Página votación monumentos *53*
Ilustración 26: Página clasificación fallas *54*
Ilustración 27: Página información *54*
Ilustración 28: Página buscar estandarte *55*
Ilustración 29: Página ubicación estandarte *55*
Ilustración 30: Página zonas populares *56*
Ilustración 31: Página calendario *57*
Ilustración 32: Página calendario con evento *57*

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

1. INTRODUCCIÓN

1.1. Motivación y justificación

Las Fallas son unas fiestas que hoy en día la mayoría de los españoles conocen pero que a nivel tecnológico no están muy actualizadas. La gente que reside en Valencia y que conocen estas fiestas, saben dónde y cuándo se realizan los diferentes eventos como las "mascletàs" o "los castillos" (popularmente hablando). Sin embargo, aquellas personas que no residen en Valencia y planean visitar la ciudad para poder disfrutar de estas fiestas, desconocen dicha información.

El interés de querer facilitar la información sobre las Fallas ha sido nuestra principal motivación a la hora de realizar este proyecto, ya que en los últimos años el turismo se está incrementando y esta aplicación facilitaría la "orientación" a los turistas e incluso a los mismos residentes de la ciudad. El realizar este proyecto me ha dado la posibilidad de trabajar con una tecnología que apenas conozco, pero que me atrae en gran medida y que además, me ha permitido aprender nuevos lenguajes, como puede ser PHP *Hypertext Pre-processor* (PHP), el cual está siendo actualmente bastante utilizado en el mundo laboral.

1.2. Objetivos del proyecto

El objetivo principal de este proyecto es el desarrollo de una plataforma móvil llamada "Fallas Mobile" que consta de dos aplicaciones móviles para proporcionar información sobre las Fallas, y además mediante geolocalización las ubicaciones de distintos eventos, incluyendo los pasacalles. Para ello, una primera aplicación estará destinada al turista general y los objetivos específicos respecto a ella son:

1. Proporcionar un mapa en el cual indicar las distintas ubicaciones de los monumentos de las Fallas, así como la posición de los estandartes que representan cada "falla".
2. Proporcionar un calendario dónde especifique el lugar y la hora de los distintos "eventos falleros" durante las fiestas.
3. Facilitar información cultural sobre las Fallas y su origen.
4. Garantizar orientación al usuario a la hora de utilizar la aplicación.

En cuanto a la segunda aplicación móvil, estará destinada a ser utilizada por los encargados de llevar los estandartes de cada Falla, con el fin de enviar en todo momento su posición y saber en tiempo real por dónde circulan los pasacalles. El objetivo principal de esta aplicación es:

- Permitir al usuario compartir y parar de compartir su ubicación a través de la aplicación móvil.

Por último, como objetivos secundarios se plantean:

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

1. Aplicar los conocimientos aprendidos en algunas asignaturas de la carrera, haciendo hincapié en *Base de datos*.
2. Estudiar y aprender de forma autodidáctica, con la ayuda de internet y otros documentos [páginas], el lenguaje de programación de código del lado del servidor *PHP*.
3. Aplicar los conocimientos aprendidos en *Desarrollo web*.

1.3. Estructura de la memoria

Este documento se estructura en nueve capítulos. El primero, es esta presente introducción que hemos dividido en tres apartados:

1. Dejar de una manera clara y concisa cual es la justificación y la motivación que nos ha llevado a la realización del presente proyecto.
2. Cuáles son los principales objetivos que hemos abarcado y como hemos estructurado nuestra memoria.
3. Facilitar y ampliar la información que refleja la tabla de los contenidos permitiendo una localización rápida de los diferentes apartados.

El segundo de los capítulos, abarca el estado del arte. En este apartado se detalla la funcionalidad que aporta nuestra aplicación y además se muestran diferentes aplicaciones relacionadas con la de este proyecto.

A continuación, en el tercer capítulo, se describe la metodología empleada en el proyecto, sus procesos y porqué hemos utilizado dicha metodología.

El cuarto capítulo, engloba la arquitectura del sistema dónde se definen los elementos que participan, cómo se comunican entre ellos y que tipos de información se envían entre ellos.

A continuación, en el quinto capítulo, se realiza un estudio de las diferentes tecnologías y herramientas utilizadas para el desarrollo de la aplicación móvil, explicando el porqué de cada una de las elecciones.

El sexto capítulo, detalla un análisis de necesidades dónde se reflejan los resultados de cuestionarios además de la definición de una “persona” en la cual nos centraremos a la hora del desarrollo de la aplicación.

El séptimo capítulo, abarca todo el diseño e implementación de lo que es la aplicación móvil. En él se detallan los bocetos de las interfaces de la aplicación móvil y el diseño de la Base de Datos con sus respectivas relaciones y atributos.

En el octavo capítulo, se muestran ejemplos de código de cada parte del sistema, así como la conexión al sistema de Base de Datos, código del servidor, interfaces... etc. Y en el noveno y último capítulo, se ven reflejadas las conclusiones.

Además, incluye referencias bibliográficas y los sitios web visitados que han hecho posible la realización, tanto de la memoria como de la aplicación móvil.

Por último, en el anexo se incluye un manual de usuario de la plataforma, que ayudará en gran medida a estos en el manejo total de la misma, en otras palabras, se explicará detalladamente cual es el funcionamiento de la aplicación para los distintos tipos de usuario existentes.

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

2. ESTADO DEL ARTE

En el actual proyecto llevado a cabo, se ha estudiado el mercado relacionado con “Fallas Mobile”. La investigación consistía en buscar aplicaciones relacionadas con las Fallas, y en el caso de encontrar alguna aplicación, estudiar su funcionalidad y ver qué servicios puede aportar. En nuestro caso, ofrece información geolocalizable y un sistema de votos para clasificar dichos monumentos en función del gusto popular. Esta información geolocalizable muestra los diferentes monumentos falleros que se encuentran en Valencia y además la posición en vivo de los diferentes falleros encargados de llevar los estandartes durante los desfiles.

Las aplicaciones que hemos encontrado son dos: “Fallas Vàlencia 2017” y “Live Fallas”. “Fallas Vàlencia 2017” ha sido actualizada funcionalmente y han incorporado bastante de las ideas que se han descrito: la ubicación de los monumentos, descripción y fecha de los diferentes eventos de fallas (Nits del foc, Mascletàs, etc....), una lista para guardar los monumentos favoritos e incluso información sobre los lugares más simbólicos y turísticos de Valencia (Catedral de Valencia, Bioparc, Basílica de la virgen de los desamparados, etc....).

Ilustración 1: Fallas Valencia 2017

“Live Fallas” es la segunda aplicación que hemos encontrado. Sin embargo, la aplicación no funciona correctamente y no carga más allá del menú principal. Pero con ver el menú, hemos podido observar que ofrece servicios similares a “Fallas Vàlencia

2017" además, de un sistema de votación y de la posibilidad de añadir fotos a la aplicación.

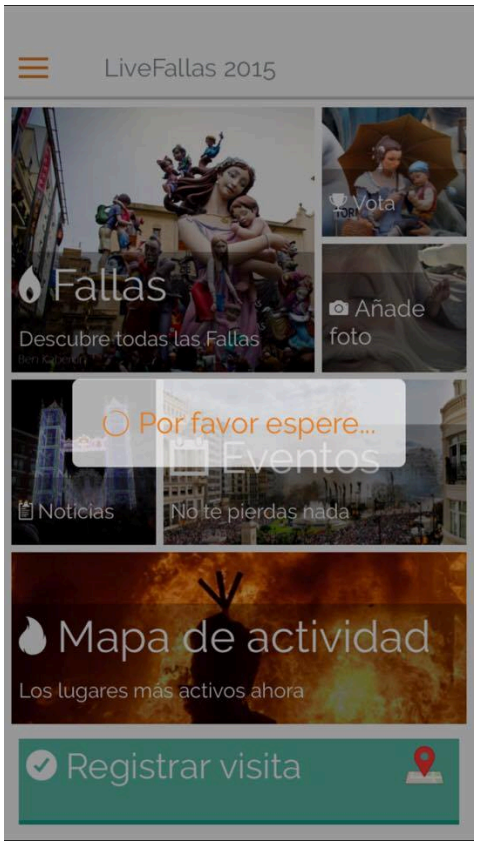


Ilustración 2: Live Fallas

Nuestra aplicación proporcionará una funcionalidad parecida a las dos aplicaciones mencionadas, pero, con la novedad de poder mostrar en un mapa, la ubicación de los usuarios portadores de los estandartes durante las Fallas.

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

3. METODOLOGÍA

Para llevar a cabo este proyecto, hemos hecho uso de una metodología aprendida en una de las asignaturas de la carrera de ingeniería informática partidas en la rama de tecnologías de la comunicación, llamada “Diseño Centrado en el Usuario” (*DCU*). El Diseño Centrado en el Usuario [12] es definido por la UPA (*Usability Professionals Association*) como un enfoque de diseño cuyo proceso está dirigido por información sobre las personas que van a hacer uso del producto.

El proceso que sigue esta metodología es la siguiente:

1. Análisis de necesidades de usuario. Consiste en identificar a las personas a las que se dirige el producto, para qué lo usarán y en qué condiciones, con el fin de determinar los objetivos del usuario.
2. Diseño. Definir los diseños de software que se ajusten a las necesidades de los usuarios considerando la usabilidad de los mismos como pilar fundamental.
3. Evaluación. Se validan las soluciones de diseño o por el contrario se detectan problemas de usabilidad o de requisitos no satisfechos.

Esto quiere decir que mediante esta metodología se quiere lograr la satisfacción de las necesidades de todos los usuarios potenciales, adaptar la tecnología utilizada a las expectativas de estos usuarios y crear interfaces que faciliten la consecución de sus objetivos.

En este proyecto hemos optado por usar Diseño Centrado en el Usuario ya que fue utilizada en la carrera y tenemos experiencia con dicha metodología.

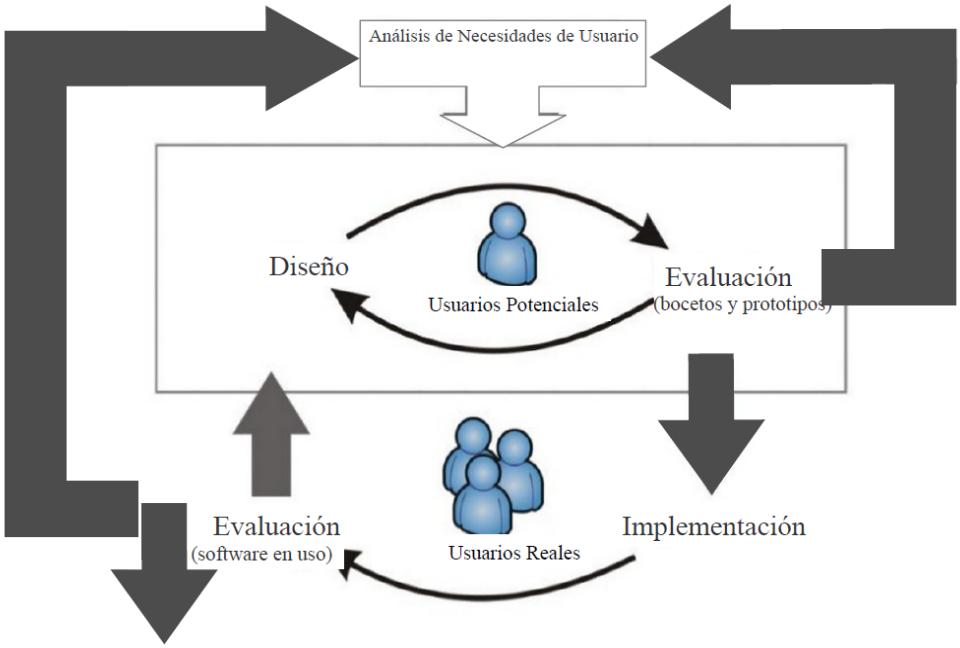


Ilustración 3: Esquema metodología DCU

4. ARQUITECTURA DEL SISTEMA

La arquitectura de nuestro proyecto está compuesta de la siguiente manera: dos servicios, una base de datos, y dos aplicaciones desarrolladas por nosotros.

En primer lugar, hay dos tipos de servicios que usamos. Una de ellas la obtenemos de *google maps*. Este servicio es un conjunto de interfaces HTTP que proporciona los datos geográficos en nuestra aplicación. Por otra parte, el servicio del que hacemos uso es una que nos proporciona el ayuntamiento de valencia [11]. Éste último, nos ofrece información completa de los distintos monumentos falleros. Tal como nombres, coordenadas, distancias... etc. Toda esta información la obtenemos en formato JSON.

En segundo lugar, tenemos nuestra base de datos. En ella se almacenan un sistema de votación popular y por otro lado, almacena las coordenadas compartidas por el usuario portador del estandarte fallero.

