

Aplicación web para la gestión de grandes eventos

Astyr Doncel-Chao

Resumen– El auge del comercio electrónico crece cada año, y la venta de entradas no es una excepción. Cada vez más, los hábitos de compra de la gente que quiere comprar entradas para cine, teatro, conciertos, festivales, entre otros eventos, decide adquirirlas mediante plataformas de venta de entradas *online*. Esta tendencia ha motivado la realización de este proyecto creando una plataforma *web* para la venta de entradas y gestión de grandes eventos (tales como festivales de música, conciertos, congresos, etc.) dando la posibilidad a los clientes de tener un espacio personal donde poder estar informados de sus eventos favoritos y gestionar sus compras. Por otro lado, los responsables de un evento pueden gestionar todo lo relacionado con la administración de las entradas, noticias y avisos con la posibilidad de ver, mediante gráficos estadísticos, datos relevantes sobre los eventos que administrar.

Palabras clave– Ticketing, venta de entradas, ticket, gestión de eventos, Codeigniter, web, plataforma online, MVC, PHP, Bootstrap, MySQL, responsive, Highcharts, Feed RSS, Código QR

Abstract– The rise of electronic commerce grows every year, and ticket sales are no exception. Increasingly, the buying habits of people who want to buy tickets for movies, theater, concerts, festivals, among other events, is acquired through online ticket sales platforms. This trend has motivated the realization of this project by creating a web platform for the sale of tickets and management of major events (such as music festivals, concerts, congresses, etc.) giving customers the possibility of having a personal space where they can be informed of their favorite events and manage your purchases. On the other hand, those managers of an event can manage everything related to the administration of the entries, news and announcements with the possibility of viewing, through statistical graphs, relevant data about the events they manage.

Keywords– Ticketing, ticket sales, event management, Codeigniter, web, online platform, MVC, PHP, Bootstrap, MySQL, responsive, Highcharts, RSS feed, QR code



1 INTRODUCCIÓN

Las ventas de entradas mediante plataformas de *ticketing* son cada vez más populares entre las personas que deciden comprar entradas para conciertos, festivales, teatro, etc. En España, en el año 2016 se experimentó un crecimiento en eventos musicales, de teatro y artes escénicas de entorno al 40 % [1].

Por otra parte, este crecimiento se vio acentuado por la utilización de las nuevas tecnologías, donde se destaca la utilización del ordenador como primera opción, seguida

muy de cerca por el *smartphone*.

El auge de estas nuevas formas de compra mediante grandes *webs* donde se pueden encontrar infinidad de eventos ha hecho que la venta por estos canales *online* haya aumentado de forma considerable a nivel español en los últimos años [2].

Un aspecto a resaltar de este fenómeno es la exigencia del consumidor cuando quiere informarse sobre un evento en concreto. Los clientes se informan más y mejor que antes cuando se disponen a comprar cualquier tipo de entrada, haciendo que estén mejor informados en todo momento. Redes sociales, noticias en *webs* especializadas, periódicos, etc. son las nuevas formas de comunicación entre eventos y clientes.

Para poder dar una alternativa más al usuario nace la propuesta de este TFG *Aplicación web para la gestión de grandes eventos* donde se quiere dar un nuevo enfoque a las plataformas ya existentes de venta de entradas o *ticketing*,

• E-mail de contacto: asty.doncel@e-campus.uab.cat
 • Mención realizada: Software
 • Trabajo tutorizado por: Marc Talló-Sendra (Dpto. Ciencias de la Computación)
 • Curso 2016/2017

haciendo que sea una experiencia más social, dando información a los clientes de múltiples formas, con un diseño *responsive* que garantice la mejor experiencia para el usuario sin importar el dispositivo desde el que acceda y, por lo que a los organizadores se refiere, poder gestionar las entradas, noticias y avisos de sus eventos de forma sencilla y rápida añadiendo la posibilidad de tener en todo momento gráficos estadísticos sobre el número de entradas vendidas, recaudación, visitas a la *web*, entre otras funcionalidades.

El resto de este artículo se organiza de la siguiente forma: sección 2, describe el contexto del proyecto donde se introduce el framework utilizado para la implementación del proyecto; sección 3, presenta los objetivos del proyecto; sección 4, presenta el estado del arte de la venta de entradas *online*; sección 5, se describe la metodología utilizada para llevar a cabo el proyecto; sección 6, se explica la implementación de la plataforma; sección 7, muestra los resultados obtenidos y, por último, sección 8, donde se presentan las conclusiones.

2 CONTEXTO

En esta sección, con el fin de contextualizar el trabajo desarrollado, se introduce el *framework* de PHP elegido para llevar a cabo la implementación de este proyecto.

Codeigniter es un *framework* para el desarrollo de aplicaciones en php que permite a los desarrolladores programar de forma más rápida gracias a sus múltiples librerías, la utilización del sistema MVC y a la fácil adaptación que requiere para aprender a utilizarlo [3].

Algunas de las características más destacadas son las siguientes:

- Utilizar el modelo MVC que nos permite tener separadas la lógica, los datos y la presentación del programa. Esto conlleva que, si estás trabajando en una de estas partes, las otras no se vean directamente afectadas.
- Programación orientada a objetos. Es ideal para poder reutilizar código durante toda la plataforma y así poder ahorrar tiempo al hacer código más compacto y entendible.
- Fácil instalación. Para poder empezar a programar en Codeigniter solo hace falta descomprimir el *zip* disponible en la página oficial.
- En Internet se puede encontrar muchos manuales (incluido el oficial), vídeo tutoriales y foros en los que poder extraer información para la implementación del trabajo. También se debe destacar que el aprendizaje con este *framework* es bastante rápido y es fácil amoldarse al estilo de codificación.
- PHP presente. Se pueden utilizar las sentencias que vienen por defecto, haciendo que el trabajo se reduzca, pero también se pueden ejecutar sentencias nativas de PHP lo que facilita los primeros pasos en este sistema.

3 OBJETIVOS

Dada las múltiples funcionalidades que se pueden implementar en una plataforma de estas características, se plantearon tres requisitos básicos para desarrollar:

- Unir la gestión de grandes eventos en una misma plataforma *online*.
- Facilitar a los usuarios la compra de entradas, así como la posibilidad de estar siempre informados de las últimas novedades de sus eventos favoritos.
- Gestionar la venta de entradas y los avisos de los eventos de forma fácil e intuitiva para los administradores.

3.1. Objetivos de la plataforma

A partir de los tres requisitos básicos que se perseguían con la implementación de este proyecto se definieron los objetivos de la implementación del mismo. (Tabla 1)

Nº	Objetivo	Prioridad
1	Diseño de la plataforma	Crítico
2	Creación y diseño de la BBDD	Crítico
3	Implementación de la plantilla <i>web</i> para alojar la información del evento	Crítico
4	Implementación del <i>Espacio personal</i>	Crítico
5	Implementación del <i>Espacio de administración</i>	Crítico
6	Test de cada uno de los módulos de la plataforma	Prioritario
7	Test de las funcionalidades de la plataforma por parte de usuarios	Secundario
8	Añadir nuevas funcionalidades	Secundario

TABLA 1: OBJETIVOS DEL PROYECTO

3.2. Requisitos por módulo

Por otro lado, y como se explica más ampliamente en la sección cinco sobre la metodología, se ha subdividido el desarrollo del proyecto en tres bloques: plantilla para eventos, el espacio personal (para usuarios) y espacio de administración. Estos bloques o módulos, tienen unos requisitos asociados para poder validar, al final de la implementación, el correcto desarrollo del proyecto que se presenta.

