



Universidad
Zaragoza

Trabajo Fin de Grado

Aplicación de gestión y venta de entradas del
Memorial Pepe Carroll

Autor/es

Santiago Peralta Martínez

Director/es

Dr. Francisco Javier Fabra Caro

Escuela de Ingeniería y Arquitectura

Zaragoza
Diciembre de 2018



DECLARACIÓN DE
AUTORÍA Y ORIGINALIDAD

[Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación].

D./D^a. Santiago Peralta Martínez

con nº de DNI 18057691T en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)
Grado en Ingeniería Informática, (Título del Trabajo)

Aplicación de gestión y venta de entradas del Memorial Pepe Carroll

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 22 de noviembre de 2018

Fdo: Santiago Peralta Martínez

RESUMEN

La realización del presente proyecto supone, por una parte, llevar a la práctica los conocimientos adquiridos a lo largo del Grado, así como en lo posible, tratar de explorar y añadir nuevos conocimientos en materia de desarrollo de aplicaciones web que, además, se integre con una pasarela de pago real.

El punto de partida de este proyecto surge de la necesidad de realizar una aplicación web que automatice la gestión y ventas para el Memorial “Pepe Carroll”. Como resultado final, se obtiene una aplicación web que da soporte a la Asociación Mágica Aragonesa para la gestión del citado evento, pero con el valor añadido de que, con modificaciones pequeñas, pueda ser fácilmente adaptado a cualquier tipo de actividad cultural en la que se puedan adquirir entradas y otros productos relacionados con el evento.

Tras las dos anteriores ediciones del memorial Pepe Carroll, surgió un problema y es que los sistemas que se usaban para la gestión de las entradas y la venta de entradas no permitían la modificación de las entradas, asignando una compra por modificación. Esto hacía que fuera muy caótico para el administrador del evento y el usuario. Por este motivo se aporta esta solución, una aplicación web que permite la gestión y automatización de las entradas.

Para la solución se ha decidido hacer uso de la tecnología MEAN, que está basada en MongoDB, ExpressJS, AngularJS y NodeJS, ya que permite realizar una implementación fullStack desde el análisis hasta el despliegue. Por último, al ser una aplicación web se ha tenido que desplegar en producción, utilizando mLab para MongoDB y Heroku para la web, de forma que se pasa por todas las fases de desarrollo en cuanto a programación web se refiere, es decir, diseño, implementación y despliegue.

Palabras clave: Aplicación web, comercio electrónico, gestión de eventos, pasarela de pago.

ABSTRACT

The realization of the present project supposes, on the one hand, to put into practice the knowledge acquired throughout the Degree, as well as, as far as possible, to try to explore and add new knowledge in the field of web application development that, moreover, integrates with a real payment gateway.

The starting point of this project arises from the need to create a website that automates the management and sales for the "Pepe Carroll's Memorial. As a final result, a web application that supports the Asociación Mágica Aragonesa for the management of said event is obtained, with the added value that, with small modifications, it could easily be adapted to any type of cultural activity in which buying tickets and other products related to the event could be done.

After the two editions of Pepe Carroll's memorial, a problem arose and was that systems that were used for the management and sale of tickets didn't allow the modification of that tickets, assigning a purchase by modification. This made it very chaotic for the event administrator and the user. For this reason a web application that allows the management and automation of entries is provided.

For the solution it has been decided to use the MEAN technology, which is based on MongoDB, ExpressJS, AngularJS and NodeJS, since it allows a fullStack implementation from analysis to deployment. Finally, being a web application, it has had to be deployed in production, using mLab for MongoDB and Heroku for the web, so that it goes through all the development phases in terms of web programming, that is, design, implementation and deployment.

Key words: Web application, e-commerce, event manager, payment gateway.

AGRADECIMIENTOS

Esta es una parte que tenía que muchas ganas de hacer. Son 7 años, 7 años de un amor odio brutal. 7 años de no dormir o dormir poco, 7 años en los que he tenido que renunciar a muchas cosas y a otras que no he renunciado ni renunciaré nunca. 7 años de grandes amigos, traiciones y falsedades. 7 años se dice pronto pero no se hacen pronto, se han hecho eternos. Pero.... Al final se sale, siempre acaba amaneciendo. Yo creía que no se salía de esto, había momentos que parecían un infierno y otros era el infierno. Pero se sale. Tras muchos disgustos dados a mis padres, hermana y amigos, ahora es el momento de decir que... **“¡Ahora sí que sí, ahora sí que sí!”**.