Por último, tenemos las dos aplicaciones. La primera está destinada a los turistas con la información de las Fallas, mientras que la segunda se trata de un *Login* dónde los usuarios portadores de estandartes, iniciarán sesión para compartir su ubicación. Ambas, usaran peticiones *GET* y *POST* de *HTTP* para comunicarse tanto con los servicios cómo con la base de datos. Con la petición *GET* realizaremos solicitudes para obtener información, mientras que con *POST* se usará para insertar información en la base de datos.

El funcionamiento del sistema es el siguiente. Por parte de los usuarios que llevan los estandartes, la aplicación se encargará de que dichos usuarios inicien sesión para poder identificarlos y que después puedan compartir su ubicación. Con una petición *GET*, obtendrá el servicio de *google* para poder conseguir las coordenadas. Estas coordenadas serán enviadas a la base de datos mediante una petición *POST*.

Por otro lado tenemos la aplicación de los turistas. Esta aplicación consta de una petición de tipo *POST* y tres de tipo *GET*. La de tipo POST se usa para poder almacenar en base de datos las votaciones de la aplicación. En cuanto a las peticiones GET las podemos dividir de la siguiente forma:

1. Una de ellas es para obtener los datos de los monumentos falleros gracias al servicio que nos proporciona el ayuntamiento.
2. Otra nos sirve para conseguir los resultados de las votaciones realizadas a través de la aplicación.
3. Por último nos proporciona las distintas posiciones de los pasacalles.

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

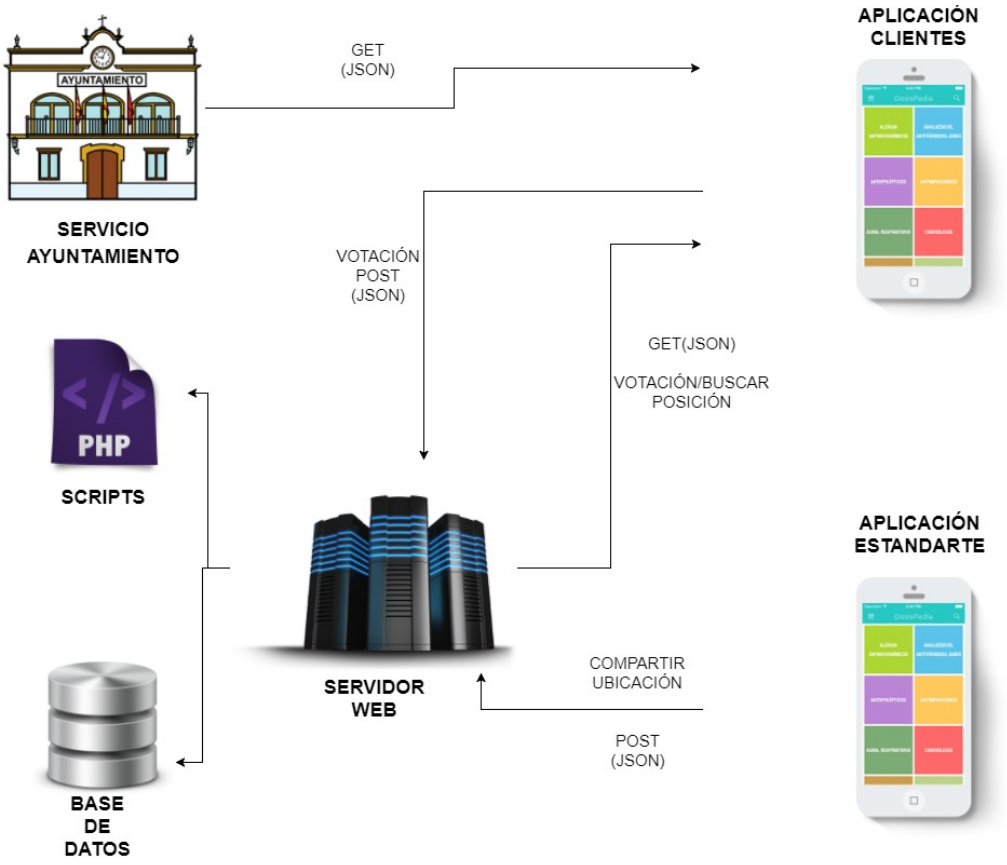


Ilustración 4: Esquema estructura del proyecto

En cuanto a la arquitectura, ambas aplicaciones están formadas por tres capas o niveles:

- El primer nivel, consiste en la capa de presentación. En otras palabras, la vista. Esta capa se encarga de mostrar los datos al usuario con una presentación amigable y fácil de usar en un formato coherente y adecuado.

- El segundo lugar, está formado por la capa de negocio. Aquí es dónde reside toda la lógica de la aplicación, es decir, toda la funcionalidad como en este caso los scripts en javascript o *PHP*. Aquí, se reciben todas las peticiones del usuario, enviando posteriormente las respuestas tras el proceso.

- Por último, nos encontramos con la capa de datos. De aquí se extraen todos los datos solicitados o se inserta la información desde la capa de negocio a la Base de Datos.

En otras palabras, la aplicación muestra la información al usuario (primera capa), el usuario solicita información de los datos o los modifica (segunda capa y tercera) y el resultado obtenido de esta solicitud, se muestra en la vista del usuario.

5. TECNOLOGÍAS UTLIZADAS

Uno de los primeros pasos a realizar era la preparación del entorno, por lo que instalamos una aplicación denominada *Aptana Studio 3*. Este programa nos permite la ejecución de nuestra aplicación móvil. Sin embargo, tras llegar a la mitad del proceso nos dimos cuenta de que *Aptana Studio 3* no implementaba *Apache*, esencial para poder ejecutar los archivos *PHP* y poder realizar peticiones a bases de datos. Por lo tanto, además del uso de *Aptana Studio 3*, tuvimos que hacer uso de *XAMPP*.

5.1. *Aptana Studio 3*

Aptana Studio 3 [2] es un entorno de desarrollo integrado (IDE) que nos permite poder ejecutar nuestros proyectos en pleno desarrollo. Es una aplicación gratuita y la hemos utilizado para poder desarrollar código en *Javascript*, *CSS*, *HTML y PHP*.

Además de facilitarnos un entorno de desarrollo, también nos facilita mucho la ejecución de nuestro proyecto en marcha. También sirve como editor de texto y de código fuente, pero en mi caso, para realizar la edición de código se ha utilizado *visual studio*.

5.2. XAMPP

XAMPP [3] es una distribución de *Apache* gratuita, fácil de instalar, configurar y usar, además de ser capaz de interpretar páginas dinámicas, que contienen tanto PHP como *MySQL*, es decir, lo que necesitábamos para poder realizar las peticiones a bases de datos. Esta distribución de *Apache* solo requería la descarga y ejecución de un archivo *ZIP*, con algunas configuraciones en sus componentes para poder integrarla junto con *Aptana Studio* y ejecutar apache junto con la aplicación ejecutada en el entorno de desarrollo.

Otra de las ventajas que tiene *XAMPP* es su actualización regular de *Apache*, *MySQL*, *PHP y Perl*, lo que permite evitar errores. Además incluye *phpMyAdmin*, dónde en nuestro caso, es esencial para poder realizar el diseño, la creación, y la gestión de la base de datos.

5.3. *PHP*

PHP [4] es un lenguaje de programación de uso general de código del lado del servidor. Esto significa que es el servidor el único que se encarga de ejecutar el código y enviar su resultado.

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

En nuestro caso, no hemos hecho una conexión directa entre *PHP* y *HTML*, es decir, los resultados obtenidos desde base de datos no han sido mostrados directamente en las vistas de la aplicación, sino que hemos usado *javascript* de por medio para poder editar los resultados y poder moldearlos en función de nuestros objetivos.

Por otra parte, otras de las razones por la que hemos usado *PHP* ha sido su parecido con los lenguajes de programación estructurada como C, java o *javascript*.

5.4. *JavaScript y jQueryMobile*

JavaScript [5] es un lenguaje interpretado, que no requiere compilación. Es similar a Java aunque no es un lenguaje orientado a objetos, y no dispone de herencias. Este código ha sido integrado en todas las vistas de nuestra aplicación. Cada vista tiene un JavaScript, dónde en función de la vista, tiene una funcionalidad distinta.

Asimismo también hemos utilizado la biblioteca (o *framework*) jQueryMobile [6]. Éste a su vez, hace uso de jQuery. La característica principal de jQuery y también una de sus mayores ventajas es permitir el cambio de contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX, utilizando las funciones $() o jQuery(). JQuery, se centra más en los navegadores, mientras que jQueryMobile, está más enfocado hacia los sitios móviles. Además no es solo la funcionalidad, sino que la interfaz también hace apoyo a estos sitios móviles ayudándonos en el aspecto de la aplicación que hemos desarrollado.

5.5. *MySQL*

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario de código abierto, que tiene una alta velocidad a la hora de realizar ciertas operaciones, lo que hace que sea uno de los gestores de bases de datos con mejor rendimiento. Su conectividad, velocidad, y seguridad permiten que el servidor de MySQL sea altamente apropiado para acceder a bases de datos en internet.

La razón por la que elegimos este gestor fue su utilización en algunas de las asignaturas en estos últimos años, es decir, gracias a la familiarización que manteníamos con él mismo, a su rapidez y a su perfecta integración en *PHP*.

5.6. *HTML*

HTML hace referencia al lenguaje de marcado para la elaboración de páginas web, que define una estructura básica y un código para la definición de contenido de una web, como texto, imágenes, etc.

En nuestro caso era imprescindible su uso puesto sin él no hubiéramos podido realizar las diferentes vistas de las que consta nuestro proyecto, ya que todas ellas están realizadas en *HTML*.

5.7. *CSS*

Cascading Style Sheets (CSS) es un lenguaje de hojas de estilo utilizado para describir el aspecto y el formato de un documento escrito en un lenguaje de marcas como puede ser *HTML*. Su uso era necesario puesto que sin él no hubiéramos conseguido todos los detalles incorporados en nuestra aplicación web y no hubiéramos podido controlar el estilo y el formato de las múltiples páginas al mismo tiempo. Así como tampoco hubiéramos tenido la flexibilidad de cambiar en cualquier momento alguna parte o la totalidad del diseño de nuestras páginas con tan solo modificar nuestra hoja de estilo, sin que ello suponga modificar el contenido.

Además otro detalle importante por el que era necesario utilizar *CSS*, era por el uso con *Jquery Mobile*. Éste nos permitía apoyo a la hora de poder realizar la aplicación. Cuándo hablamos de apoyo, hablamos de orden o ubicación de los distintos elementos en la vista. Sin embargo esto no es suficiente, ya que tiene colores y tamaños estandarizados. Con *CSS* hemos podido cambiar los colores, tamaños y posiciones de los diferentes elementos de la vista.

5.8. *Visual Studio Code*

Visual Studio Code [8] es, al igual que *Aptana Studio 3*, un IDE que soporta múltiples leguajes de programación. Tales como: *C++*, *C#*, *Visual Basic*, *.NET*, *Java*, etc...

En nuestro caso no haría falta el uso de éste, ya que con *Aptana Studio*, sería suficiente para la edición de código. Sin embargo, Visual Studio aporta comodidad a la hora de programar. La tipografía que utilizan y la combinación de colores en la interfaz que usan es mucho más atractiva que la de *Aptana Studio*, por ejemplo el fondo oscuro que usa *Visual Studio*. No solo es el fondo de la interfaz lo que resulta cómodo, sino también el uso de colores a la hora de desarrollar. *Visual Studio* pinta por defecto funciones y declaraciones de variables con un color distintivo, lo que hace que sea mucho más claro a la hora de leer el código.

5.9. *Justinmind*

Justinmind [9] es una herramienta de diseño utilizada para realizar prototipos de proyectos webs o aplicaciones móviles.

La razón por la cual hemos utilizado esta herramienta ha sido por su uso en una asignatura de la carrera. Aprovechamos tener conocimiento y experiencia con la

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

herramienta, para poder sacar un buen prototipo. Además de estos dos factores mencionados, es muy útil a la hora de realizar interacción con los usuarios. Ya que *Justinmind* permite aportar al prototipo, una mínima funcionalidad, para poder evaluar el comportamiento de los usuarios frente a la aplicación.

5.10. *Google maps Javascript*

Google maps Javascript es un servicio de terceros, en este caso de google, que nos permite hacer uso de ella para mejorar o ampliar la experiencia o funcionalidad de nuestra aplicación. En nuestro caso, hemos hecho uso de *google maps* para el uso de la geolocalizacion en el mapa que nos proporciona el mismo servicio.

Para poder hacer uso de ella, hemos tenido que obtener una clave o autenticación [10].

5.11. *Google form*

Google form es una herramienta online proporcionada por google que permite realizar cuestionarios de manera online y mostrar los resultados gráficamente. Esta herramienta ha sido utilizada para poder llevar a cabo la metodología DCU y conseguir opiniones de usuarios con respecto a la aplicación.