Bloque 1 - Plantilla para eventos

La creación de una plantilla sobre la cual se van a alojar diferentes eventos es primordial para la plataforma. Esta plantilla será la *web* donde los usuarios podrán ver toda la información sobre un evento en concreto, así como la posibilidad de comprar entradas para el mismo.

Los requisitos asociados a este bloque son los siguientes: creación y diseño de la BBDD única de la plataforma, diseño e implementación de la plantilla para eventos, desarrollo del formulario de registro y el *login*, así como el desarrollo de la vista para adquirir entradas mediante registro previo.

Bloque 2 - Espacio para usuarios.

Los usuarios tienen un espacio en la plataforma donde poder ver las noticias de sus eventos favoritos, avisos

y su historial de entradas compradas. Esto hace que el espacio para usuarios esté localizado tanto en la *web* del evento como en la *web* de ADCTickets, donde el usuario puede acceder a su espacio sin la necesidad de iniciar sesión en la *web* de un evento de la plataforma.

Para la realización de este bloque los requisitos a desarrollar son: diseño del espacio para usuarios (el mismo para la *web* de un evento como para la *web* de ADCTickets), implementación de la compra de entradas, generación de entradas mediante PDF, creación de códigos QR, implementación del historial de compras, noticias y avisos que puede ver el usuario registrado, implementación de un *Feed RSS* de noticias, dar la posibilidad al usuario de poder seleccionar sus eventos favoritos y cambiar sus datos personales cuando desee.

Bloque 3 - Espacio para administradores.

El último bloque, asociado a la gestión de los eventos, basa su funcionalidad en dos partes: administración de entradas, noticias y avisos y, por otra parte, control de estos mediante gráficos.

El diseño de este último bloque es el primero requisito a desarrollar. A su vez, se debe implementar la gestión de usuarios administradores (creación, modificación e inhabilitación) y creación de roles; gestión de entradas, noticias y avisos (creación, modificación e inhabilitación), y creación de los gráficos.

4 ESTADO DEL ARTE

La venta de entradas *online* o *ticketing* ha ido creciendo de manera exponencial en los últimos años en España, provocando que las entradas vendidas por canales *online* conlleven el 80 % de las ventas de *tickets* y entradas para eventos.

Los canales de venta habituales por los usuarios son grandes plataformas de *ticketing* como Ticketmaster, Atrápalo o entradas.com donde el usuario puede comprar, entre una gran variedad de eventos de diferentes temáticas y géneros, entradas de forma sencilla y con la seguridad de estar comprando en un medio oficial y seguro.

Muchos de los eventos que se pueden encontrar en estas plataformas disponen de *webs* propias donde se puede encontrar toda la información relevante que puede necesitar una persona interesada sobre el evento en cuestión. Dentro de estas *webs* es posible comprar entradas directamente a la empresa organizadora o mediante un intermediario que le proporciona el motor de compra.

Plataformas como Ticketmaster están presentes en múltiples *webs* de eventos como motores de compra asociados a estas para así ahorrar la implementación de un motor propio a las compañías que organizan dichos eventos y poder dar la gestión de las entradas a una empresa especializada en la venta de entradas, dejando la organización del evento a la empresa organizadora.

Una vez la plataforma de *ticketing* administra la venta de entradas de un evento en concreto, este último pasa a ser publicitado, a su vez, por la plataforma en cuestión, haciendo que más personas tengan la posibilidad de comprar entradas del evento. Los usuarios que compran en este tipo de plataformas disponen de un espacio personal donde pueden

modificar sus datos y ver el historial de compras.

Lo que se persigue con este proyecto es dar la posibilidad a las empresas organizadoras de eventos de tener toda la *web* de un evento asociada a una única plataforma de venta de entradas, permitiendo que solo tenga que gestionar las noticias, avisos y entradas que crea desde la parte de administración y poder ver en todo momento, mediante gráficos, la venta de entradas, visitas a la *web* un evento en concreto, noticias vistas o recomendadas por los usuarios, etc.

5 METODOLOGÍA

Este apartado da una visión global de método y las fases del proyecto que se han llevado a cabo para la realización del proyecto.

5.1. Desarrollo Ágil

Para la realización de este proyecto se ha utilizado la metodología ágil, permitiendo que el desarrollo de la plataforma se hiciese de forma iterativa e incremental [4].

Una de las metodologías ágiles más implementadas en los proyectos de *software* es el SRCUM, el cual no permite elaborar etapas o *sprints* en los que se reparte una cantidad de trabajo y se analiza, al empezar uno nuevo, si se ha terminado el trabajo asociado al anterior *sprint*, el trabajo que se llevará a cabo en el actual y cual será el trabajo a realizar en las próximas etapas.

Para este proyecto se ha utilizado esta metodología en lo que respecta a la planificación por *sprints* (o bloques de trabajo) permitiendo solucionar posibles retrasos en la planificación inicial y cambiar las estimaciones hechas si es necesario.

5.2. Planificación

La planificación del proyecto ha sido dividida en cinco *sprints* de una duración media de cuatro semanas, haciéndolos coincidir expresamente con las principales entregas de documentación del trabajo de fin de grado. Esta planificación y división de las tareas ha sido llevada a cabo mediante el programa Microsoft Project, que nos permite de manera muy visual y eficaz mantener un control del estado del proyecto. A continuación, se exponen los cinco *sprints* llevados a cabo en este trabajo:

Sprint 0 – septiembre. Diseño inicial de la plantilla para eventos e instalación de todos los recursos necesarios.

Sprint 1 – octubre a noviembre. Desarrollo de la plantilla para eventos, diseño de la BD e implementación del *login/registro*.

Sprint 2 – noviembre a diciembre. Implementación del *Espacio personal*.

Sprint 3 – diciembre a enero. Implementación del *Espacio de administración*.

Sprint 4 – enero. Pruebas, experiencia con usuarios y resolución de problemas.

5.3. Roles

Para llevar a cabo este proyecto se ha tenido en cuenta los roles que, en un proyecto con un equipo real, se deberían de

poseer para la realización del mismo. Los roles necesarios para este trabajo han sido:

- Ingeniero de requisitos

Determina las características y necesidades del software a desarrollar para satisfacer a los clientes y usuarios. La carga de trabajo estimada ha sido de 16h concentradas en el primer *sprint*, permitiendo la toma de requerimientos y la validación de los objetivos.

- Diseñador

Mente creativa del equipo, satisface los requisitos que el sistema necesita y las posibilidades que el programador puede desarrollar. En este caso, a tenido una carga de trabajo más importante que el ingeniero de requisitos para poder diseñar una plataforma funcional para todo tipo de dispositivos. La estimación ha sido de 25h de trabajo.