Por eso he de agradecer a toda esa gente que ha estado, está y estará. También a todos esos que estuvieron y se fueron, a esos que se aprovecharon. Por eso he de decir **GRACIAS** a todos. Gracias a los que me apoyasteis aun en mis peores momentos (recuperando 6 en septiembre llevando 8). Gracias a esos que me decían que no podría con esta aventura, y me comparaban con el tonto del pueblo. Gracias a todos esos que me daban un año de vida en esta aventura. Gracias a esos profesores de instituto que *“no se querían hacer responsables de mi equivocación”*. Que tengan una cosa clara, a pitera no me gana ni Dios. Así que con estos agradecimientos quiero dar las gracias a todas esas personas que no creyeron que podría. Porque gracias a ellos ahora soy lo que soy. Pero GRACIAS a los que realmente se lo merecen y sin los cuales no podría estar hoy aquí dando mal, GRACIAS a esa gente que me ha aguantado soportado y ha tenido la paciencia de aguantar y soportar mi bipolaridad. Solo decir gracias y que aquí me tenéis para lo que necesitéis.

GRACIAS a mi tutor Javier Fabra por confiar en mí y apoyarme en este proyecto y darme ánimos cada vez que lo he necesitado.

GRACIAS a mis padres, Antonio Peralta y Mari Mar Martínez, por aguantar y soportar tanto y lo más importante, confiar en mí y apoyarme económicamente pese al mal momento que vivimos en primer año. ESTO VA POR VOSOTROS.

GRACIAS a mi hermana, Cristina Peralta, por estar siempre allí y por tener que aguantarme en casa, la convivencia ha sido dura y más cuando estoy con nervios. Pero ahora todo ha terminado y empieza una nueva etapa en la que me toca cuidarte como me has cuidado durante todos estos 7 años.

GRACIAS a Alexis, por confiar en mí ciegamente y apoyarme siempre, gracias de corazón.

GRACIAS a mi gran amigo Javier Bielsa, él se ha ganado el cielo, no solo por estar cuando lo necesitaba, que siempre ha estado, sino por ser ese corrector tan paciente y soportar mis patadas al diccionario. Sin él esta memoria no sería lo que es, así que gracias por ayudarme a ordenar mis pensamientos y plasmarlos en esta memoria. GRACIAS POR TODO. Una mención especial para mi amigo Álvaro Barrios, el cual es tan culpable como Javi de que esta memoria esté como está, gracias a ambos, porque sin vosotros nunca hubiera podido terminar la memoria en el plazo.

GRACIAS a todas esas personas y amigos, que han estado cuando les he necesitado y me han ayudado en todo, podría nombraros, pero me dejaría alguno y al igual que no me gusta que se olviden de mí no me olvidaré de vosotros. Este trocito es para todos vosotros, vosotros sabréis quienes soy. Gracias de corazón.

No quiero despedirme sin nombrar a las personas más importantes, **mis abuelos**. Me habéis visto crecer y algunos no veréis el final, pero como os dije en múltiples ocasiones: “Acabaré porque a cojones no me gana ni dios”, y así es, aquí estoy. En recuerdo a mis abuelos que hoy no están, pero estarán orgullosos de mí estén donde estén, que, seguro que estarán muy cerca de mí, yo los llevo en mi corazón. Antonio Peralta, Gregorio Martínez, Mari Carmen Almazán, es por vosotros. Y a ti que hoy sí que lo podrás celebrar conmigo, Mercedes Casaña, ya soy Ingeniero.

GRACIAS a ese grupo de personas que conocí en Teruel, con los que hemos pasado tan buenos momentos, pensabais que sería el último en acabar el Grado, pero no ha sido así, gracias chicos por estar allí.