5.12. *Datepicker widget*

Resulta ser, que jQuery mobile no soporta dicho *widget* [16]. Por lo tanto, se ha tenido que usar unos archivos externos realizados por un tercero para que mezclado junto con jQuery, pudiera funcionar en *Jquery mobile.*

6. ANÁLISIS DE NECESIDADES

Para saber las características que iba a tener nuestra aplicación, se realizó un cuestionario con *google form* [13] para poder llevar a cabo la metodología DCU (Diseño Centrado en el Usuario). El cuestionario consiste en una serie de preguntas sobre distintas funcionalidades que se le podría otorgar a la aplicación móvil. El sondeo se ha distribuido por una red social en concreto, *Facebook*, para obtener la respuesta de los usuarios. Dicho cuestionario está formado por nueve preguntas y ha sido respondido por sesenta y dos personas.

En la primera pregunta, se hizo referencia a la edad del usuario. Se puede observar que un prototipo de persona debe ser de entre dieciocho y treinta años.

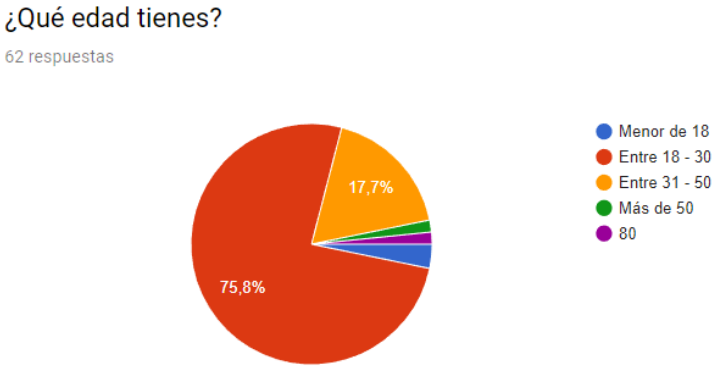


Ilustración 5: Resultado primera cuestión

En la siguiente nos interesaba saber si el usuario era un hombre o una mujer para seleccionar un tipo de género a la hora de generar la persona.

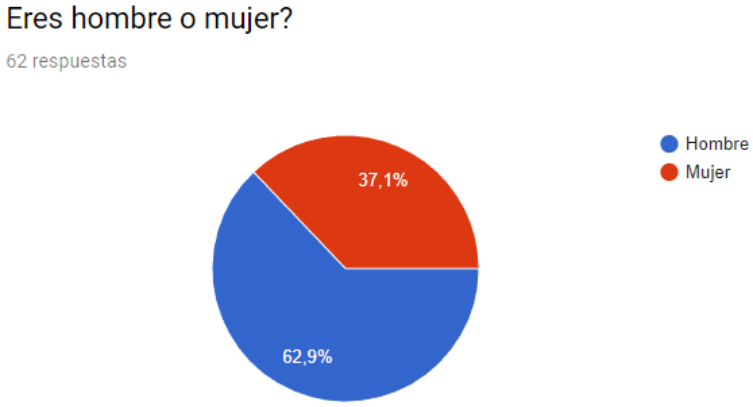


Ilustración 6: Resultado segunda cuestión

A continuación se deseaba saber si el usuario era o no fallero.

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

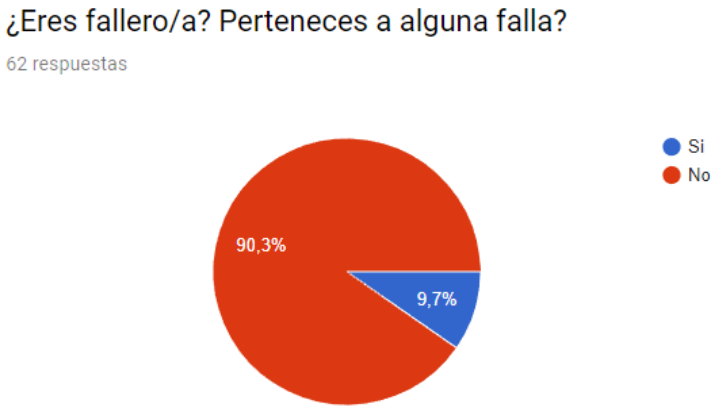


Ilustración 7: Resultado tercera cuestión

En la cuarta, buscamos saber si el usuario reside en Valencia para proporcionar información turística de la ciudad de Valencia.

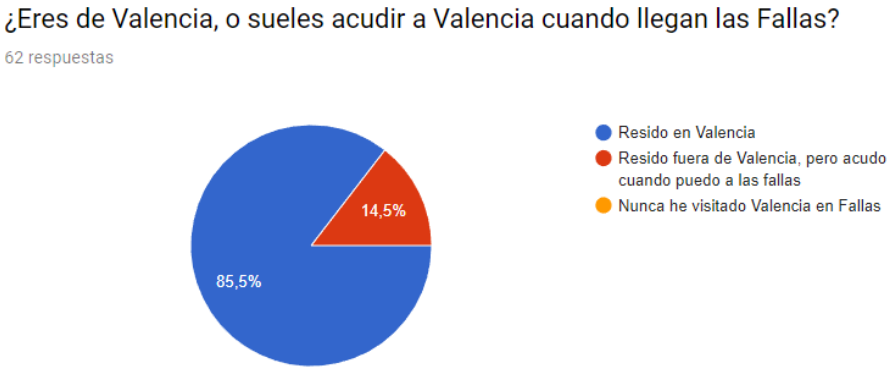


Ilustración 8: Resultado cuarta cuestión

En la quinta, se cuestiona si sería de utilidad que la aplicación tuviera algún apartado en dónde se indiquen las localizaciones de los monumentos falleros distribuidos por todo Valencia.

Ilustración 9: Resultado quinta cuestión

En la sexta, queremos averiguar si al usuario interesaría que en un mapa se mostrara uno o varios círculos con un determinado radio para indicar en dónde se suele encontrar más ambiente (música, comida, gente, etc.) durante estas fiestas.

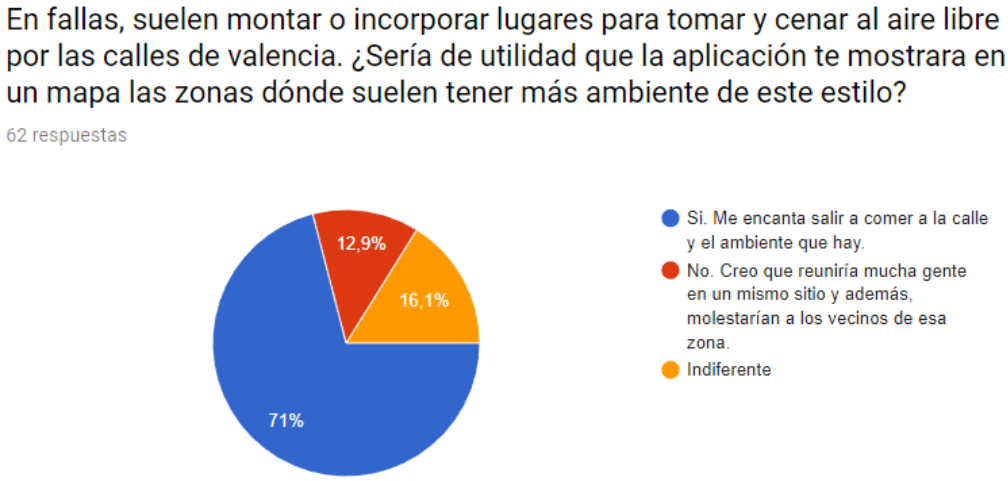


Ilustración 10: Resultado sexta cuestión

A continuación se pregunta a los usuarios que sean falleros si les resultaría interesante que en un mapa se muestre la localización de los pasacalles, para poder seguir el recorrido de estos junto con las falleras durante dichas fiestas.

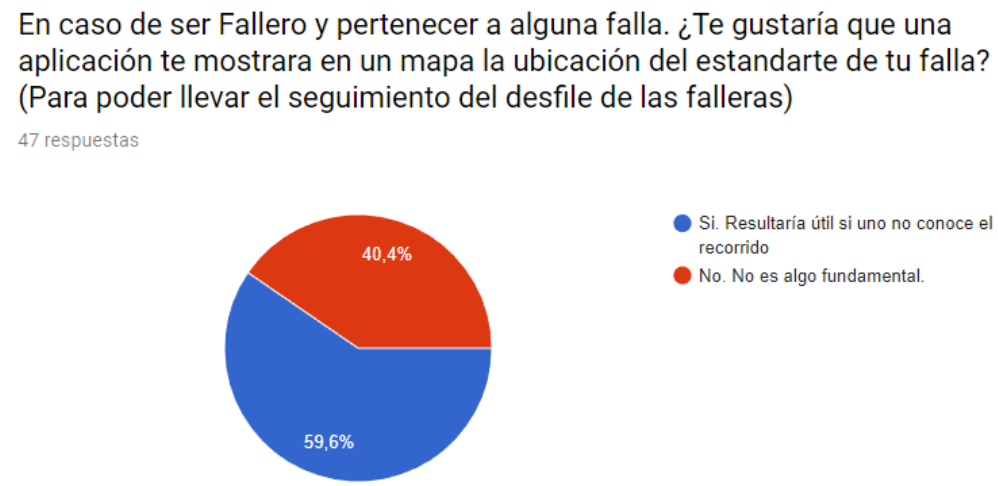


Ilustración 11: Resultado séptima cuestión

En la penúltima pregunta, hacemos referencia a cerca de un calendario que proporcione información sobre los distintos eventos falleros.

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

Ilustración 12: Resultado octava cuestión

Por último, se ofrece al usuario a posibilidad de que la aplicación tenga un apartado de votación y poder ser clasificados en función de dichas votaciones.

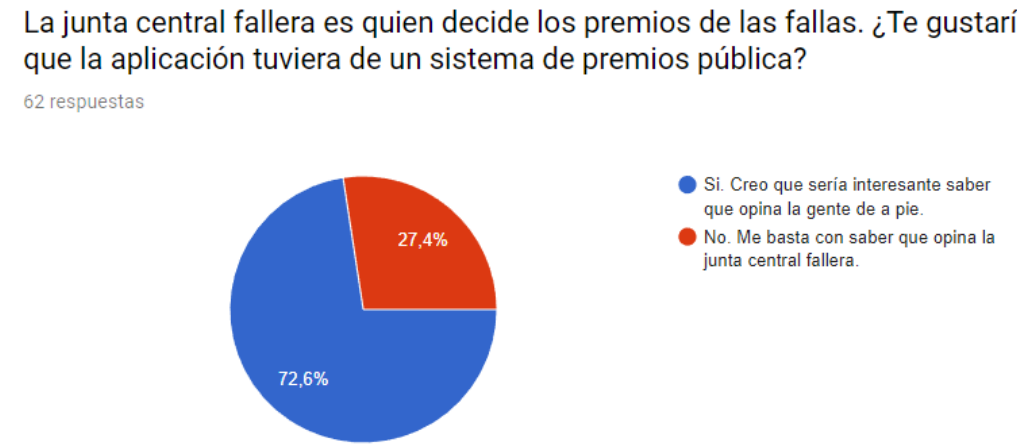
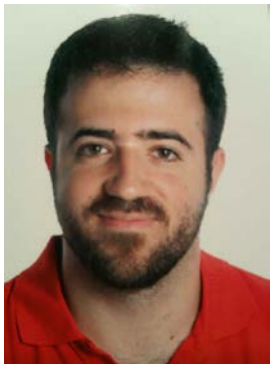


Ilustración 13: Resultado novena cuestión

El objetivo de este cuestionario, es plasmar nuestras ideas, y descartar opciones en función de la mayoría de los votos. Como resultado, obtenemos que a la mayoría de los usuarios que han participado en el cuestionario, están a favor de las distintas ideas presentadas. La única cuestión descartada es la tercera, en dónde se pregunta si reside o no en Valencia. Esta cuestión, era para incorporar en la aplicación, puntos de interés o simbólicos de la ciudad de Valencia. Esa idea fue rechazada debido a la gran cantidad de usuarios residentes de Valencia y para fijar un tipo de persona a la cual desarrollar la aplicación.

Con estos resultados, ya podemos ser capaces de poder crearnos una persona a la cual referenciarnos en el momento de desarrollar el proyecto. La persona es la siguiente:



Joan Pérez Lliri

Bibliografía:

- 23 años de edad
- Reside en Valencia
- Vive con sus padres actualmente
- Le gusta leer, la tecnología y practicar natación
- Siempre está con el teléfono móvil a mano
- Le encanta chatear y las redes sociales
- Pertenece a una falla

Salud

- Le gusta echarse la siesta constantemente
- No padece ningún tipo de problema físico o de movilidad
- No tiene ningún tipo de problema de visión

Tecnología

- Usa mucho el ordenador, tiene uno personal en su habitación
- Tiene teléfono móvil de gamma media-alta
- Escucha mucho la radio y la música con aplicaciones móviles

Objetivos

- Quiere estar en contacto con el mundo fallero
- Tener un punto de encuentro con su falla
- Saber qué lugares son los más divertidos o con más ambiente durante estas fiestas

Una vez originado nuestra persona, debemos crear una situación dónde nuestra persona alcance sus objetivos usando el producto que le ofrecemos. Los escenarios que vamos a desarrollar, serán unos específicos, ya que tenemos un diseño del proyecto definido.