- Programador

Encargado de desarrollar y depurar el código fuente de un programa informático y, para optimizar roles, es el encargado de testear el programa en las diferentes fases del proyecto. Este rol conlleva la mayor parte del desarrollo de la plataforma haciendo que la estimación de trabajo suba a 234h, provocando que sea el rol más importante para poder satisfacer la mayor parte de los objetivos de este trabajo.

5.4. Tecnologías utilizadas

Por lo que respecta al *back end*, la aplicación *web* ha sido desarrollada mediante el *framework* de PHP llamado Codeigniter (explicado en la sección 2 de este artículo) y, para la BBDD, se ha utilizado MySQL.

En cuanto al *back end*, hay que remarcar que la aplicación ha sido diseñada e implementada para todo tipo de navegadores *web*, ya sea para dispositivos de sobremesa, como ordenadores y portátiles, como dispositivos móviles, tabletas o *smartphones*. Esto ha sido posible gracias a la utilización de un diseño *web* adaptable o *responsive*, al lenguaje CSS y al *framework* Bootstrap que nos permite elaborar diseños *web* muy adaptativos y sencillos de programar.

A lo largo del desarrollo de la plataforma, se han ido añadiendo diferentes librerías y *plugins* adicionales tanto para mejorar la experiencia del usuario en la *web* como para poder ofrecer mejores servicios a nuestros clientes. Y son los siguientes:

- HTML2PDF. Librería para generar PDFs mediante HTML. Utilizado para generar el PDF de un *ticket* [5].
- qrlib.php. Librería para la generación de códigos QR. Utilizado para la generación de las entradas [6].
- jQuery Payment. *Plugin* para validar formularios de pago, donde se valida si la tarjeta introducida es correcta [7].
- Highchart.js. Librería que permite elaborar gráficos estadísticos para la parte de administración de la plataforma [8].
- Alertify.js. Librería que permite generar ventanas emergentes en cualquier posición de la pantalla para mostrar avisos y notificaciones. [9].

En cuanto a los programas utilizados para la realización de la plataforma se encuentran: como IDE, el programa Netbeans por la facilidad y sencillez que aporta a la hora de desarrollar código, especialmente en lenguajes como PHP o Java; XAMPP, un instalador de Apache, PHP y la base de datos MySQL que permite contar con estos programas ya instalados y configurados y, por último, phpMyAdmin como gestor de base de datos MySQL, ya que es una herramienta muy completa con una interfaz web sencilla e intuitiva.

Cabe destacar que todas las tecnologías utilizadas y mencionadas en este apartado son gratuitas, la mayoría de ellas tienen licencias *Open Source* que nos permiten utilizar las librerías y *plugins* de forma gratuita. Solo hay una excepción sobre la librería Highcharts que, en el caso de querer dar un uso comercial a la plataforma, se debe de contactar con los creadores y pagar la licencia correspondiente.

5.5. Diseño de la plataforma

Una parte que se ha tenido muy en cuenta en el desarrollo de todo el proyecto ha sido el diseño de la plataforma.

Este, ha sido ideado para ser funcional, intuitivo y fácil de entender por todos los usuarios potenciales que puedan llegar a una plataforma de *ticketing*. Para ello, se ha diseñado una plataforma funcional para todo tipo de pantallas, sin importar el tamaño de esas, dando prioridad a los dispositivos mas grandes (ordenadores y portátiles) pero sin olvidar el sector móvil (*tablets*, *phablets* y *smartphones*) que también juegan un papel importante en las compras *online*.

Para ello se han elaborado para cada una de las vistas de la plataforma tres diseños o *wireframes* diferentes, es decir, se ha diseñado cada una de las vistas teniendo en cuenta los tres tipos de pantalla más habituales en el mercado: ordenador (grande), *tablet* (mediano) y *smartphone* (pequeño)[11].

De esta forma, se permite lograr que en las pantallas más pequeñas se vea el contenido más concentrado, eliminando cierta información no necesaria y creando una interfaz en sintonía con las aplicaciones móviles habituales.

En el caso de las pantallas grandes o de tamaño medio, estas comparten muchas similitudes estéticas, aunque, en las segundas, le estructuración del contenido también se ve afectada debido a la falta de espacio.

6 DESARROLLO

Como se ha comentado a lo largo de este artículo, la complementación de la plataforma ha sido dividida en tres bloques: plantilla para eventos, espacio para usuarios y espacio de administración.

En esta sección se explican por separado cada uno de los bloques, su desarrollo y funcionalidades para así poder tener una visión más extensa de la plataforma. Cabe destacar que la plataforma de *ticketing* desarrollada en este trabajo ha sido denominada como ADCTickets, y así se hace referencia en los siguientes puntos del artículo.

ADCTickets.

Fig. 1: Logo de ADCTickets.com

6.1. Plantilla para eventos

El primer paso en la implementación de ADCTickets ha sido la plantilla para eventos, en la cual se han fijado los cimientos de lo que serán las *webs* de cada uno de los eventos que estén disponibles en la plataforma.

En este caso, se ha desarrollado la pantalla inicial, la sección de entradas, noticias y registro/login. El resto de secciones propias de un evento (contacto, horarios, localización, etc.) al no ser esenciales para la implementación de este proyecto no se han llegado a crear.

En la sección de entradas, y la de noticias, se hace una llamada a la BBDD para obtener las que están habilitadas, según la preferencia preestablecidas por los administradores del evento. Haciendo que puedan variar en tiempo real los registros que aparecen en la *web*.

En el caso del registro y el *login*, se han desarrollado diferentes formularios para cada una de las funcionalidades. Esta sección tiene una vista propia, y se puede acceder a ella desde cualquier punto de la *web*. En el caso de que se registre un usuario, al validar que acepta las condiciones de la *web* e indicar que es mayor de edad, se le envía un correo confirmando su alta en la plataforma y, dado que se ha registrado desde la *web* de un evento, este evento pasa a formar parte de sus eventos favoritos (funcionalidad explicada en el siguiente bloque). Por otro lado, en el *login* se validan las credenciales del usuario mediante un correo y la contraseña asociada, que es cifrada en md5 para poder dar más garantías de seguridad a los usuarios [10].

A la hora de comprar entradas, si el usuario ya se ha registrado en ADCTickets, accederá a la vista de compra de entradas donde se muestran los datos básicos de cada una de ellas y se le da la opción de seleccionar el número de entradas que desee, mínimo una. Al realizar este paso tendrá que introducir los datos de la tarjeta con la que quiere realizar el pago, será validada y se aceptará el pago en el caso de que haya aceptado las condiciones y políticas de la *plataforma*. Con la entrada ya pagada, se redirige al cliente, por un lado, a su espacio personal (como usuario registrado) y, por el otro, se abre una nueva pestaña donde se genera el *ticket* en PDF con las entradas asociadas a este, para poder ser guardado y/o imprimido. A su vez, se envía un *mail* a la cuenta de correo que utiliza el usuario para iniciar sesión en la plataforma donde se adjunta el mismo PDF para que pueda tener siempre a su disposición de una forma más práctica.