Y como no, agradecer a esa droga que descubrí en Teruel, en mis inicios del grado, con tan solo 18 años, me fui de casa y descubrí esa droga, que se ha metido en mí y nunca podré eliminar, has sido la culpable de que esto haya durado 7 años, pero también has sido la culpable de que no me viniera abajo nunca y de quitarme el estrés en los momentos difíciles.

GRACIAS POR TODO LO QUE ME HAS DADO, GRACIAS M-A-G-I-A. Y a mi maestro en el mundo de la magia Manolo Tena.

Gracias a todos, esta ingeniería es gracias a vosotros, que sepáis que un trocito de esto es vuestro, así que podéis decir que sois un trocito de ingenieros.

ÍNDICE

1. Introducción	1
1.1. Contexto y motivación	2
1.2. Objetivos	3
2. Estado del Arte	5
3. Diseño del sistema	9
3.1. Arquitectura Software.....	9
3.2. Implementación	11
3.2.1. Modelo de datos	11
3.2.2. Back-end.....	16
3.2.3. Front-end.....	18
3.2.4. Pasarela de pago	19
3.3. Prototipos de la interfaz de usuario.....	22
3.3.1. Mapa de navegación	23
3.3.2. Prototipos.....	29
3.4. Despliegue	41
4. Costes y planificaciones.....	43
5. Mejoras Futuras	44
6. Conclusiones.....	45
7. Bibliografía	46

ÍNDICE DE FIGURAS

FIGURA 1. Logotipo del Memorial Pepe Carroll	1
FIGURA 2. Logotipo de SELLA	5
FIGURA 3. Logotipo de IACPOS.....	5
FIGURA 4. Logotipo de EVENBOOST.....	6
FIGURA 5. Logotipo de HOOH.....	6
FIGURA 6. Logotipo de GESTIONENTRADAS.COM.....	7
FIGURA 7. Logotipo de TICKETMASTER	7
FIGURA 8. Logotipo de TICKETEA.....	8
FIGURA 9. Logotipo de TICKET TAILOR	8
FIGURA 10. Arquitectura de la solución.....	9
FIGURA 11. Arquitectura de las tecnologías utilizadas.	10
FIGURA 12. Modelos de datos de la aplicación.....	11
FIGURA 13. Estructura del proyecto	16
FIGURA 14. Dashboard de Stripe	20
FIGURA 15. API de Stripe.....	20
FIGURA 16. Mapa de navegación de Registro, Login y Compra.....	23
FIGURA 17. Mapa de navegación de "Mis compras"	24
FIGURA 18. Mapa de navegación de Perfil, Contacto y Salir	25
FIGURA 19. Mapa navegación lado Administrador principal.....	26
FIGURA 20. Mapa de navegación lado Administrador - Usuario	27
FIGURA 21. Mapa de navegación de Eventos	28
FIGURA 22. Registro de usuarios.....	29
FIGURA 23. Login y recuperación de contraseña.....	29
FIGURA 24. Barra superior del Administrador	30
FIGURA 25. Vista principal del Administrador.....	30
FIGURA 26. Filtro de fechas.....	30
FIGURA 27. Lista de todos los usuarios del sistema.....	30
FIGURA 28. Vista de los usuarios inscritos en el evento	31
FIGURA 29. Detalle sobre un usuario concreto.....	31
FIGURA 30. Detalle de la compra realizada.....	31
FIGURA 31. Histórico de compras	32
FIGURA 32. Vista de los eventos	32
FIGURA 33. Creación de un evento principal	33
FIGURA 34. Vista del evento principal.....	33
FIGURA 35. Modificación de un evento	34
FIGURA 36. Configuración de la acreditación	34
FIGURA 37. Creación de un evento dentro de un evento principal.....	35
FIGURA 38. Resumen de un evento	35
FIGURA 39. Modificación de un evento	36
FIGURA 40. Modal para enviar notificaciones	36
FIGURA 41. Barra superior del usuario estándar	37
FIGURA 42. Vista principal del usuario estándar.....	37
FIGURA 43. Compra de acreditación.....	37
FIGURA 44. Compra de acreditación como concursante.....	38
FIGURA 45. Vista de "Mis compras".....	38
FIGURA 46. Vista del detalle de una compra	39