Joan se levanta una mañana tarde en Fallas y todos sus compañeros de las fallas ya ha salido a desfilar. Por lo tanto, Joan saca el teléfono móvil y busca en la aplicación el nombre de su falla. La aplicación le muestra en vivo que su falla, está de camino al centro por la calle ruzafa. Joan consigue encontrarlos gracias a su ubicación compartida por el teléfono.

Joan se encuentra de vacaciones en Fallas. Sin embargo, no está actualizado con respecto a las últimas novedades y cambios de horarios de los eventos durante las fiestas. Joan coge el móvil y busca en el calendario, que días hay eventos especiales durante las Fallas

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

7. DISEÑO

En este apartado se va a detallar y mostrar los diferentes diseños realizados en este proyecto. Por una parte se han realizado *mockup* para el diseño de las aplicaciones y por otra parte el diseño de la base de datos creada para dichas aplicaciones.

7.1. Mockup

Para llevar a cabo el diseño y la implementación de nuestra aplicación, primeramente hicimos un *mockup* con *Justinmine* para poder ver las ideas principales que teníamos sobre la aplicación. Sin embargo, con el paso del tiempo y el desarrollo de la aplicación, muchos de estos diseños se fueron modificando en función de la aplicación.

Los esquemas de diseño dibujados con la herramienta nombrada en el párrafo anterior se muestran a continuación con una pequeña descripción de lo que se tenía en mente en un primer momento.

En el índice se le proporciona al usuario una lista de todas las funcionalidades que proporciona la aplicación de manera sencilla para que no le sea difícil encontrar aquella que le interese. En la vista de los monumentos se señalan las diferentes posiciones de las fallas en un mapa. Los marcadores, muestran las posiciones, que son insertadas en el mapa mediante coordenadas que son obtenidas a través del servicio web del ayuntamiento de Valencia. Como se puede observar, tiene dos botones dónde uno (“Atrás”) volverá al índice y el otro irá a la vista de clasificaciones de las fallas.

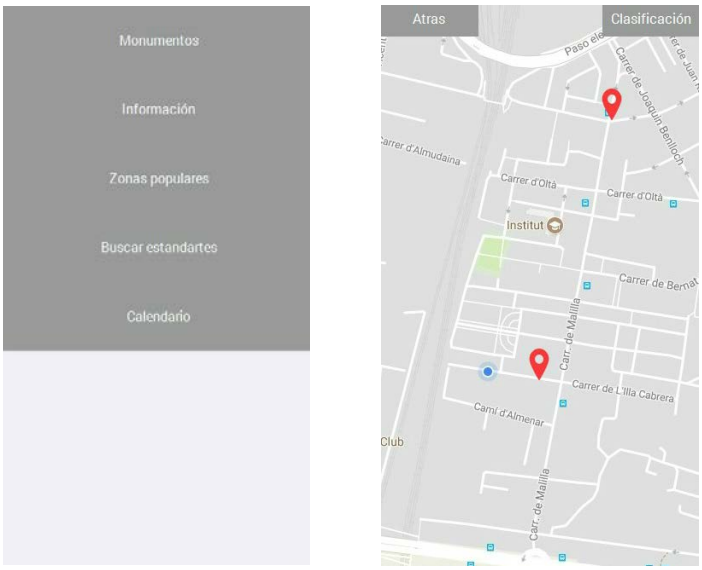


Ilustración 14: Diseño vista índice y vista monumentos

Aquí, se desea mostrar las clasificaciones de las diferentes fallas ordenadas en función de los votos realizados por los usuarios. Constará de una tabla con dos columnas, dónde en una mostrará el nombre de la Falla, y la otra, mostrará la calificación total. El primer botón (“Atrás”) volverá al mapa de monumentos, mientras que el de inicio volverá directamente al índice.

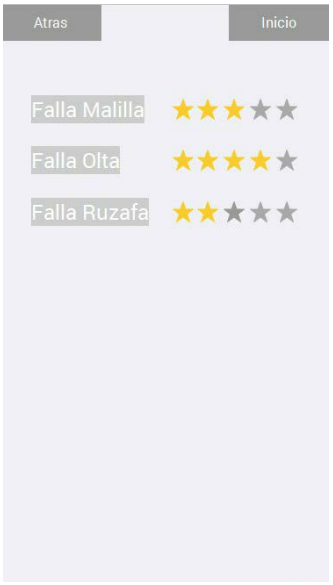


Ilustración 15: Diseño vista clasificaciones

En el diseño de esta vista se muestra información cultural sobre las Fallas. Consta de una lista de apartados, dónde cada uno de ellos tiene un subapartado en el cual se encuentra el desarrollo de cada una de las cuestiones.

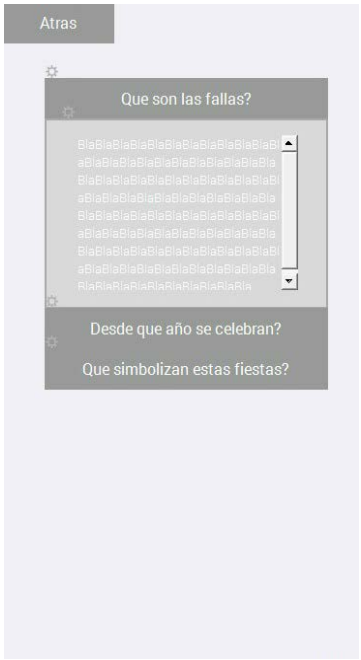


Ilustración 16: Diseño vista información

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

En esta vista de las zonas populares hemos optado por mostrar círculos de colores sobre el mapa, para representar las distintas zonas populares de Valencia durante las fiestas. A continuación en el apartado de calendario el usuario podrá buscar eventos en función de la fecha dónde esté marcado.

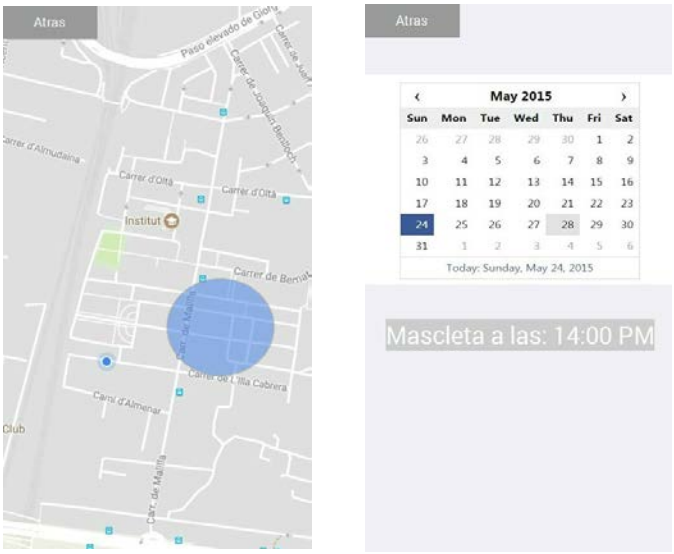


Ilustración 17: Diseño vista zonas populares y vista calendario

Aquí, se quiere enseñar al usuario las distintas Fallas que se encuentran disponibles. Las distintas Fallas se listan con sus respectivos nombres y mediante un *input*, buscarán la Falla que más les interese. Al seleccionar la Falla pasará a una vista dónde se señala en un mapa la correspondiente posición de dicha Falla. La posición será marcada mediante un círculo azul, al igual que la aplicación de *google maps*.

Ilustración 18: Diseño vista buscador de estandartes

Por otro lado, en la aplicación de los estandartes, se crearan dos vistas. Una de ellas será para iniciar sesión. Esta vista permitirá identificar a los distintos usuarios que vayan a compartir su ubicación.

En la segunda vista de la aplicación de los estandartes es dónde se realizará la compartición de su ubicación mediante dos botones. Tal y como se definen, uno compartirá la ubicación y el otro parará de compartirla en caso de que el usuario así lo desee. También nos encontramos con un tercer botón, que al igual que en la otra aplicación de cliente, tiene como función volver a la vista anterior, en este caso, al inicio de sesión.

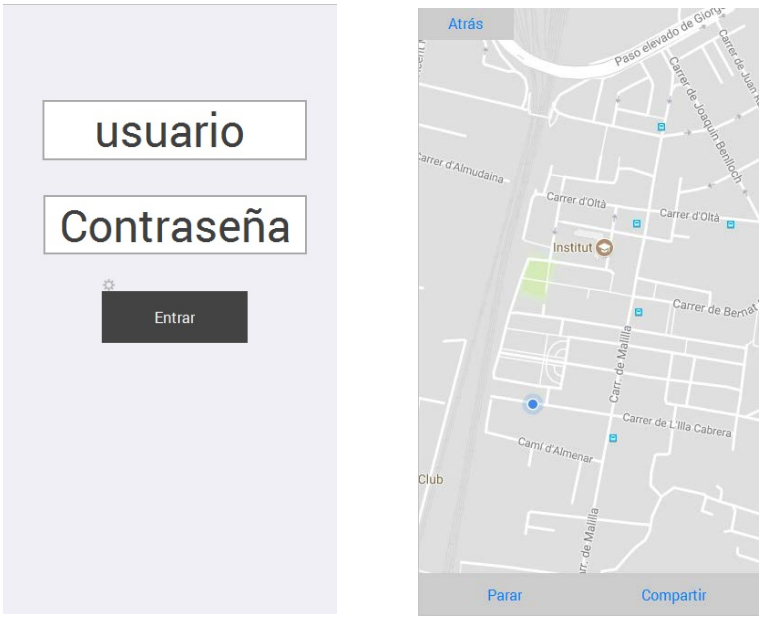


Ilustración 19: Diseño vista inicio de sesión y compartir ubicación

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

7.2. Diseño de la base de datos

El diseño de base de datos realizado para esta aplicación es simple, ya que no existe una relación directa entre las distintas tablas de la misma. La base de datos de nuestra aplicación denominada **fallas** está formada por dos tablas. Una de ellas se llama ***sharepositions*** en el cual se almacenan las coordenadas de los usuarios portadores de estandartes. Por otra parte, la otra tabla se llama ***votemonuments***. En ésta última se guardan las votaciones que se realizan a través de la aplicación.

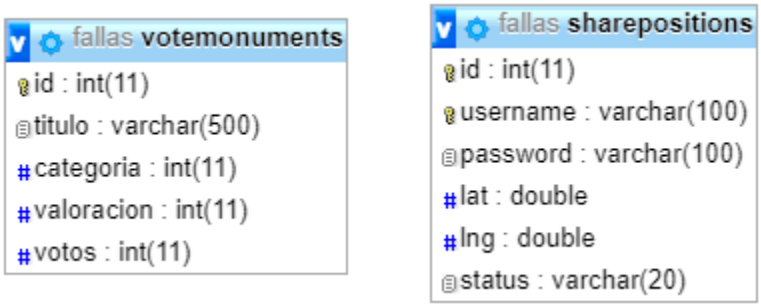


Ilustración 20: Diseño base de datos

sharepositions es una tabla formada por seis atributos:

- "*Id*": Es el identificador del usuario.
- "*username*": Es el nombre del usuario.
- "*password*": Es la contraseña del usuario.
- "lat": Contiene la latitud de la coordenada.
- "lng": Contiene la longitud de la coordenada.
- "*status*": Guarda el estado del usuario en "*ON*" u "*OFF*"

El atributo "*Id*" es de tipo entero el cual se encarga de tener un valor único en la tabla ya que es la clave primaria de esta tabla. Cada vez que se inserta un usuario, su valor aumenta en uno.

"Username" y "password" son dos atributos de tipo "string" (cadena de caracteres). Uno almacena el nombre del usuario, mientras que el otro almacenará la contraseña de este mismo usuario. Además esta contraseña se encuentra encriptado en la base de datos en SHA-1. Éste, es un método de cifrado que garantiza seguridad a la hora de intercambiar datos. Se ha utilizado SHA-1 porque si no se cifra, en la base de datos, se mostraría como un "string" normal y corriente pudiendo contemplar el texto sin problemas. Con este método se genera, con la palabra de la contraseña, una palabra totalmente aleatoria e ilegible.

"lat" y "lng" son de tipo "*double*", ya que se tratan de coordenadas con decimales. Tal y como se mencionó, estos atributos, almacenan la latitud y la longitud de las coordenadas del usuario portador del estandarte.

Por último, el atributo "*status*" se encarga de guardar el estado del usuario que comparte la ubicación. A la hora de compartirla, este estado pasará a ser "*ON*" y persistirá en ese valor hasta el momento en dónde el usuario desee detener esa función. Al detenerla, el "*status*" de ese usuario en la base de datos pasará a ser "*OFF*". Ésta lógica se ha utilizado para limitar la búsqueda del estandarte desde la otra aplicación, Ya que si no tuviéramos este atributo y el usuario deja de compartir la ubicación, desde la otra aplicación podría buscarse ese mismo estandarte y devolvería su última posición. Algo que no es correcto. Con esto, impedimos que eso ocurra.