Además, tanto en la *web* del evento, como en los posteriores bloques que se explican a continuación, el usuario tiene la posibilidad de poder cambiar de idioma cuando lo crea oportuno. Esta funcionalidad solo afecta a la *web* en sí, no a sus contenidos como noticias y avisos generados por los administradores, que solo se encuentran en el idioma en el que han sido publicados. Tanto en el registro, anteriormente mencionado, como cuando cambia de idioma al estar

registrado, se almacena en la BBDD para saber cuál es el idioma por defecto que utiliza el usuario.

6.2. Espacio para usuarios

Una vez el usuario inicia sesión desde la *web* de un evento o desde la propia *web* de ADCTickets accede al panel central dividido en dos partes: la barra de secciones del espacio personal, donde se encuentran el historial de *tickets* comprados, las noticias, los avisos, los eventos favoritos y editor de los datos de la cuenta y, a la derecha de la barra, se encuentran las últimas noticias y avisos publicados, según los eventos que tiene el usuario como favoritos, y las últimas compras realizadas.

También se encuentra un *Feed RSS* de noticias, el cual permite al usuario añadirlo a su aplicación *RSS*, blogs y demás aplicaciones para poder estar siempre informado de las noticias sobre sus eventos favoritos.

Tanto si accede desde esta pantalla, o desde la sección de avisos, puede ver el contenido de un aviso gracias a un *dialog* de pequeñas dimensiones que aparece si selecciona un aviso concreto. Además, si decide guardar el aviso, será destacado en las primeras posiciones del listado de avisos, siempre que acceda a la sección de avisos.

En el caso de las noticias, el usuario tiene en todo momento cuatro opciones: compartir la noticia en Facebook y/o Twitter, clicar en un botón con forma de corazón para dar me gusta a la noticia (en el que se ve el número de *likes* total) o poder ver el contenido de la noticia, al ser redirigido a la vista de la noticia específica.

Por último, el usuario registrado puede acceder al historial de entradas compradas. En esta sección se muestran las entradas en una tabla en la que se puede buscar, en el caso de que el usuario haya hecho muchas compras, mediante un pequeño buscador propio de la misma tabla. Si quiere ver el contenido del *ticket* se generará el PDF con las entradas asociadas a esta compra.

La sección de eventos favoritos está pensada para que el usuario pueda seleccionar, de un listado de todos los eventos de la plataforma, de qué eventos quiere ver noticias y avisos (y recibir correos con las últimas publicaciones) para estar lo más informado posible.

El usuario registrado también puede editar sus datos personales mediante la validación del correo electrónico y la contraseña de este.

6.3. Espacio de administración

En este último bloque se han generado los roles de toda la plataforma: *root* para los administradores de ADCTickets, *admin* para los administradores que crean y modifican entradas, noticias y avisos y *grafic* para los administradores que pueden acceder a los gráficos relativos a sus eventos asociados. Además se da el rol de *user* a los usuarios que se registran en la plataforma de normal.

Los administradores de ADCTickets son los únicos que pueden crear nuevos usuarios. Una vez se crea un usuario (con correo y contraseña predefinidos) se envía un correo al nuevo administrador para que pueda acceder a la plataforma y cambiar la contraseña por una nueva. Cabe destacar que los usuarios administradores no pueden comprar entradas ni acceder al espacio de usuarios dado que no tienen este rol

asociado.

Un administrador puede crear entradas (introduciendo el nombre, descripción, fotografía adjunta y número de entradas a la venta), modificarlas o inhabilitarlas para que los usuarios no puedan comprarlas más.

Pueden hacer lo mismo con las noticias y con los avisos. Sin embargo, cuando el administrador inserte esta nueva publicación el sistema enviará un correo a todos los usuarios que tienen como favorito este evento para que vayan a ver en la plataforma la nueva información relativa al evento.

Para la visualización de los gráficos, se pueden ver los datos relativos a las visitas recibidas en la *web* del evento, los usuarios únicos que la han visitado, el tiempo por sesión que han estado, entre otros datos.

Por lo que respecta a los gráficos de las entradas, se pueden ver de forma general (resumen y comparación de todas las entradas) o de forma individual (datos relativos a una sola entrada). Tanto de una forma como de otra se pueden ver los datos por género de los usuarios que han comprado entradas, el tipo de entrada más adquirida en un país, edades que más compran, la modalidad de entrada que da más dinero o la que se vende más, entre otros valores interesantes que se pueden extraer de los datos de los usuarios y de sus compras.

En el caso de los avisos y las noticias se pueden ver los mismos tipos de gráficos: una primera estadística del número de visitas recibidas comparado con los *likes*, en el caso de las noticias, o el número de personas que ha guardado un aviso y, en definitiva, el tipo de usuarios que ha visitado las publicaciones y cuando lo han echo. Poder conocer todos estos datos resulta interesante, ya que permite saber qué noticias resultan más atractivas para nuestros clientes, y cuáles son los avisos más vistos para, así, poder mejorar cómo se comunica y se informa de las noticias relacionadas con los eventos a los asistentes.

7 PROBLEMAS ENCONTRADOS

En la implementación de la plataforma no han ocurrido grandes imprevistos que hayan afectado la planificación o las funcionalidades iniciales que se especificaron en el informe inicial. Pero se pueden resaltar ciertos problemas que han tenido que ser solventados para la óptima realización del proyecto, a continuación se explican las tres más importantes:

- Para el envío de los correos electrónicos se especificó, en un inicio, utilizar un servidor SMTP para poder realizar los envíos de las entradas, noticias, avisos y del alta de usuarios. En un primer momento el servidor que se configuró funcionó con los primeros *mails* que se enviaron pero luego se perdía la comunicación con el programa y el servidor. Ante esta situación se decidió crear una dirección de correo llamada *adcticket@gmail.com* desde la cual se envían los correos que se han ido enviando con las pruebas realizadas durante toda la implementación de la plataforma.
- La generación de los PDF con los códigos QR asociados, para la creación de las entradas, ha llevado cierto tiempo de investigación. En el caso de los códigos QR, no se podían generar en la página HTML donde está el formato de las entradas. La solución encontrada

ha sido crear el código QR antes de generar el PDF, guardarlo y, una vez se ha generado el *ticket*, eliminarlo para que no se almacenen todos los códigos que se han ido generando en los meses que se lleva realizando el proyecto.

- Los formularios que se encuentran en la plantilla para eventos y en la *web* de ADCTickets son diferentes si accedemos desde el ordenador o desde un *smartphone*. El hecho de hacer una *web responsive* denomina que se deban que diseñar diferentes vistas según el tamaño de pantalla de diferentes dispositivos. En el caso de los formularios de registro y pantallas más pequeñas, ha hecho que se oculten y aparezcan diferentes formularios (estéticamente iguales) para que el diseño de la vista sea óptimo en todo momento.

8 RESULTADOS

A continuación se muestran los resultados obtenidos de la implementación de la plataforma ADCTickets, validando cada uno de los objetivos expuestos en la subsección 3.1 Objetivos de la plataforma de este informe.

Diseño de la plataforma

Mediante la utilización de *wireframes* antes de implementar cada uno de los módulos hemos conseguido acortar los tiempos de implementación y poder dar a toda la plataforma un aspecto moderno, funcional e idéntico para todas las vistas que la conforman.