FIGURA 47. Modal de confirmación de compra y modificación	39
FIGURA 48. Vista del perfil de usuario	40
FIGURA 49. Modal de confirmación de baja de usuario	40
FIGURA 50. Modal de cambio de contraseña	40
FIGURA 51. Pestaña de contacto.....	40
FIGURA 52. Arquitectura del despliegue.....	41
FIGURA 53. Arquitectura del despliegue de Heroku	42

ÍNDICE DE TABLAS

TABLA 1. Pasarelas de pago estudiadas.....	19
TABLA 2. Planificación final.....	43

GLOSARIO

- AMA** Asociación Mágica Aragonesa
- API** Application Programming Interface (en castellano Interfaz de Programación de Aplicaciones)
- AWS** Amazon Web Services
- HTML** HyperText Markup Language (en castellano Lenguaje de Marcas de Hipertexto)
- JS** JavaScript
- JSON** JavaScript Object Notation (en castellano Notación de Objetos JavaScript)
- MEAN** Acrónimo para MongoDB, ExpressJS, AngularJS y NodeJS
- MVC** Modelo – Vista - Controlador
- NPM** Node Packages Management (en castellano Gestor de Paquetes de Node)
- SEI** Sociedad Española de Ilusionismo

1. INTRODUCCIÓN

El Memorial Pepe Carroll se hace en honor al mago bilbilitano José Arsenio Franco Larraz, más conocido como Pepe Carroll, por su faceta de mago y presentador de televisión [1]. Pepe Carroll se formó en la magia en Zaragoza, más concretamente en Asociación Mágica Aragonesa (en adelante AMA), que es una de las sociedades mágicas más antiguas de España [2], por detrás de la SEI de Barcelona (Sociedad Española de Ilusionismo) [3]. El memorial se realiza a finales de enero por la proximidad con la fecha de su fallecimiento, el 5 de enero del 2004. Tras 11 años, el AMA quiso rendirle un homenaje con un memorial, pensado plenamente para magos con alguna actividad para público no mago [4]. Dicho evento es un congreso de magia que se realiza anualmente en la ciudad de Zaragoza. Es un evento que cuenta con multitud de actividades tanto para público mago como para público no mago. El AMA es la encargada de planificar y gestionar todo el Congreso, contando con terceras empresas para llevar a cabo la venta de las inscripciones de los congresistas magos y de las entradas para las actividades para público no mago.



FIGURA 1. Logotipo del Memorial Pepe Carroll

Los procesos más complejos derivados de la gestión de este evento están relacionados con la parte de las inscripciones de los congresistas, debido a la existencia de multitud de actividades opcionales. A lo largo de las dos primeras ediciones (2017 y 2018), todos los actos del Memorial se concentraron en 4 días (de jueves a domingo), teniendo tan sólo el sábado y el domingo para las actividades destinadas a magos inscritos. En la tercera edición, en 2019, está previsto que se desarrollen actividades durante toda la semana.

El principal problema que se tenía con los sistemas de ventas era la flexibilidad. Estos sistemas no son flexibles, es decir, cuando se compraba una entrada, ésta ya no podía ser modificada. Por ese motivo se tiene que realizar una nueva compra. Esto es muy molesto para los organizadores del evento, al igual que para el usuario, que ahora cuenta con dos o más compras en lugar de tener una única, mientras que los organizadores tenían múltiples compras de un único usuario, lo que hacía molesto el tener que localizar la compra de ese usuario. Además de todo lo anterior, estos sistemas suponen un gran coste para una asociación como el AMA, ya sea por la suscripción o por las elevadas comisiones cobradas por las pasarelas de pago.

Por ese motivo se propuso realizar un Trabajo Fin de Grado (en adelante TFG) para esta asociación, con un sistema que les permitiera vender las entradas para el Memorial Pepe Carroll y que estas pudieran ser modificadas por los usuarios, de forma que se efectuara una compra por parte del usuario, dado que facilitaría la labor a todas las partes implicadas. Respecto a la parte de la organización, se quería una gestión de las acreditaciones vendidas para tener valores concretos de comidas, cenas y camisetas vendidas para realizar los pedidos y reservar con la suficiente

antelación, además de poder facilitar la gestión de los concursantes, indicando un número máximo y sabiendo un breve resumen de su número, permitiendo además realizar avisos rápidamente, sin tener que modificar las listas de correos manualmente, es decir, se buscaba un poco la automatización de tareas triviales de la organización. Además, genera unas gráficas que ayudan al administrador del evento a conocer en todo momento el estado, de forma que haciendo uso de las gráficas se puede ver cómo van los acreditados y el número de acreditados por día.