Por otra parte, nos encontramos con ***votemonuments***. Esta tabla está dirigida especialmente hacia la aplicación de los clientes que votan los diferentes monumentos. Cada votación realizada, se almacenará en esta tabla y de ella misma se obtendrán los resultados en la tabla de clasificación.

votemonuments es una tabla formada por cinco atributos:

- "*Id*": Es el identificador del monumento.
- "*titulo*": Es el nombre del monumento.
- "*categoria*": Es la categoría a la que pertenece el monumento.
- "valoracion": Contiene la valoración total del monumento.
- "votos": Contiene el número de votos totales.

El atributo "Id", al igual que en la otra tabla, es el identificador del objeto. El "Titulo" es el nombre del monumento. Al ser un nombre, será atributo de tipo "*string*". La "categoría" es un atributo útil en el caso de que queramos obtener los diferentes monumentos en función de su categoría. La "valoración" y los "votos", al igual que la "categoría", son de tipo enteros. Estos dos atributos guardarán las distintas puntuaciones con las que sean valoradas por los usuarios. Además, con los valores de estos últimos atributos calcularemos la media para mostrarla en la vista.

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

8. IMPLEMENTACIÓN

En este apartado nos centraremos en mostrar y explicar diferentes bloques o trozos de código que se han implementado durante el proyecto.

8.1. Servicios de *google maps*

Para hacer uso de *google maps* se ha añadido el siguiente código en el *HTML:*

El *script* que aparece en la imagen es el que se encarga de solicitar a google, el servicio. Como se ha comentado, para hacer uso del servicio, era necesario pedir una *APIKEY* para poder tener acceso. En cuanto a los dos parámetros que aparecen en el *script* "async" y "defer", permiten una carga paralela junto con el archivo *HTML*. Cuándo se carga un archivo *HTML*, éste se carga línea por línea de código hasta que se encuentra la etiqueta "<*script*>". Una vez lo encuentra, la carga del *HTML* se detiene, y pasa a ejecutar el *script*. Una vez termina el *script*, el *HTML* continúa con su carga. "defer" permite q se cargue el *HTML* a la vez que se descarga el fichero *javascript*. Una vez ha finalizado de descargarse el *HTML*, el *javascript* se ejecuta. "async" por otra parte, no espera al *HTML*, se cargan a la vez ambos y al descargarse el *javascript* se ejecuta, indiferentemente si está o no cargado el *HTML*.

Este código ha sido incorporado en ambas aplicaciones, ya que el servicio era necesario para las dos aplicaciones de nuestro proyecto.

8.2. Almacenamiento de los monumentos

El ayuntamiento ofrece un servicio del cual se puede obtener información de los monumentos falleros en la ciudad de Valencia. El servicio, tiene la siguiente dirección: http://mapas.valencia.es/lanzadera/puntoInteres/fallasvalencia?radio=40000&lang=es&lat=39465212&lon=-374521&filtros=N. Sin embargo, para poder acceder, se necesita de unas credenciales que son otorgadas por el mismo ayuntamiento. Para solicitarlas, hay que entrar en su página oficial de datos abiertos [14] y una vez solicitada, es posible acceder al servicio para obtener los datos. Como se puede apreciar, en la URL, hay un campo "filtros" definido como "N". Esto es, porque se obtienen por secciones, es decir, por cómo están clasificadas las distintas fallas. Por lo tanto, este valor "N" puede tener cuatro valores distintos: 1, 2, 3, 4. El 1 nos devuelve las Fallas de categoría especial, el 2 las de categoría A, con 3 el resto de secciones y con 4 las Fallas innovadoras.

Aun así no nos ha ofrecido alguna forma de obtener todos los monumentos y así poder almacenarlos en base de datos. Por lo tanto, he necesitado hacer uso de este código, que solo se ha ejecutado una vez.

```
for (i = 1 ; i <= 4; i++) {
    $.ajax({
    type: "GET",
    url: "http://http://mapas.valencia.es/lanzadera/puntoInteres/fallasvalencia?radio=40000&lang=es&lat=39465212&lon=-374521&filtros=" + i,
    dataType: "json",
    async: false,
    beforeSend: function (xhr) {
                xhr.setRequestHeader ("Authorization", "Basic " + btoa(usuario + ":" + contra));
            },
    success: function(data) {
                $.each( data, function( key, val ) {
                    var object = {
                        titulo: val.titulo,
                        categoria: i,
                        valoracion: 0,
                        votos: 0,
                    };

                    $.ajax({
                        type: "POST",
                        url: "php/saveObra.php",
                        data: object,
                    });
}
```

Se trata de un bucle dónde una variable, en este caso “i”, va incrementando su valor en uno cada vez que acaba toda su lógica interna. Esto se ha hecho para poder almacenar en base de datos todos los monumentos y solo con los datos que más nos interesaban. Para poder obtener los datos, he hecho uso de la URL mencionada anteriormente dónde el campo “N” ahora tiene el valor “i”. Por lo tanto, cada vez que inicie el bucle, “filtros” tendrá un valor distinto. En cuanto a la petición, se ha tenido que usar de *Jquery* un método *ajax* [18]. El método ajax nos permite comunicarnos con un servicio sin tener que recargar la página. En la imagen del código se pueden ver que el método ajax tiene varios atributos distintos para poder especificar el tipo de petición a declarar:

- Type: Aquí se indica el tipo de petición a realizar. En nuestro caso puede ser *POST* o *GET*.
- url: En este campo se le indica la dirección de la petición.
- dataType: Indica el tipo de datos que se espera de la respuesta.
- Async: Aquí se señala si queremos que la petición sea síncrona o asíncrona.
- beforeSend: Si la solicitud requiere de algún tipo de información extra, como en este caso de autenticación, podemos incorporársela en este campo.
- Success: En este último campo se ejecuta un código si la petición es satisfactoria.

HTTP es un protocolo [15] de transferencia que se usa para transferir información, el cual tiene ciertas cabeceras. Una de ellas es “*authorization*”, dónde se le indica que la comunicación que se va a realizar tiene algún tipo de credenciales. En nuestro caso se trata de unas credenciales básicas, dónde tan solo hace falta el usuario y la contraseña cuyas variables tiene por valores ese usuario y contraseña para poder obtener el servicio.

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

Con síncrona o asíncrona hacemos referencia a la espera o no de dicho método. Se suele poner "false" cuándo se necesita mostrar o volcar unos datos en la vista, es decir, se espera a que acabe el método y se muestra. En este caso, los almacenamos en base de datos, con lo cual no haría falta. Sin embargo para prevenir cualquier tipo de fallo, hemos optado por incorporarlo.

Por otra parte tenemos el parámetro "*success*". La función que hace referencia tiene una variable llamada "*data*". Esta variable contiene todos los objetos que nos devuelve dicha petición a la url del ayuntamiento. Esta petición nos devuelve un *array* o lista de *json's*. Cada uno de ellos tiene esta estructura:


```
titulo : Convent de Jerusalem-Matemàtic Marzal (12)
texto : Sección ESPECIAL\nLema: Bollywood\nArtista: Pere Baenas
        García\nAyuntamiento de Valencia
imagen : http://mapas.valencia.es/WebsMunicipales/layar/img/falla_1.png
lat  : 39466451
lon  : -379525
distancia : 452
tipo : 1
▼ acciones [2]
  ▼ 0  {3}
        nombre : Más información
        tipo : text/html
        uri  : http://www.fallas.com
  ▼ 1  {3}
        nombre : Boceto
        tipo : text/html
        uri  : http://mapas.valencia.es/WebsMunicipales/MSController?
               serv=fallas&lang=es&func=getBocetoFallas&id=12
```


La mayoría de la información que el ayuntamiento nos proporciona, no la hemos usado. En nuestro caso a la hora de guardarlo en la base de datos, he usado solo el titulo, ya que al acceder a las fallas de categoría 1, todos van a ser de la misma categoría. Así que con el uso de la variable "i" era suficiente.

A continuación, hablaremos de la función "each" de jquery. Esta función recorre todos los elementos de la lista de "data", cuyo resultado nos devuelve la posición del objeto en la lista "*key*" y el objeto "*value*". Al ser un elemento de tipo *JSON* dónde podemos acceder a sus valores tal y como se muestra en el código "*value.titulo*". Con estos valores, creamos el objeto para poder almacenarlo en base de datos con atributos que nos interesan, como

viene a ser en nuestro caso “valoración” y “votos” para poder realizar el sistema de votos de la aplicación. Por último, nos encontramos con un *POST* que llama a un archivo “sql” para almacenar cada elemento que recorre el método ”each” en base de datos.

8.2.1. saveObra.php

Este archivo se encarga de almacenar en base de datos cada monumento de la lista mencionado anteriormente. En el diseño, se habló de una tabla llamada ***votemonuments***. En esa misma tabla, es dónde se almacenarán estos objetos.

```php
<?php
$server = "localhost";
$username = "root";
$password = "";
$database = "fallas";

$db = new PDO('mysql:host='.$server.';dbname='.$database.';charset=utf8', $username, $password);
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

$titulo=$_POST['titulo'];
$categoria=$_POST['categoria'];
$valoracion=$_POST['valoracion'];
$votos=$_POST['votos'];

$stmt = $db->prepare("INSERT INTO `votemonuments`(`titulo`, `categoria`, `valoracion`, `votos`) VALUES (:titulo, :categoria, :valoracion, :votos)");
$stmt->bindParam(':titulo', $titulo);
$stmt->bindParam(':categoria', $categoria);
$stmt->bindParam(':valoracion', $valoracion);
$stmt->bindParam(':votos', $votos);
echo $stmt->execute();
?>
```

Con “$_POST” obtenemos los valores del objeto y las almacenamos en una variable para poder asignarlas a cada parámetro de la petición “sql”. En cuanto a la petición “sql”, se trata en este caso de un “*INSERT*”. Esta consulta, tal y como indica su nombre, inserta un objeto en una tabla. En este caso, insertamos el monumento en la tabla ***votemonuments.***

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

8.3. Almacenamiento de coordenadas

Para almacenar las coordenadas del usuario, hemos usado una función que nos proporciona google maps: *watchPosition* [19].

```
function shareLocation() {
    id = navigator.geolocation.watchPosition(paintShareLocation);
}
```

Esa función se encarga de devolver la latitud y la longitud cada vez que se realiza un cambio de posición. Por lo tanto, a la hora del seguimiento es perfecto. *watchPosition* solo se activa cuándo la función *shareLocation* es llamada desde el *HTML*. Quien se encarga de esta llamada es el botón de "compartir" que se encuentra en la vista, que al ser pulsado, llama a esta función y pone en marcha el funcionamiento.

```
function paintShareLocation(position) {
    var loginId = window.location.href;
    var userId = loginId.indexOf("=") + 1;
    userId = loginId.substring(userId, userId+1)
    if (showMyPosition) {
        showMyPosition.setMap(null);
    }
    var lat = position.coords.latitude;
    var lng = position.coords.longitude;
     $.ajax({
        type: "POST",
        url: "php/coordenates.php",
        data: { id: userId, lat: lat, lng: lng , status:'ON'}
      });

    var myPosition = {
        center: {lat: lat, lng: lng},
        population: 1000
    };
    var image = {
        url: 'current-position.png',
        size: new google.maps.Size(21, 21),
        origin: new google.maps.Point(0, 0),
        anchor: new google.maps.Point(0, 0)
    };
    showMyPosition = new google.maps.Marker({
        position: myPosition.center,
        map: map,
        icon: image,
    });
}
```

PaintShareLocation es una función que se encarga de pintar en el mapa la posición que *watchPosition* le pasa.

En la imagen se puede apreciar la llamada a un archivo “php”, el cual se encargará de almacenar esas posiciones en base de datos en la tabla ***sharepositions***.

```php
<?php
$server = "localhost";
$username = "root";
$password = "";
$database = "fallas";

$db = new PDO('mysql:host='.$server.';dbname='.$database.';charset=utf8', $username, $password);
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

$id = $_POST["id"];
$lat = $_POST["lat"];
$lng = $_POST["lng"];
$status = $_POST["status"];
$stmt = $db->prepare("UPDATE `sharepositions` SET `lat`='$lat',`lng`='$lng',`status`='$status' WHERE id='$id'");
$stmt->execute();
$obra = $stmt->fetch(PDO::FETCH_ASSOC);

?>
```

En este archivo “sql” nos encargamos de actualizar los datos cada vez que se cambian las coordenadas. Para ello hemos utilizado “*UPDATE*” la cual busca al usuario que ha iniciado la sesión y actualiza su posición en la tabla ***sharepositions***.