Este diseño ha perseguido familiarizar al usuario con el estilo único que tiene ADCTickets, permitiendo que la navegación entre eventos, ADCTickets y la pantalla de administración se vea como una única *web de ticketing* con diferentes estilos en los eventos, a gusto de la empresa organizadora del evento.

Para poder entender mejor el diseño de la plataforma, se puede consultar el Apéndice 2 donde se muestran los resultados obtenidos, una vez ha sido finalizada la aplicación, comparándolos con los bocetos que se llevaron a cabo en el inicio de cada bloque.

Creación y diseño de la BBDD

Para diseñar la BBDD de la plataforma había dos opciones para implementar: hacer una base de datos por cada evento que se fuese incorporando en la plataforma y poder tener más estructurados los datos de cada uno de los eventos (aunque estas compartirían una base de datos común donde, por ejemplo, se encontrarían los datos de los usuarios) o diseñar una base de datos conjunta en la que primen la utilización de las mismas tablas para diferentes eventos.

Se decidió decantarse por la segunda opción, una base de datos común, donde podemos destacar las tablas de usuarios (donde almacenamos sus datos personales), evento (donde se encuentra la información básica de este) y *ticket* (donde se almacenan las compras realizadas por los clientes, aunque depende en gran medida de la tabla *entradas* donde se guarda la información de cada una de las entradas asociadas a un *ticket*).

También cabe resaltar que las noticias y avisos se almacenan en la BBDD y los registros que se hacen de cada visita, *likes* o guardado de un aviso por parte de un usuario registrado se guardan en las tablas *avisousuario* y *noticiausuario* donde queda constancia de las interacciones que se hacen en la plataforma.

Además, cada vez que se visita alguna de las vistas de la aplicación, se guarda el número de sesión y la IP del usuario para así registrar el número de visitas realiza por sesión y el tiempo que está visitando un evento concreto o la *web* de ADCTickets.

En el Apéndice 3 se puede encontrar la estructura de la BBDD resultante de la plataforma ADCTickets.

Implementación de la plantilla *web* para alojar la información del evento

La plantilla para eventos nace con el objetivo de tener un estándar de página *web* para los nuevos eventos que se pueden ir incorporando a la aplicación. En el desarrollo de este proyecto se ha creado una *web* simple para el festival de música FIB que nos ha permitido jugar con diferentes tipos de entradas, noticias variadas y un diseño acorde con el espíritu del evento.

Lo que se ha tratado de conseguir con esta plantilla es poder utilizar todo el código desarrollado en estos meses de trabajo para que, cuando se añada un nuevo evento en la aplicación, haya unas bases sobre cómo tiene que ser la *web* y qué estilo de programación y diseño va a seguir. Hay que resaltar que no es un diseño cerrado, es decir, la capacidad de personalización que se le puede dar a un nuevo evento es ilimitada dado que el desarrollo de la nueva *web* será a gusto del organizador del evento, solo compartirá funciones similares en las secciones de entradas, noticias y registro/login de la *web* (Figura 2).

Fig. 2: Registro en la *web* del FIB

Además, se han conseguido implementar diferentes idiomas en las vistas del evento, de manera que, según las preferencias del evento y el público al que vaya orientado, se puedan incorporar nuevos idiomas a la plataforma bajo petición expresa de los organizadores.

No solo se puede modificar el diseño y el lenguaje de la *web* a petición del organizador, sino que también se pueden añadir métodos de pago adicionales a los ya existentes (se ha implementado únicamente el pago con tarjetas de crédito).

En el caso del formato de las entradas ocurre lo mismo. Por ejemplo, el código QR que se encuentra en cada una de las entradas compradas por un cliente puede ser substituido

o compartido con un código de barras, que suele ser más común en eventos y productos en el mercado. Véase la figura 3 (Entrada en formato PDF) donde se puede apreciar el diseño final que se ha implementado.



Fig. 3: ABONO de 4 días para el FIB

Implementación del Espacio personal

El espacio personal se puede encontrar en dos partes de la plataforma: en la *web* del evento (Figura 4) o en la *web* madre de ADCTickets.

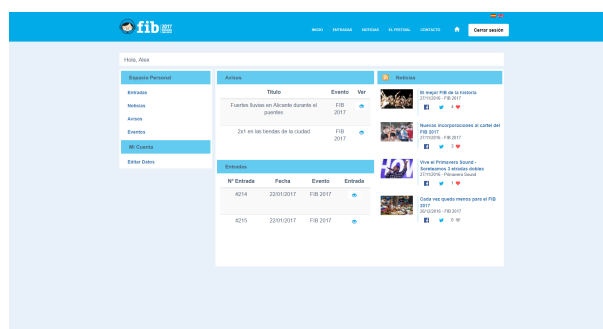


Fig. 4: Pantalla inicial del espacio para usuarios (Evento FIB)

Se ha implementado de esta forma para que el usuario que ha accedido a la *web* de un evento no sea redirigido a la *web* de ADCTickets cuando inicia sesión para comprar una entrada o ver las últimas noticias del evento en cuestión. Por lo tanto, no hay necesidad de que los usuarios accedan a la *web* de ADCTickets para iniciar sesión en la plataforma, ya que pueden hacerlo desde cualquier vista.

Una vez que el usuario accede tiene la posibilidad de, entre otras funciones, comprar una entrada para cualquier evento disponible en la plataforma y seleccionar de qué eventos quiere recibir las últimas publicaciones.

En el caso de las noticias se ha echo una primera aproximación al mundo de las redes sociales gracias a la posibilidad de compartir con tus amigos y seguidores las noticias de la plataforma. Además, para dar un toque más social a la aplicación, se puede ver el número de *likes* actuales que tiene una noticia, como se puede apreciar en la Figura 5.



Fig. 5: Posibilidades de compartir una noticia

Aunque es una funcionalidad muy limitada, se ha querido dar los primeros pasos en cuanto a red social se refiere para que, en un futuro, los usuarios puedan dejar comentarios en una publicación o exista un *timeline* de noticias con una mayor participación de los usuarios.

Otra funcionalidad implementada satisfactoriamente ha sido la creación de un *feed* de noticias personalizado (Figura 6) que se puede encontrar justo al entrar en el espacio personal de un usuario, permitiendo que este pueda visitar el enlace cuando quiera y disfrutar de las noticias sobre sus eventos favoritos desde cualquier aplicación RSS.

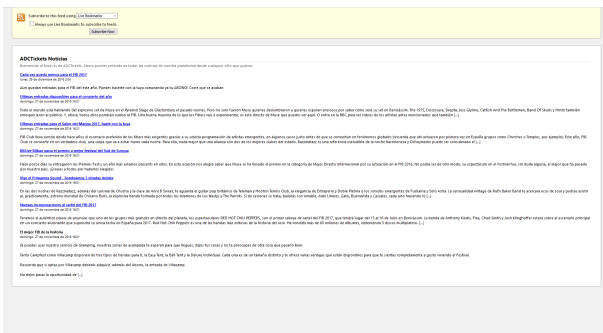


Fig. 6: Feed RSS de noticias

Implementación del Espacio de administración

Este bloque ha sido la culminación y validación de todo lo que se ha ido implementado a lo largo de los dos bloques anteriores. Esto significa que se han creado los roles que rigen toda la plataforma, la creación y validación de las entradas noticias y avisos que se han utilizado para hacer las funcionalidades de los demás módulos y la puesta en marcha de los gráficos mediante los datos recogidos durante todas las pruebas que se han ido ejecutando. En la parte de administración de usuarios, solo accesible para los administradores de ADCTickets, se pueden crear nuevos usuarios o dar de baja los actuales.