A lo largo de esta memoria se expondrán distintos capítulos que describen el funcionamiento de la aplicación, así como otras cuestiones de interés. En una primera fase se realiza un análisis del problema, haciendo un estudio del estado del arte y haciendo una explicación de las distintas tecnologías utilizadas en este proyecto. El estado del arte pretende indagar sobre las distintas aplicaciones que puedan existir en el mercado y que realicen las mismas funciones que se hayan desarrollado en este proyecto.

En cuanto al diseño de la aplicación, en el capítulo relativo a la arquitectura y su funcionamiento, se explican con detalle los distintos aspectos técnicos de los procesos con mayor relevancia en la página, tales como el proceso de compra o la gestión de los distintos usuarios.

Se continúa hablando de las distintas tecnologías que han sido utilizadas, presentando una breve descripción y el papel que desempeñan en el proyecto. Para terminar, la última parte comenta las mejoras y las conclusiones obtenidas al realizar este TFG. También se referencia la bibliografía utilizada y se aportan anexos con información complementaria.

1.1. Contexto y motivación

Este proyecto surge del AMA por la necesidad de una aplicación para poder gestionar todos los eventos, y poder llevar un control de asistencia en tiempo real de las entradas vendidas, la gente que hay disfrutando de la actividad. Tras varias experiencias previas, el AMA ha detectado que no existe una solución en el mercado adaptada a este contexto, teniendo que recurrir a distintas soluciones de distintos proveedores, con la problemática que supone el que no esté todo integrado en un mismo entorno. Esto deriva en que muchas tareas que son automatizables no pueden automatizarse al no tener una aplicación donde se centralice toda la gestión.

Años atrás las inscripciones de los congresistas se gestionaban a través de *Google Forms* [5], donde las limitaciones eran bastante grandes. Además, este sistema hacía más complejo el cruce de los pagos con las inscripciones, ya que había que verificarlos de forma manual. Tras el primer año de memorial se decidió abandonar esta solución y fue cuando surgió el problema actual. Esta solución sólo servía para obtener la información de los acreditados y saber la información referente a la acreditación. Según los campos que rellenaban se les indicaba el precio, que debía ser abonado mediante transferencia y luego enviar el resguardo, lo que hacía un proceso lento y laborioso, ya que hasta que no se recibía el resguardo del pago no se creaba la acreditación y en muchos casos había que ir recordando a la gente que realizara dicho paso o se le anularía el formulario de inscripción. Por este motivo, el segundo año se buscó la opción de usar un sistema informático que realizara la venta de las acreditaciones y a la vez confirmara el pago.

En la segunda edición se optó por automatizar todo el proceso de inscripción a través de un sistema de venta de entradas donde cada inscrito era una compra y el sistema se encargaba de realizar el cobro. En la segunda edición se usó el sistema *Tickettailor* [6], ya que era el que más se ajustaba a las necesidades, pero aun así no servía porque tenía un problema con la flexibilidad, y es que solucionaba el ocasionado con *GoogleForms* pero creaba uno nuevo. La flexibilidad no permitía modificar las compras, lo que hizo que muchos usuarios avisaran de que querían

modificar su acreditación y se contara con esa nueva modificación y lo pagaran en efectivo el día del memorial en el momento de recoger las acreditaciones. Esta eventualidad hizo que en ese momento se formaran muchas colas de espera y la gente se impacientara debido a que era necesario comprobar el estado actual de cada usuario.

Otro problema surgía al tener que leer el código QR de las entradas adquiridas a través de *Tickettailor*, puesto que este sistema generaba obligatoriamente un código. Posteriormente, había que leer este código QR y luego confirmar en el papel que ese código era válido o que tenía que abonar o darle el importe de la nueva modificación. Por esos motivos se busca una aplicación que englobe todas las compras de un usuario en una única, de forma que se facilite la confirmación y de esa forma, evitar el problema de trabajar con dinero en efectivo el mismo día del evento.