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

8.4. Almacenamiento de votaciones

El almacenamiento de votaciones es similar al almacenamiento de monumentos, realiza un *GET* a la *URL* del ayuntamiento pero con la diferencia de que "N" no recorrerá un bucle, sino que desde el *HTML* se llamará a una función y esta función le pasará el entero. Esto es porque en esa misma función es dónde se colocan los marcadores [17] de *google* en el mapa y en cada marcador del mapa es dónde se coloca la funcionalidad de la votación.

```
function addMarkerWithTimeout(position, timeout, i) {
    setTimeout(function() {

        markers.push(new google.maps.Marker({
        position: position,
        map: map,
        animation: google.maps.Animation.DROP
    }));
        var selectId = 'clas'+key;
        contents[i] = '<div><strong>' + val.titulo + '</strong><br>' +
            'Distancia: ' + val.distancia + " m " + '<br></div>'+
            '<select id="clas'+key+'">'+
            '<option value="1"></option>'+
            '<option value="2"></option>'+
            '<option value="3"></option>'+
            '<option value="4"></option>'+
            '<option value="5"></option>'+
            '</select>'
        infowindows[i] = new google.maps.InfoWindow({
            content : contents[i],
            maxWidth: 200
        });

        google.maps.event.addListener(markers[i], 'click', function() {
            infowindows[key].open(map, markers[key]);
            map.panTo(markers[key].getPosition());
            $('#clas'+key).barrating({
                theme: 'fontawesome-stars',
                onSelect: function(value) {
                    updateVote(value , val.titulo)
                },
```

En esta función, nos encargamos de incorporar los marcadores de los distintos monumentos de las fallas. *Google maps* nos proporciona diversas funciones como son: "*google.maps.Marker*" el cual nos permite marcar en el mapa una posición, "*google.maps.infoWindow*" que se encarga de abrir el contenido de la información de cada falla. Ese contenido se almacena en "*contents*" que se trata de una lista de los distintos contenidos de cada monumento.

Al final del código, vemos una llamada en "*jQuery*" a los distintos identificadores de cada "select" que incorporamos en los marcadores de los monumentos. Esta función viene del *plugin* de "*jQuery Bar Rating*" y nos permite

transformar ese "select" en una barra de estrellas la cual usaremos para las votaciones. Tiene varios atributos, pero en nuestro caso hemos usado: "theme" que sirve para seleccionar el tipo de estrella, ya que hay varios tipos. "onSelect" llama a una función al seleccionar alguna estrella. En nuestro caso llama a la función "*updateVote*" y se encarga del almacenamiento del voto.

"*updateVote*" llama a un archivo *PHP* para realizar el voto y se realiza mediante *POST*.

```
function updateVote(value, title) {
    var object = {
        titulo: title,
        valoracion: value,
    };

    $.ajax({
        type: "POST",
        url: "php/updateVote.php",
        data: object,
        success: function() {
            $("#voteMessage").delay(500).fadeIn().delay(1000).fadeOut()
        }
    });
}
```

Esta función recibe dos parámetros: la valoración y el nombre del monumento. Una vez que el POST se ha realizado correctamente, en la vista se muestra un mensaje y se vuelve a esconder, para informar a usuario que la votación se ha realizado correctamente.

```
<?php
$server = "localhost";
$username = "root";
$password = "";
$database = "fallas";

$db = new PDO('mysql:host='.$server.';dbname='.$database.';charset=utf8', $username, $password);
$db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

$titulo = $_POST["titulo"];
$valoracion = $_POST["valoracion"];

$stmt = $db->prepare("UPDATE `votemonuments` SET `valoracion`= valoracion + $valoracion,`votos`= votos + 1 WHERE `titulo`= '$titulo'");
$stmt->execute();
$obra = $stmt->fetch(PDO::FETCH_ASSOC);

?>
```

En cuanto al archivo "*updateVote.php*" se trata de una consulta de tipo "*UPDATE*" al igual que al compartir las coordenadas de los estandartes. Sin embargo no buscamos por identificador, sino por nombre de monumento ya que el ayuntamiento no nos proporcionaba un identificador de cada monumento.

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

8.5. Obtener votaciones

Las votaciones las mostramos en la aplicación de dos formas, una de ellas es en el momento en dónde se muestran los marcadores en el mapa, y la otra es cuándo se desean ver las clasificaciones. En ambas vistas, los datos se obtienen de la misma manera, mediante una petición *GET* con *ajax* pero cada uno a un archivo distinto. En el primer caso, en el momento de mostrar el mapa, la media de las votaciones se obtiene cada vez que se inserta un marcador en la vista.

```
var middleVotes;
$.ajax({
    type: "GET",
    url: "php/getVotes.php",
    data: {titulo: val.titulo},
    success: function(data){
        data = JSON.parse(data);
        middleVotes = data.valoracion/data.votos;
    }
});

google.maps.event.addListener(markers[i], 'click', function() {
    infowindows[key].open(map, markers[key]);
    map.panTo(markers[key].getPosition());
    $('#clas'+key).barrating({
        theme: 'fontawesome-stars',
        onSelect: function(value) {
            updateVote(value , val.titulo)
        },
        initialRating: middleVotes,
```

En la petición, le pasamos como dato el titulo de dicho monumento para que lo busque y nos devuelva el número de votos y la valoración total que tiene para poder sacar una media. Este número, se usará para incorporarlo en la barra de estrellas como "valor inicial", es decir, mostrará la media valorada de cada uno de los monumentos.

Por otra parte, en el segundo caso, cuándo se desean ver las clasificaciones de los monumentos mostrados en una tabla, tenemos una vista preparada para ello. Esta vista obtiene los valores de base de datos y los incorpora en la vista.

```php
if(isset($_GET["category"])){
    $category = $_GET["category"];
    $stmt = $db->prepare("SELECT * FROM votemonuments WHERE categoria='$category' ORDER BY `votemonuments`.`valoracion` DESC");
    $stmt->execute();
        $obras=array();
        while($obra = $stmt->fetch(PDO::FETCH_ASSOC)){
            $obras[]=$obra;
        }
        echo json_encode($obras);
}

else if(isset($_GET["all"])) {
    $stmt = $db->prepare("SELECT * FROM votemonuments ORDER BY `votemonuments`.`valoracion` DESC");
    $stmt->execute();
        $obras=array();
        while($obra = $stmt->fetch(PDO::FETCH_ASSOC)){
            $obras[]=$obra;
        }
        echo json_encode($obras);
}
```

En esta petición es condicionada, ya que en la vista es posible seleccionar el tipo de categoría que se desea ver. Se le pueden pasar dos tipos de argumentos: o bien la categoría para obtener solo las fallas de esa categoría o con “all” obtendremos todas las fallas con sus valoraciones medias y, al igual que en el mapa, mostrará sus medias en la tabla de la vista.

8.6. Obtener usuarios estandartes

Para obtener este resultado se ha tenido que realizar una petición *GET* con *ajax* al archivo *search.php*. A este archivo se le ha tenido que pasar el nombre de usuario que el cliente desea obtener. Por ello, la aplicación ofrece un listado de los distintos usuarios disponibles en la base de datos que se desean buscar.

```javascript
$.ajax({
    type: "GET",
    url: "php/search.php",
    data: {all : 'all'},
    success: function(data){
        data = JSON.parse(data);
        $.each(data, function(key, val) {
            var name = "<li>"+val.username+"</li>";
            $('#listview').append(name).listview('refresh'); //refresh necesario para aplicar estilo jquerymobile
        })
    }
});
```

La petición, al devolver todos los usuarios disponibles en la base de datos, con *javascript*, insertamos los nombres de cada uno consiguiendo formar la lista.

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

8.7. Obtener coordenadas estandartes

Por otra parte para obtener las coordenadas, obtenemos el usuario que el cliente ha solicitado, y lo buscamos en base de datos en el mismo archivo: *search.php*

```
if(isset($_GET["username"])){
    $usu = $_GET["username"];
    $stmt = $db->prepare("SELECT * FROM sharepositions WHERE username='$usu'");
    $stmt->execute();
    $user = $stmt->fetch(PDO::FETCH_ASSOC);
    echo json_encode($user);
}

else if(isset($_GET["id"])) {
    $id = $_GET["id"];
    $stmt = $db->prepare("SELECT * FROM sharepositions WHERE id='$id'");
    $stmt->execute();
    $user = $stmt->fetch(PDO::FETCH_ASSOC);
    echo json_encode($user);
}
else if(isset($_GET["all"])) {
    $stmt = $db->prepare("SELECT username FROM sharepositions");
    $stmt->execute();
        $users=array();
        while($user = $stmt->fetch(PDO::FETCH_ASSOC)){
            $users[]=$user;
        }
        echo json_encode($users);
```

La primera condición es la que se encarga de esta funcionalidad. Si la petición que llama al archivo *search.php* le pasa un dato con una propiedad "*username*", ésta, buscará al usuario. La segunda condición la usamos para el inicio de sesión de la aplicación de los estandartes. Por último la tercera, la usamos para obtener todos los usuarios y mostrarlos en la lista como mencionamos anteriormente.

8.8. Funcionamiento del calendario

Para el funcionamiento del calendario, se mencionó que se ha tenido que hacer uso de archivos externos para que funcionara en *jQuery mobile*. En concreto son dos archivos en *javascript* y dos archivos *CSS*. Los archivos en *javascript* son para poder darle la funcionalidad como un calendario normal de *jQuery* pero integrado en *jQuery mobile* y los *CSS* son para aplicarle unos estilos de diseño modernos. Por otra parte, para la información de los distintos eventos que ocurren durante las fallas, no existe ningún tipo de servicio que se pueda consumir.

Por ello, se ha tenido que recurrir a terceros [20] para poder obtener información de dichos eventos y colocarlos en nuestra aplicación. En cuanto al código, el calendario muestra los eventos de dos maneras distintas. Una es cuándo se carga la página o vista, se muestra automáticamente si existe o no un evento. La otra ocurre cuándo cambiamos la fecha en el mismo calendario.

```
$(document).on("pagecreate","#calendar-page",function(){

  $( "#datepicker" ).datepicker();
  var firstLoadDate = $( "#datepicker" ).datepicker("getDate");
  var stringDate = getAsString(firstLoadDate);
  document.getElementById("descripcion").innerHTML = searchEvent(stringDate);

  $( "#datepicker" ).change(function() {
    var date = $( "#datepicker" ).datepicker("getDate");
    var stringDate = getAsString(date);
    document.getElementById("descripcion").innerHTML = searchEvent(stringDate);
  });
})
```

Cada vez que cambia el día en el mismo calendario pinchando sobre él o cargando la vista, llama a un método "*getAsString*" para obtener la fecha como una cadena de caracteres.

```
function getAsString(date) {
  date = date;
  let month = date.getMonth() + 1;
  month = (month < 10) ? `0${month}` : month.toString();
  let day = date.getDate();
  day = (day < 10) ? `0${day}` : day.toString();

  return `${date.getFullYear().toString()}${month}${day}`;
}
```

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

De esta forma conseguimos tener una fecha de la siguiente manera: “yyyy/mm/dd” como por ejemplo “20170215”. Luego se trataría de buscar con esa fecha en forma de cadena de caracteres, buscar el evento. El resultado de este evento es un fragmento en *HTML* que se insertará en la vista junto con un pequeño mapa que indicará las posiciones de dichos eventos.

```
case "20170226":
  event = '<div> '+
  '<p>07:30h: Despertà Infantil</p>'+
  '<p>07:40h: Despertà en colaboración con la Falla Mossen Sorell - Corona.Al acabar la despertà, final con un terremoto espectacular.</p>'+
  '<p>12:30 h: Entrada de Bandas de Música</p>'+
  '<p>14:00h: Mascletà en la plaza del Ayuntamiento</p>'+
  '<p>20:00 horas: Crida en las Torres de los Serranos. Al acabar, efectos de luz y color con espectáculo pirotécnico.</p>'+
  '</div>';
  listEvents.push(events.MASCLETA);
  listEvents.push(events.CRIDA);
  showMap(listEvents);
  break;
```

La función “*showMap*” muestra en el mapa los eventos que coinciden con esa fecha. Estos, son puestos por nosotros en función de la información que nos proporcionaba este tercero. En caso de que exista dicho evento, se crea un marcador con una posición específica que nosotros mismos hemos buscado y con el nombre del evento.

```
function showMap(listEvents) {
    document.getElementById("map-evento").style.display = "block";
    listCoordsWithEvents = [];
    var center = new google.maps.LatLng(39.469732, -0.376398);
        if (listEvents.indexOf(events.MASCLETA) > -1) {
            var coords = new google.maps.LatLng(39.469732, -0.376398);
            var event = {
                coords: coords,
                event: 'mascleta',
            }
            listCoordsWithEvents.push(event);
        }
```

9. CONCLUSIONES

Cerrando el texto que compone el presente proyecto, concluimos que finalmente se han podido alcanzar de manera satisfactoria los objetivos marcados al inicio del mismo.

Por parte de la aplicación para los usuarios portadores de estandartes, conseguimos que la aplicación le permita al usuario iniciar sesión para después compartir o parar, la compartición de su ubicación. También, hemos aplicado nuestros conocimientos aprendidos en *Base de Datos* para la inserción de las coordenadas y del estado del usuario en nuestra base de datos. Por último, aprendimos *PHP* que nos ha permitido realizar las consultas a la base de datos.