Estos nuevos usuarios, a los cuales se les asignan los eventos que pueden administrar, pueden ver el listado de clientes que han comprado una entrada para uno de sus eventos, saber el número de entradas que llevan vendidas hasta la fecha y así poder tomar decisiones sobre promociones especiales, creación de nuevas entradas para satisfacer la demanda, entre otras posibilidades.

Por lo que respecta a las publicaciones, pueden crear nuevas noticias o avisos sin límite. Además, es posible saber la cantidad de visitas que han recibido y, en el caso de que haya habido errores en la redacción de la publicación

se puede cambiar el texto esta, así como las fotografías adjuntas.

Por otro lado, la parte más interesante es la posibilidad de ver los datos que se acaban de comentar de forma ampliada y visualmente más atractiva mediante gráficos estadísticos. Como se puede ver en la Figura 7, es posible extraer múltiples datos sobre la compra de entradas que pueden ayudar en gran medida a saber orientar el evento, la venta de entradas y la publicidad del evento.



Fig. 7: Gráficos de las entradas disponibles para el FIB 2017

También, se pueden ver gráficos individuales, ya que se ha creído conveniente que haya entradas o noticias más relevantes que otras y que pueden ser tratadas de una forma más específica y aislada tal como se puede observar en la

Figura 7, mencionada con anterioridad.

Test de cada uno de los módulos de la plataforma

Las pruebas se han ido haciendo a lo largo del desarrollo de este proyecto. En concreto, al terminar cada uno de los módulos se realizaban varias pruebas de usabilidad y de las nuevas funcionalidades para que, una vez se empezase con el nuevo módulo a implementar, no hubiese problemas de compatibilidades entre las nuevas funciones, que muchas veces dependían del código ya desarrollado, ya que se ha podido reutilizar, en ciertas ocasiones, código de otros módulos.

Test de la funcionalidad de la plataforma por parte de usuarios con diferentes aptitudes

Una vez el usuario ha podido comprobar las funcionalidades y la usabilidad de la aplicación desarrollada se le ha pasado una extensa encuesta sobre los aspectos que se consideran más relevantes de la plataforma, para poder saber si el trabajo realizado en este proyecto puede satisfacer a los usuarios potenciales de un tipo de plataforma de *ticketing* como la que se ha creado en este trabajo de final de grado.

Para poder llevar a cabo la encuesta se ha utilizado la aplicación *Google Forms* que permite elaborar encuestas de forma *online* de forma sencilla, rápida y gratuita.

En cuanto a los usuarios han participado en la encuesta, los cuales han sido escogidos de manera aleatoria, la muestra ha sido de siete personas con edades comprendidas entre los 21-24 años. Han podido probar, sin límite de tiempo y preguntas, toda la aplicación *web* desarrollada pudiendo acceder como usuarios a comprar entradas o como administradores para poder gestionarlas.

Los objetivos principales de esta encuesta han sido poder obtener *feedback* de los usuarios, constatar que las funcionalidades implementadas útiles y han sido implementadas a gusto de los usuarios, constatar que el diseño de toda la plataforma es correcto y funcional y poder saber, gracias a comentarios escritos por los propios encuestados, que se puede cambiar y/o mejorar desarrollo realizado.

Los resultados obtenidos se pueden constatar en los siguientes puntos:

- La interfaz creada y llevada a cabo para ADCTickets es funcional, con contenido entendible, sencilla y adecuada para el tipo de web que se ha elaborado.
- Los *mails* que se envían cuando se genera una nueva noticia o aviso y cuando se da de alta un nuevo usuario son valorados por los usuarios como muy útiles y funcionales.
- El diseño adaptable o *responsive* de las *webs* son un punto excelente para poder tener una plataforma funcional en cualquier dispositivo y sistema operativo, sin necesidad de hacer *apps* específicas.
- A los usuarios les gusta compartir contenido que consideran relevante con sus amigos de las redes sociales. En el caso de ADCTickets, se tendrá que valorar la implementación de nuevas funcionalidades para mejorar la interactividad entre usuarios.

- A colación con el anterior punto, los me gusta o *likes* que se pueden hacer sobre las noticias son aprobados por los usuarios, aunque se puede observar cierta disparidad de opciones sobre su utilidad, dada la poca idea de red social con la que se ha diseñado ADCTickets.
- En cuanto al pago de las entradas, se ve una clara discrepancia entre los usuarios sobre el método de pago ideal. El método de pago, solo posible con tarjetas de crédito, es insuficiente para los intereses y necesidades de los posibles clientes.
- Con el *Feed RSS* de noticias podemos encontrar, de nuevo, una disparidad de opiniones sobre la utilidad de esta funcionalidad que, aunque sea poco conocida para el gran público, es una buena forma de llegar a más público que sí que utiliza este tipo de aplicaciones diariamente.
- Los gráficos de la sección de administración han tenido una buena acogida por parte de usuarios inexpertos con este tipo de funcionalidades, valorando el contenido que se muestra como útil y relevante.
- Por último, en la parte de comentarios escritos por los encuestados, podemos resaltar la importancia de crear una comunidad de usuarios que pueden interactuar entre ellos, permitiendo que la aplicación ADCTickets pueda albergar funcionalidades más orientadas a las redes sociales.

Añadir nuevas funcionalidades

Por último, cabe destacar que cuando se definieron los objetivos de este trabajo se contemplaron unas funcionalidades muy generales como la venta de entradas, la posibilidad de crear noticias y avisos o hacer que el usuario inicie sesión en la *web* para poder acceder a más funcionalidades ocultas para los usuarios que no son clientes.

A medida que se iba diseñando la plataforma se fueron añadiendo nuevas funcionalidades que se han ido comentando a lo largo de este artículo, como, por ejemplo: el envío de *mails* cuando se crean noticias y avisos, *Feed RSS* de noticias, generación de códigos QR o la implementación de *plugins* para poder validar tarjetas de crédito en el formulario de pago de entradas.

Esto ha sido posible gracias al método de organización utilizado, que nos ha permitido ir añadiendo nuevas funcionalidades en cada *sprint* sin que se viesen afectados los tiempos del proyecto.

9 CONCLUSIONES

Como se ha podido comprobar a lo largo de este artículo, todos los objetivos marcados en el inicio del proyecto han sido llevados a cabo de forma satisfactoria, tanto los que fueron considerados como críticos para la óptima realización de la aplicación, como los secundarios, que podrían no haberse ejecutado por falta de tiempo o una mala organización por la carga de trabajo real de las funcionalidades establecidas.

Se puede señalar que han habido pequeñas desviaciones de tiempo en lo que respecta a la planificación del proyecto por el diseño que se tenía que hacer de la plataforma. En un principio se estableció que, por parte del diseñador, se crearían los diseños de los 3 bloques de ADCTickets en el *sprint* 0 para así poder tener la base de la plataforma desde el principio. Pero, al empezar a plantear la plataforma y al ir surgiendo nuevas funcionalidades no previstas en un inicio, se decidió dividir el diseño de los bloques en dos *sprints* diferentes (como se puede observar en el diagrama de Gantt del apéndice A) para así poder crear unas vistas más acordes a las funcionalidades que se iban planteando a lo largo de los meses que ha durado el proyecto.