Resumiendo, se busca un sistema que resuelva las necesidades nombradas anteriormente, ya que no hay un sistema en el mercado que se ajuste a estas necesidades, sabiendo que se es una asociación sin ánimo de lucro y que dicha asociación no dispone de capital.

1.2. Objetivos

Entre otras, se ha escogido la temática que comprende los espectáculos de magia por la familiaridad y relación que se tiene en el AMA, de ahí que haya sido mucho más sencilla la realización del proyecto, ya que los requisitos y lo que el “cliente”, por así decirlo, quiere ya ha sido pensado desde hace tiempo, tal y como se ha comentado en el contexto.

Los objetivos del proyecto son:

- Realizar una aplicación web que dé solución a las necesidades del AMA (siendo además fácilmente reutilizable para otro tipo de eventos y espectáculos).
- Diseñar e implementar un proyecto web que permita gestionar la venta de entradas para el “Memorial Pepe Carroll”, siendo un sistema flexible con las acreditaciones en el sentido en que permite modificarlas y que sean actualizadas en lo que se refiere al coste mediante una pasarela de pago que se encargará de realizar las transacciones necesarias según las acreditaciones y el usuario para tener en todo momento las cuentas actualizadas, de forma que se pueda acceder desde cualquier parte, ya que esta aplicación será utilizada en las actividades de los eventos.
- Aprender y afianzar los conocimientos relacionados con el desarrollo web.
- Identificar cada uno de los procesos que acontecen desde que el usuario se registra hasta que se realiza el pago.

Además de los objetivos principales, existen otros que se especifican a un nivel de detalle más técnico:

- Crear un proyecto desde cero, empezando por el diseño, continuando con la implementación con tecnologías novedosas y finalizando con un despliegue real, pensando que la aplicación ha de desplegarse en un entorno de producción real, lista para ser utilizada en el próximo memorial.

- Desarrollar una interfaz de usuario lo más usable, sencilla e intuitiva que facilite el uso de la aplicación por parte de los usuarios, de forma que sea cómoda y fácil.
- Proporcionar un entorno de implantación del sistema y funcionamiento de fácil mantenimiento y bajo coste económico.
- Realizar una aplicación basada en tecnología de servicios, con tecnología muy novedosa.
- Realizar una aplicación utilizando bases de datos no relacionales, con un modelo de datos apropiado a la aplicación y facilitando la modificación de las acreditaciones.

2. ESTADO DEL ARTE

El primer paso para llevar a cabo este proyecto ha sido la investigación y análisis de la información sobre aplicaciones similares a la desarrollada en este trabajo. De esta manera se puede entender el funcionamiento de éstas y basarse en ellas para llevar el proyecto a buen puerto y que tenga éxito.

Asimismo, se pueden conocer las diferencias entre ellas, de tal manera que se tenga una ventaja competitiva y saber aprovechar esa diferencia. A continuación, se nombran algunas de las más relevantes, ya que hay muchas aplicaciones web de este tipo y lo importante es ver si éstas podrían servir para el objetivo de este proyecto.

SELLA



FIGURA 2. Logotipo de SELLA

El software SELLA de Venta de Entradas y Control de Accesos [7] es un sistema de escritorio creado para gestionar e integrar informáticamente las necesidades de una empresa relacionadas con la venta de entradas, la gestión de socios y el control de accesos en eventos donde sea requerido. Para ello la aplicación se complementa gestionando aforos, documentos, formatos a imprimir, control de caja, perfiles y taquillas.

Esta aplicación permite gestionar los usuarios y perfiles, así como las personas y tipos de personas que acceden a los eventos. Por otro lado, se pueden crear eventos y gestionar el aforo que acudirá a ellos. Además, realiza la venta y la gestión de las entradas para dichos eventos con el valor añadido de la realización de informes con distintas estadísticas sobre todos los sujetos implicados (eventos, personas, aforo...). Finalmente, la aplicación permite tramitar documentos, tales como la impresión de las entradas en el formato que se defina.