Por otro lado, en la aplicación de los clientes hemos conseguido también los distintos objetivos.

En primer lugar, se ha podido incorporar un mapa que mediante la geolocalización nos proporciona las diferentes ubicaciones de los monumentos durante dichas fiestas. En cada una de las posiciones mostradas en el mapa, indica información sobre la clasificación popular del monumento, nombre y la distancia a la que el cliente se encuentra del mismo. Además, también proporciona la posibilidad de poder buscar los diferentes estandartes mientras estén compartiendo su ubicación.

Con lo referente al segundo objetivo, también hemos sido capaces de llevarlo a cabo en su totalidad. El calendario muestra los diferentes eventos en función de la fecha que se busque el usuario o en el día en el que se encuentre. Estos eventos constan de una descripción de los diferentes horarios en los que se celebran y de un pequeño mapa dónde se muestran la ubicación en dónde se celebra.

En cuanto a proporcionar información cultural sobre las fiestas, también se ha conseguido. Se ha tenido que hacer uso de terceros y se ha incorporado la información en el proyecto de manera “manual”.

En cuarto lugar, conseguimos garantizar al usuario orientación gracias a que en el mapa que utilizamos, se muestra un punto de color azul el cual le indica en qué lugar se encuentra el cliente o turista que esté usando la aplicación.

Por último, hemos logrado aplicar los conocimientos aprendidos en *Desarrollo web*. Hemos puesto en práctica los lenguajes de programación *HTML*, *javascript* y *CSS* aprendidos en dicha asignatura y además, hemos aprendido más respecto a estos lenguajes de manera autodidacta como podrían ser las funciones *ajax* con las cuales hemos realizado las distintas peticiones. Además de mejorar nuestro conocimiento con dichos lenguajes, también hemos aprendido a utilizar el servicio de google maps y sus distintas funciones que nos proporciona, tal como: *google.maps.Marker* para insertar los marcadores o *google.maps.infoWindow* para incorporar información adicional a dichos marcadores.

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

De lo dicho hasta el momento puede concluirse que la realización del presente proyecto ha sido una experiencia enriquecedora en mucho de los sentidos, puesto que hemos aprendido algunos aspectos informáticos y otros más técnicos, haciendo hincapié en un mundo bastante interesante en la actualidad y con bastantes utilidades en el futuro, tanto gracias al servicio de *google* como a las tecnologías y lenguajes aprendidos durante su realización.

BIBLIOGRAFÍA

[1] ¿Qué es una aplicación móvil? Disponible en: https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_m%C3%B3vil

[2] ¿Qué es Aptana Studio 3? Disponible en: http://www.aptana.com/products/studio3.html

[3] ¿Qué es XAMPP? Disponible en: https://www.apachefriends.org/es/index.html

[4] ¿Qué es PHP? Disponible en : http://php.net/manual/es/intro-whatis.php

[5] ¿Qué es Javascript? Disponible en: https://developer.mozilla.org/es/docs/Web/JavaScript

[6] ¿Qué es jQuery Mobile? Disponible en: https://jquerymobile.com/

[7] Introduction to InnoDB MySQL Server. Disponible en: https://dev.mysql.com/doc/refman/5.5/en/innodb-introduction.html.

[8] ¿Qué es Visual Studio code? Disponible en: https://code.visualstudio.com/?wt.mc_id=DX_841432

[9] ¿Qué es justinmind? Disponible en: https://www.justinmind.com/

[10] Obtener api key o autenticación. Disponible en: https://developers.google.com/maps/documentation/javascript/get-api-key?hl=ES

[11] API proporcionada por el ayuntamiento de Valencia. Disponible en: http://gobiernoabierto.valencia.es/va/info-api/

[12] Diseño Centrado en el Usuario. Disponible en: http://www.nosolousabilidad.com/manual/3.htm

[13] Información sobre google form. Disponible en: https://www.google.es/intl/es/forms/about/

[14] Solicitar credenciales para el uso de la API del ayuntamiento. Disponible en: http://gobiernoabierto.valencia.es/va/info-api/

[15] Votación con estrellas: http://antenna.io/demo/jquery-bar-rating/examples/

[16] Datepicker widget jQuery Mobile. Disponible en: http://demos.jquerymobile.com/1.4.5/datepicker/

[17] Simple marker. Disponible en: https://developers.google.com/maps/documentation/javascript/examples/marker-simple

[18] Jquery ajax. Disponible en: http://api.jquery.com/jquery.ajax/

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

[19] WatchPosition de google maps. Disponible en: https://developer.mozilla.org/es/docs/Web/API/Geolocation/watchPosition

[20] calendario de fallas. Disponible en: http://www.lasprovincias.es/fallas-valencia/201702/24/actos-fallas-mascleta-vertical-crida-20170224101445.html

[21] Información sobre las fallas. Disponible en: http://www.abc.es/local-comunidad-valenciana/20130318/abci-fallas-valencia-201303142302.html

[21] Login para jQuery Mobile. Disponible en: https://afinandocodigo.wordpress.com/2013/05/05/simple-login-con-jquery-mobile-mysql-y-php/

[22] Cargar páginas de jquery mobile. Disponible en: https://api.jquerymobile.com/page/#event-create

[23] Eliminar o parar la monitorización de watchPosition. Disponible en: https://developer.mozilla.org/es/docs/Web/API/Geolocation/clearWatch

[24] Imagen para mostrar ubicación actual en el mapa. Disponible en: https://www.kh.hu/kh-theme/images/current-position.png

[25] Colocar un marcador personalizado en google maps. Disponible en: https://developers.google.com/maps/documentation/javascript/examples/icon-complex?hl=es-419

[26] Encriptar datos en sql. Disponible en: https://dev.mysql.com/doc/refman/5.5/en/encryption-functions.html#function_sha2

[27] Marcar aéreas circulares en el mapa para las zonas populares. Disponible en: https://developers.google.com/maps/documentation/javascript/examples/circle-simple?hl=es-419

[28] Como colocar varios marcadores en google maps. Disponible en: https://stackoverflow.com/questions/24884197/declaring-google-map-markers-in-a-loop

ANEXO A: MANUAL DE USUARIO

El objetivo de este anexo es explicar el funcionamiento de las aplicaciones llevadas a cabo en este proyecto durante las fallas de valencia. La aplicación permite la geolocalización e información de monumentos y eventos durante dichas fiestas. Las aplicaciones están asignadas para usuarios específicos. Una de ellas es para los clientes o turistas que la utilizarán para informarse o encontrar las ubicaciones de los monumentos. La otra es para usuarios específicos que son los encargados de llevar los estandartes. Estos Se encargarán de compartir su ubicación para que la otra aplicación pueda mostrarla, es decir, que los clientes o turistas puedan seguir el estandarte a través de la aplicación.

APLICACIÓN ESTANDARTE

En el inicio de sesión tenemos un formulario dónde el usuario rellena sus credenciales que nosotros mismos le hemos ofrecido. Si las credenciales no son correctas, saldrá un aviso indicando que los datos no son correctos, creando así una seguridad a la hora de acceder a la plataforma.

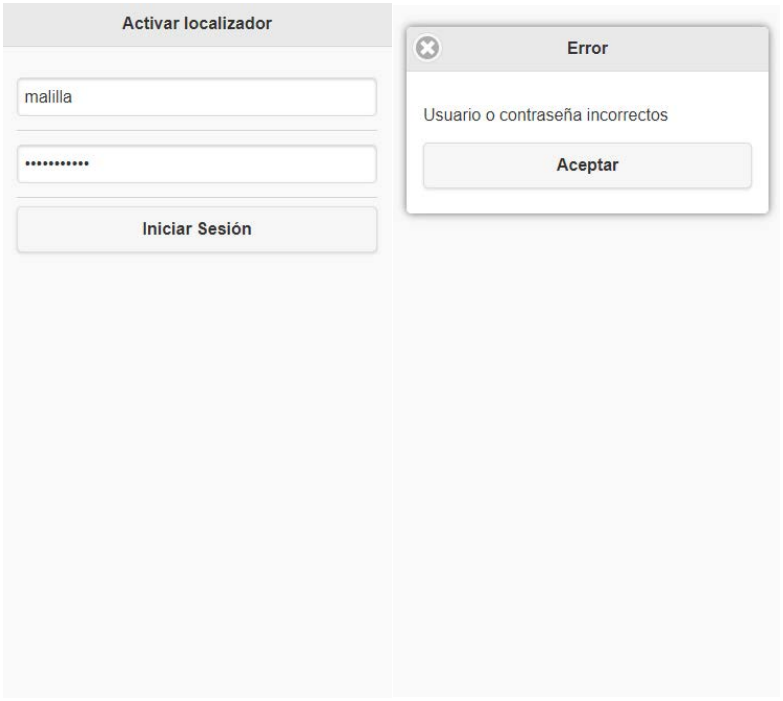


Ilustración 21: Página *Login* estandarte

En caso contrario, accederá a la vista del mapa dónde podrá compartir su ubicación. En esta vista tendrá tres botones: uno para volver a la vista de inicio de sesión y los otros dos para compartir o no su ubicación. En el momento en el que se pinche o seleccione “compartir”, la aplicación mostrará su ubicación actual e irá cambiando cada vez que su ubicación varíe.

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

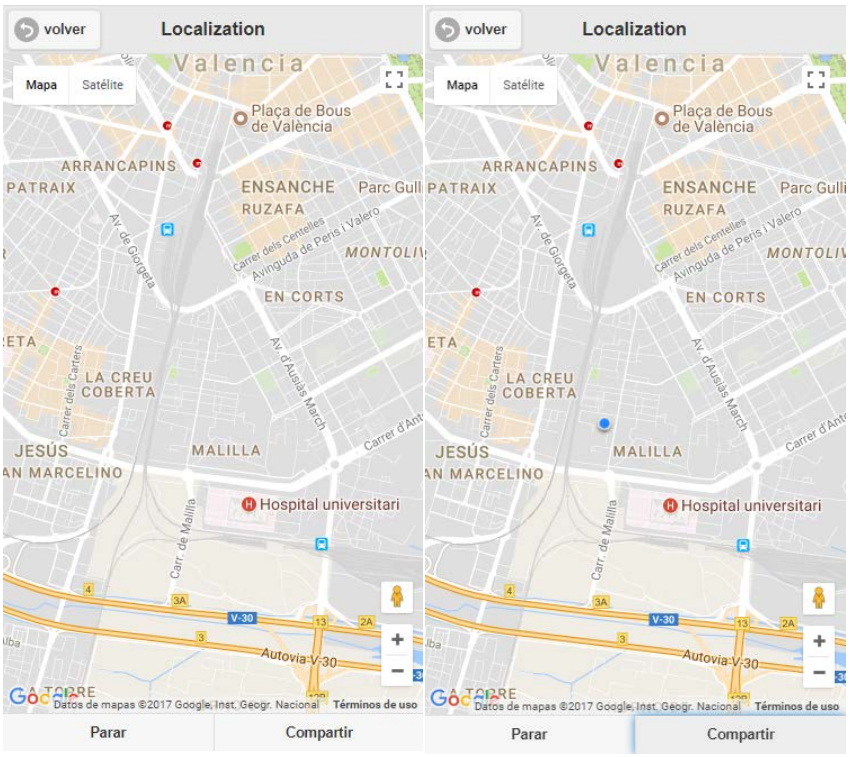


Ilustración 22: Página compartir ubicación

APLICACIÓN CLIENTE O TURISTA

En cuanto a la aplicación del cliente, tenemos una vista principal en el cual se muestran diferentes opciones disponibles para que el usuario pueda tener de primera mano todas las funcionalidades ofrecidas.

Ilustración 23: Página índice

En primer lugar, al seleccionar los monumentos, ésta redirigirá a la vista con *google maps*, mostrando de primera mano las posiciones de los monumentos de categoría especial. En caso de querer más información de algún monumento, solo será necesario seleccionar el marcador, abriendo la información adicional.