En cuanto a la ampliación de conocimientos por parte del programador, no ha sido necesaria dada la experiencia previa que se tenía en las tecnologías utilizadas permitiendo utilizar la mayor parte del tiempo en el desarrollo de la plataforma. Por lo tanto, se ha querido expresar las posibilidades que nos puede dar CodeIgniter y demostrar que una plataforma de *ticketing*, pese a la complicación que se puede dar en una aplicación de esta magnitud a nivel de funcionalidades, se puede crear desde la nada una plataforma completamente funcional, con ciertas limitaciones, que puede llegar a ser un referente en la venta *online* de entradas.

9.1. Líneas de mejora

Aunque este trabajo se haya centrado en cimentar las bases de una nueva plataforma de venta de entradas *online*, se han dejado de lado muchos aspectos que por falta de tiempo no se han considerado como posibles de realizar en estos meses de trabajo.

Una de las mejoras que se deben realizar después de este trabajo es la inserción de un pago real de las entradas y la posibilidad de poder pagar mediante PayPal, muy utilizado y cada vez más aceptado por los usuarios, para poder dar múltiples opciones de pago a los clientes.

También se debe de valorar la posibilidad de poder comentar, a modo de red social, las noticias que se publican en la plataforma, permitiendo que los administradores del evento tengan la posibilidad de responder a los usuarios y conseguir un *feedback* que mejoraría la relación organización-cliente.

Por último, debido al tiempo que se ha tenido para realizar este proyecto, los roles que se han creado en esta primera versión son insuficientes para las posibles funcionalidades que puede llegar a tener una plataforma de estas características. La solución que se debe de llevar a cabo pasa por dividir aún más los roles, en especial los de administrador, para que las funcionalidades a las que se puede acceder sean más acordes con las necesidades de las empresas que tienen sus eventos en la plataforma.

AGRADECIMIENTOS

Agradecer al tutor de este trabajo, Marc Talló, su ayuda y consejos en la elaboración de documentos y el enfoque del trabajo que se ha ido desarrollando a lo largo de estos meses de trabajo.

También me gustaría agradecer a todas las personas que se han ido interesando por el proyecto a lo largo de la imple-

mentación, así como destacar la colaboración en las pruebas realizadas con usuarios, por parte de amigos y familiares, y la posterior realización de las encuestas.

REFERENCIAS

- [1] Diario Expansión, "La venta de entradas crece un 20% en el primer semestre". (Julio de 2016) [En línea] Disponible en: <http://www.expansion.com/sociedad/2016/07/20/578fa82946163f7c668b4600.html>
- [2] Nyhetssook, "El auge del sector de la venta de entradas por Internet". (Enero de 2016) [En línea] Disponible en: <https://nyhetssook.com/el-auge-del-sector-de-la-venta-de-entradas-por-internet/>
- [3] Mario Fontán, "CodeIgniter, un framework PHP para el desarrollo rápido de aplicaciones web.". (Octubre de 2012) [Online] Disponible en: <http://www.adwe.es/codigo/codeigniter-framework-php-desarrollo-aplicaciones-web>
- [4] Ophelia Pastrana, "5 beneficios de aplicar metodologías ágiles en el desarrollo de software". (2015) [En línea] Disponible en: <http://www.i2btech.com/blog-i2btech-deployment/5-beneficios-de-aplicar-metodologias-agiles-en-el-desarrollo-de-software/>
- [5] "5PDF desde PHP: HTML2PDF (1/5)". (Abril de 2016) [En línea] Disponible en: <https://huguidugui.wordpress.com/2015/04/06/pdf-desde-php-html2pdf-15/>
- [6] Sandra SCS, "Generar imágenes QR con PHP usando qrlib.php". (Septiembre de 2015) [En línea] Disponible en: <http://sandritascscs.blogspot.com.es/2015/09/generar-imagenes-qr-con-php-usando.html>
- [7] jquery.payment (Plugin) [En línea] Disponible en: <https://github.com/stripe/jquery.payment>
- [8] Highcharts.com [En línea] Disponible en: <http://www.highcharts.com/>
- [9] Alertifyjs.org [En línea] Disponible en: <https://alertifyjs.org/>
- [10] Iván Martínez, "Qué es MD5, cómo funciona y para qué se usa". (Septiembre de 2013) [En línea] Disponible en: <https://rootear.com/seguridad/md5-como-funciona-usos>
- [11] Santi, "Wireframes: Que son y como crearlos". (Octubre de 2015) [En línea] Disponible en: <http://webdesdecero.com/wireframes-que-son-y-como-crearlos/>

APÉNDICE

A.1. Diagrama de Gantt

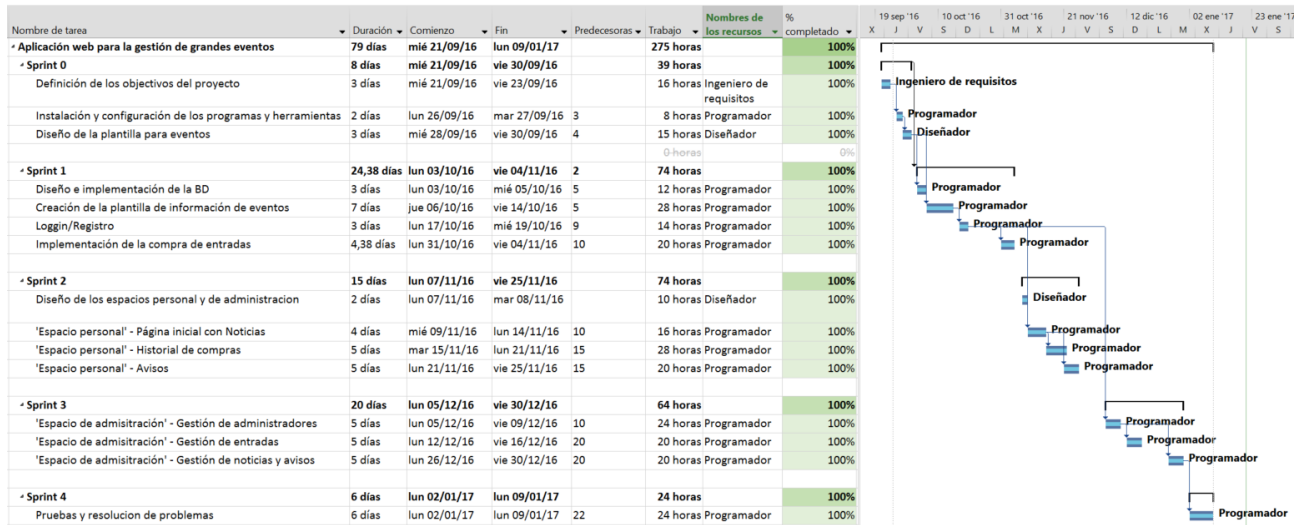


Fig. 8: Diagrama de Gantt

Como se puede observar en la figura 8, el proyecto ha sido dividido en cinco *sprints* con cargas desiguales de trabajo, según la importancia que estos tienen en la implementación de la plataforma ADCTickets.