Esta aplicación por sus características se ajusta mucho a las necesidades de la asociación, pero hay una característica que la echa para atrás, ya que se busca una aplicación que se pueda usar desde distintos dispositivos y desde móviles, sobretodo, puesto que no siempre se va a disponer de un PC para la gestión. Por eso, este motivo ha sido una de las claves por las que no se adapta a las necesidades del Memorial Pepe Carroll.

IACPOS



FIGURA 3. Logotipo de IACPOS

IACPOS [8] ofrece una amplia variedad de soluciones integradas que proporcionarán herramientas para la gestión de museos, venta online para parques de atracciones *fastpass* o abonos máquinas automáticas. Asimismo, ofrece soluciones al servicio de la sala de cine con diferentes posibilidades para la compra cómoda de entradas en taquilla o adelantando la venta, trabajando de forma online, lo que hace que sea accesible desde cualquier punto.

IACPOS encajaría a la perfección con los requisitos, solo que esta aplicación no es viable ya que es muy cara y hace que se pase del presupuesto de la asociación, a pesar de que se consiguió IACPOS por 50€ al mes más un porcentaje por acreditación vendida tras unas negociaciones con la

empresa que desarrolla este software. Aun así, la aplicación que se utilizó en la segunda edición, Tickettailor, solo costaba 20€ al mes activado, por lo que seguía teniendo menor coste para la asociación.

EVENBOOST



FIGURA 4. Logotipo de EVENBOOST

Eventboost [9] permite gestionar los distintos invitados de un evento, personalizar la página de los distintos espectáculos, generar acreditaciones para el evento, seguir el rendimiento, revisar las inscripciones online y las actualizaciones de asistencia, almacenando los datos de forma segura y encriptada. Permite la creación de venta de entradas en línea de eventos propios del usuario gracias a la asociación con el sistema de comercio electrónico Shopify [10] con la ventaja de que, si se dispone de una página web sobre el evento, ésta puede ser integrada con los sistemas de EVENBOOST.

Esta aplicación también podría encajar en los requisitos de la aplicación, pero hay un factor que la hace inviable y es la pasarela de pago, ya que si no se hace uso de la propia del sistema, el coste se eleva mucho.

HOOH



FIGURA 5. Logotipo de HOOH

HooH [11] es una aplicación creada en entorno web que permite a la sala la gestión de eventos o espectáculos, reservas, abonos, usuarios, la venta online y la venta en taquilla. Está pensada para ser utilizada en un entorno multiusuario, lo que permite a los distintos administradores gestionar la sala desde su puesto de trabajo o desde cualquier otro ordenador o dispositivo móvil con acceso a internet.

Esta aplicación web se compone de tres partes diferentes gestionadas desde un mismo entorno: la web pública, la taquilla y la parte de administración. En la primera, el usuario podrá ver todos los espectáculos que la sala tiene reservados, reservar cualquier evento o realizar la compra de entradas, eligiendo el asiento. La segunda parte permitirá toda la gestión relacionada con la taquilla, tal como la impresión de las entradas, la aplicación de descuentos, etc. Por último, la tercera parte dejará administrar usuarios, gestión de abonos, informes y otras funciones administrativas.

Este sistema es uno de los que más se aproxima a las necesidades de la aplicación, pero el problema es el presupuesto, lo que hace que la asociación no se la pueda permitir. Tiene una versión, pero entonces no ofrece todas las funcionalidades que se necesitan.

GESTIONENTRADAS.COM

The logo for 'Gestion Entradas' features the word 'Gestion' in a bold, black, sans-serif font. The letter 'o' in 'Gestion' is replaced by a blue power button symbol. Below 'Gestion' is the word 'e-ntradas' in a smaller, black, sans-serif font, with a hyphen between 'e' and 'ntradas'.

Gestión Entradas [12] es una aplicación web que permite realizar la gestión de las entradas, así como su venta a través de la red. Esta aplicación tiene como punto fuerte la utilización de los denominados teléfonos inteligentes para leer los códigos QR en las invitaciones y entradas de los distintos usuarios.