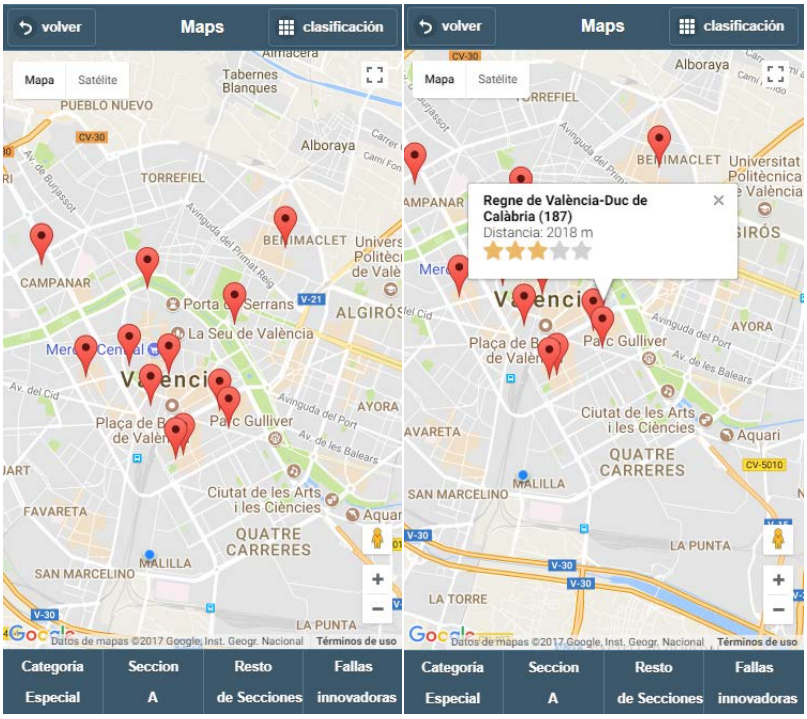


Ilustración 24: Página monumentos

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

En caso de querer realizar la votación de ese monumento, será tan sencillo como seleccionar la valoración. Automáticamente, después de votar, saldrá un mensaje indicando que la votación se ha realizado correctamente para informar al usuario que no ha ocurrido ningún error a la hora de realizar el voto.

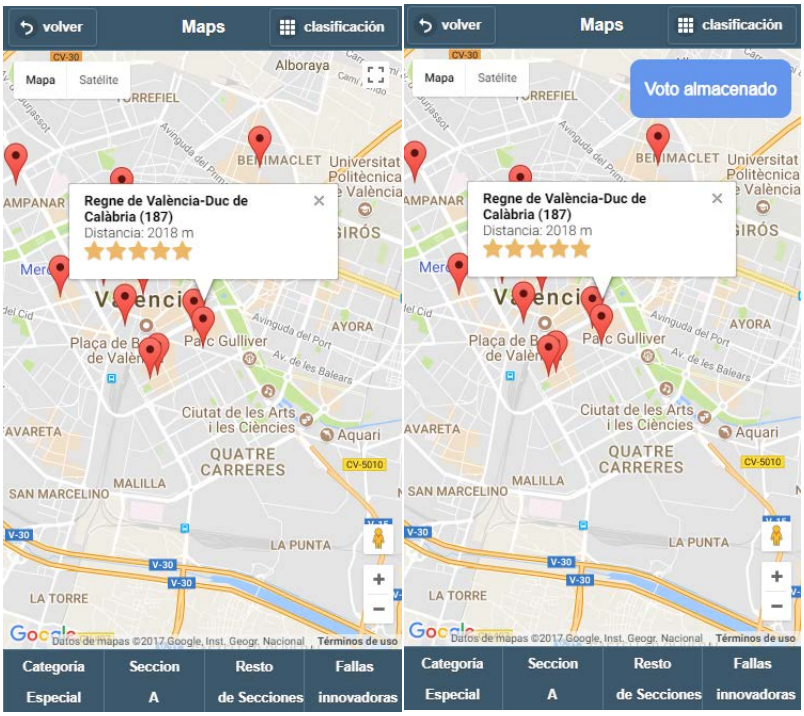


Ilustración 25: Página votación monumentos

A la hora de entrar en las clasificaciones, se mostrará una lista con todos los monumentos disponible en la base de datos. Si el usuario desea buscar por tipo de categoría, deberá seleccionar el botón “categoría”. En esta ventana será posible seleccionar una de las cinco posibilidades dónde mostrará la categoría seleccionada o bien todos los monumentos de nuevo.

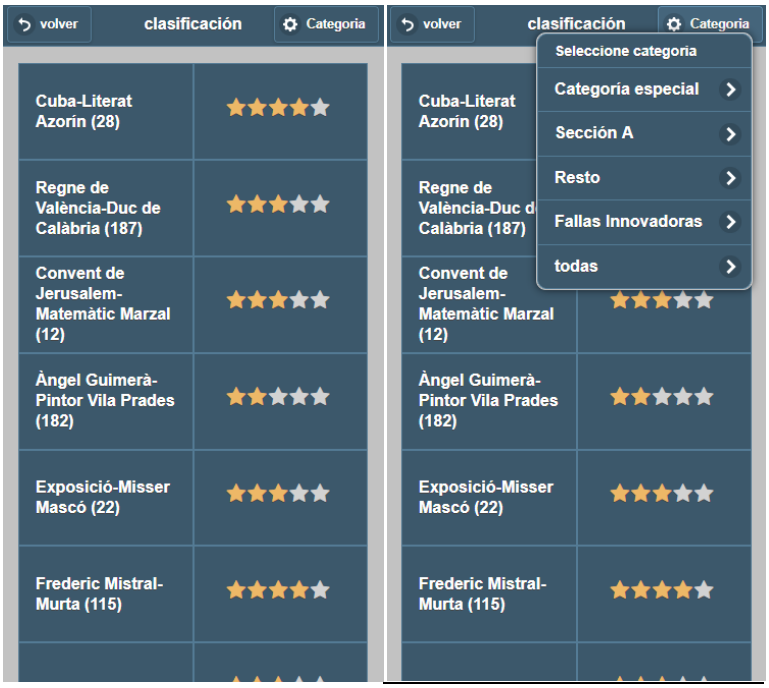


Ilustración 26: Página clasificación fallas

A continuación, en segundo lugar, si en el índice se selecciona el apartado de información, enviará al usuario a una vista dónde se muestran diversas cuestiones sobre las fallas de Valencia. Si el usuario pincha o selecciona una de las cuestiones, se despliega un contenido con la información detallada de dicha pregunta.

Ilustración 27: Página información

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

En tercer lugar, nos encontramos con la opción de localizar estandartes. Al seleccionar dicha opción, la aplicación cambiará de vista a un formulario el cual muestra la lista de los estandartes disponibles en base de datos para poder buscar. Mientras se escribe el estandarte interesado, la lista se va filtrando en función del contenido.

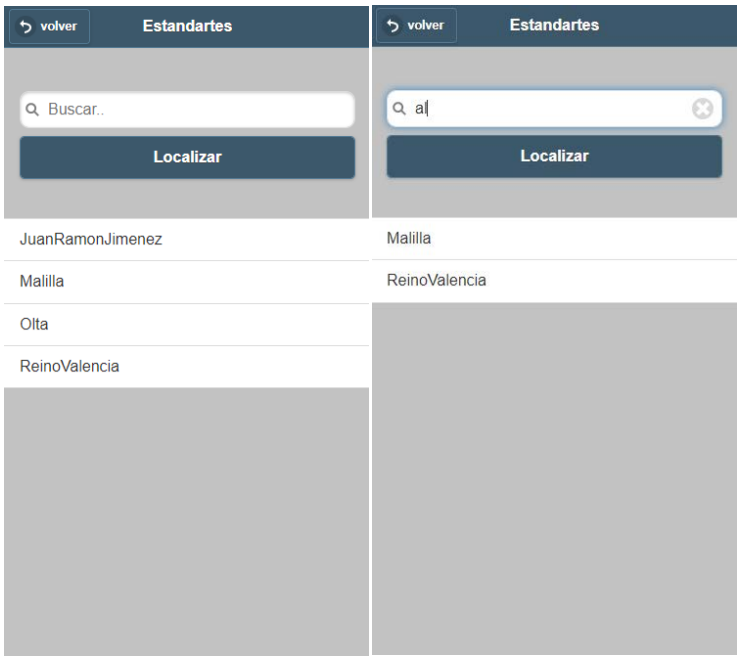


Ilustración 28: Página buscar estandarte

Una vez localizada el estandarte que se desea seguir, solo se tendrá que pulsar el botón y nos redirigirá al mapa con la ubicación de dicho estandarte.

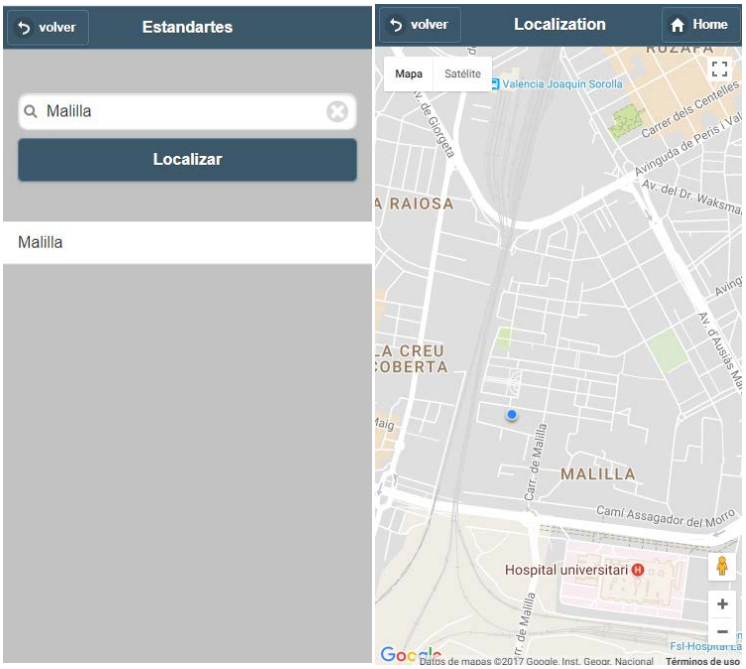


Ilustración 29: Página ubicación estandarte

En el caso de que el estandarte dejara de compartir su ubicación mientras el cliente o turista está observando su posición en el mapa, la aplicación retirará al usuario del mapa y mostrará la vista del formulario. En cuanto al botón de "Home" tiene como objetivo volver al índice de la aplicación.

En cuarto lugar, tenemos las zonas populares que nos mostrará un mapa en dónde se señalarán las zonas populares durante las fiestas mediante círculos simulando un área.

Ilustración 30: Página zonas populares

En último lugar, el calendario indicará los distintos eventos durante las fiestas. Dicho calendario en caso de encontrarse en una fecha que no tenga ningún tipo de evento, lo indicará debajo de él.

[15]: En informática, un protocolo es un conjunto de reglas usadas para la comunicación entre distintas máquinas.

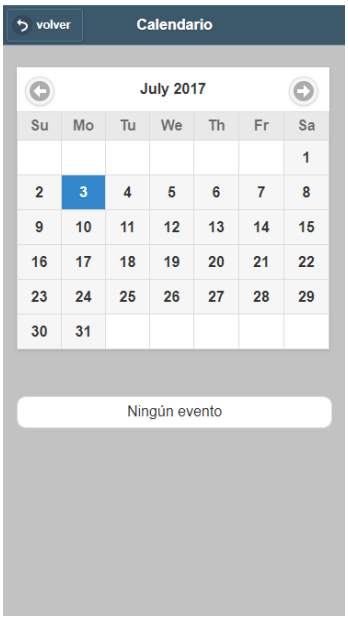


Ilustración 31: Página calendario

El usuario también puede buscar una fecha en concreta para consultar si hay o no un evento. En caso de que exista uno o varios eventos, Se mostrará información con respecto de esos eventos y un mapa de la ubicación de estos.

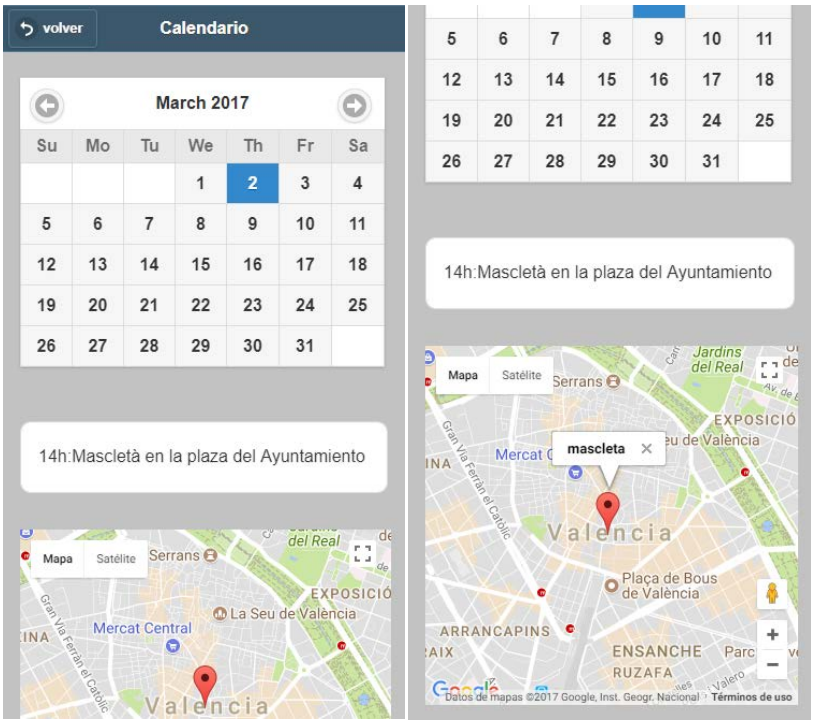


Ilustración 32: Página calendario con evento