La carga de trabajo más grande, y por tanto más importante para ser llevada a cabo, ha sido concentrada en los *sprints* 1, 2 y 3 que hacen referencia a la plantilla para eventos, el espacio para usuarios y el espacio para administradores, respectivamente. Los tres *sprints* son responsabilidad del programador, el cual acapara casi la totalidad de la carga de trabajo.

Por lo que respecta a los otros dos roles involucrados en la implementación, el ingeniero de requisitos y el diseñador, se encuentran en el *sprint* 0 para el primero (toma de requerimientos) y en los *sprints* 0 y 2 en el caso del diseñador.

A.2. Wireframes iniciales y resultado final

En este apéndice se pueden ver algunos de los bocetos que se llevaron a cabo para diseñar la plataforma ADCTickets y el resultado logrado al final de la implementación. De cada ejemplo, se pueden ver las diferentes vistas según el tipo de pantalla (grande, mediana o pequeña).

El primer ejemplo (Fig. 9) corresponde a la página inicial del festival de música FIB 2017 creado durante el segundo *sprint*:



Fig. 9: Página inicial del FIB

También podemos ver en la figura 10 la pantalla de compra de entradas donde se seleccionan el número de entradas que el usuario desea. Como se puede apreciar en la versión móvil, la información menos relevante desaparece para poder hacer una vista más limpia y que permita al usuario poder tener en todo cualquier dispositivo la información esencial a su alcance.



Fig. 10: Página de compra de entradas del FIB

Entrando en la parte del espacio de usuarios, en la figura 11 se muestra la vista de una noticia de la plataforma concentrando la información en las pantallas más pequeñas pero siguiendo manteniendo la armonía base de la plataforma en todo momento.

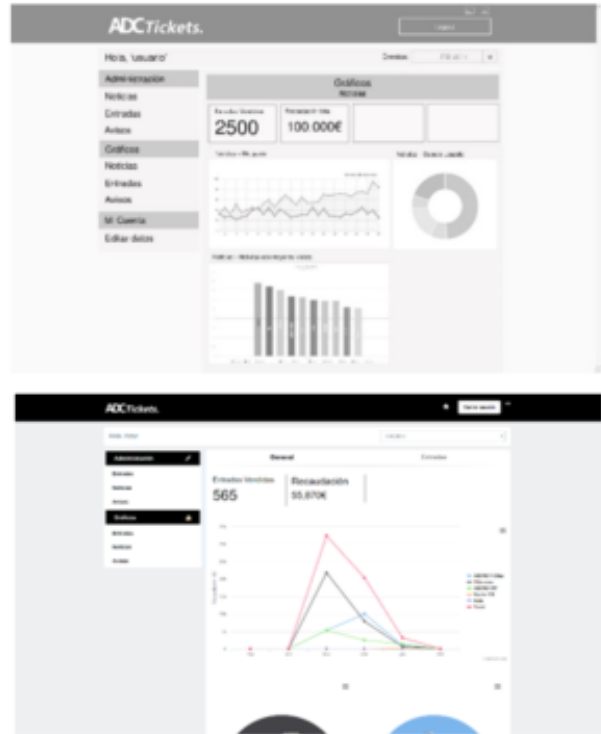


Fig. 11: Vista de una noticia en ADCTickets

Fig. 12: Gráficos sobre las entradas del FIB 2017

Por último, del espacio de administración, no hay que dejar pasar por alto los gráficos que están disponibles para los administradores de un evento. En este caso, no se ha dado prioridad a optimizar las vistas para las pantallas más pequeñas ya que se espera que este tipo de secciones solo se utilicen desde dispositivos más grandes como portátiles u ordenadores de sobremesa aunque, como se puede apreciar, el resultado es más que satisfactorio en cualquiera de las vistas que se pueden ver en la figura 12.

A.3. Base de datos

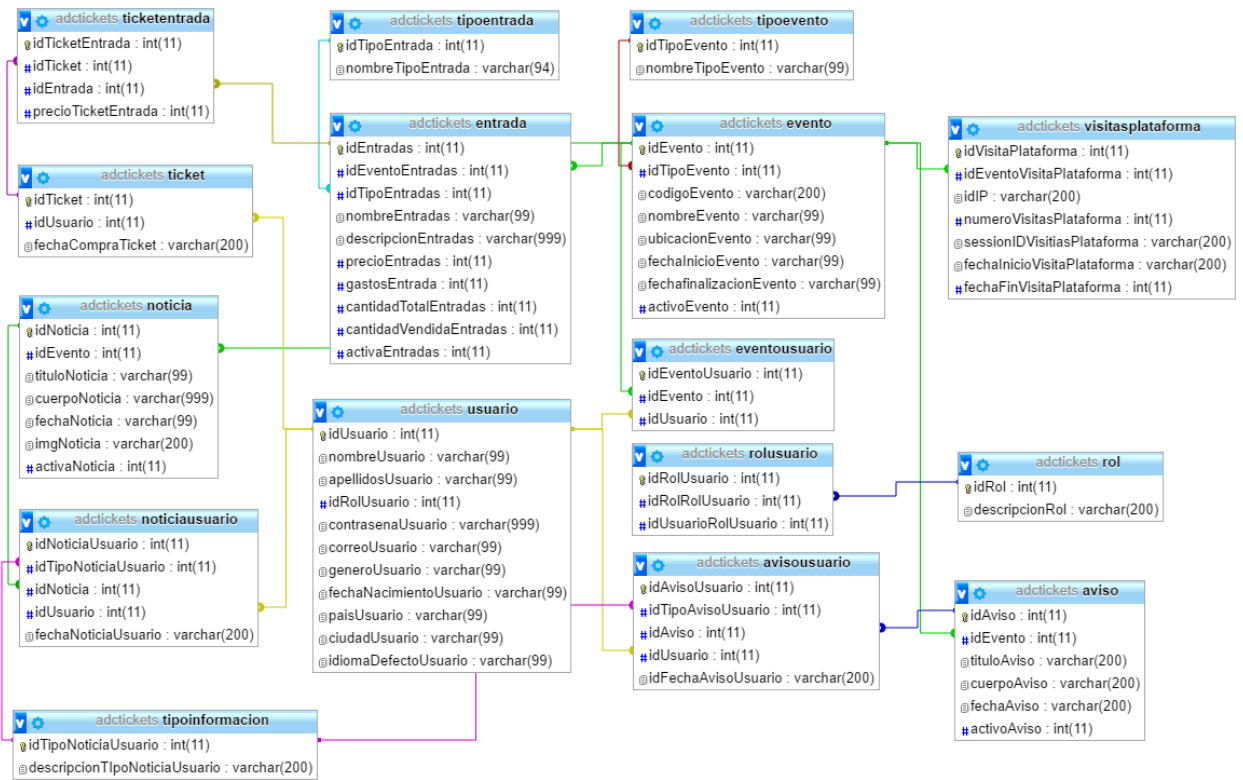


Fig. 13: Diagrama de la BBDD de ADCTickets