FIGURA 6. Logotipo de
GESTIONENTRADAS.COM

Detrás de la aplicación se encuentra una empresa que ofrece la gestión de las entradas, de forma que, por cada entrada vendida, a dicha empresa se le paga una cantidad por entrada. Como empresa, cuantas más entradas gestionen, mejor precio por entrada cobrará, permitiendo además el uso de diferentes pasarelas de pago.

El principal problema que surge con esta aplicación es la cantidad de entradas a gestionar, ya que se cuenta con un número muy limitado de entradas y, al ser así, el mero hecho de utilizar este sistema hace que se encarezca el coste de la gestión como en casos anteriores.

TICKETMASTET

The logo for Ticketmaster features the word 'ticketmaster' in a lowercase, italicized, black, sans-serif font. A registered trademark symbol (®) is located to the upper right of the word.

FIGURA 7. Logotipo de
TICKETMASTER

Ticketmaster [13] es una empresa y aplicación con el mismo nombre, desarrollada para la venta y distribución de entradas, especialmente arenas, teatros y conciertos, con la particularidad de que Ticketmaster es un agente que vende únicamente las entradas que sus clientes ponen a disposición.

Se trata de la empresa líder global en la venta de entradas para eventos y uno de los primeros sitios de comercio electrónico especializado en este tipo de menesteres. Haciendo una comparativa con la aplicación web desarrollada en este proyecto, Ticketmaster permite realizar la comprobación de que la compra se ha hecho correctamente, así como elegir dónde y cómo desea el cliente recoger las entradas. También ofrece un seguro de no asistencia en caso de que el cliente no sepa si puede asistir al evento comprado y un servicio de venta de entradas para personas en silla de ruedas. En relación al proyecto, tiene una aplicación interna para poder gestionar el perfil del usuario y modificar las entradas si así se desea.

Es una de las soluciones que más se adapta a los requisitos. Además, es la más importante del mundo de la venta de entradas, pero hay un problema, y es que las altas comisiones que hay que pagar hacen que sea totalmente inviable esta opción porque la comisión llega a alcanzar los 10€ en algunas entradas.

TICKETEA



FIGURA 8. Logotipo de TICKETEA

Ticketea [14] se trata de una solución tecnológica que simplifica la forma en que se organizan y gestionan los eventos online y se venden entradas. Permite la venta y distribución de pases para multitud de eventos, entre ellos conciertos, festivales, ballet u opera.

Asimismo, permite la creación de eventos a usuarios registrados en la aplicación de forma que éstos tienen que elegir el tipo de entrada, el recinto, la fecha de celebración y el precio, así como un panel de control de cada evento para la consulta de la evolución, por ejemplo, ganancias, promociones o el estado del evento.

Este sistema está muy bien para las necesidades que se están buscando, pero no permite realizar las modificaciones de acreditaciones que es necesario realizar según los objetivos.

TICKET TAILOR



FIGURA 9. Logotipo de TICKET TAILOR

Ticket Tailor [6] permite la creación de entradas para cualquier tipo de evento y su posterior venta. Para contratar este servicio, es necesario desembolsar una cantidad mensual para poder utilizar la aplicación, aunque sólo se paga por los meses que el evento esté activado. Asimismo, permite el uso de las pasarelas de

pago Stripe y Paypal y la visualización de gráficas para el análisis del evento creado e informes de todo tipo con datos estadísticos que permitan evaluar los resultados. En contrapartida, al igual que Ticketmaster, no permite la modificación de las acreditaciones, por lo que, aunque se adaptaba en su mayoría a lo buscado y siendo lo utilizado en el segundo año del memorial, se ha desechado.

Todas estas opciones han sido descartadas por dos motivos principales. El más importante es que no tienen la flexibilidad que demandaban para este evento, que es poder modificar compras. El otro motivo es que todas estas opciones suponían realizar un desembolso y al ser una asociación sin ánimo de lucro no podían permitirse el lujo de realizarlo, ya que el memorial que organizan no supera las cien personas entre participantes, invitados y miembros de la organización.

3. DISEÑO DEL SISTEMA

En esta sección se va a detallar el diseño de la solución propuesta para este proyecto.

3.1. Arquitectura Software

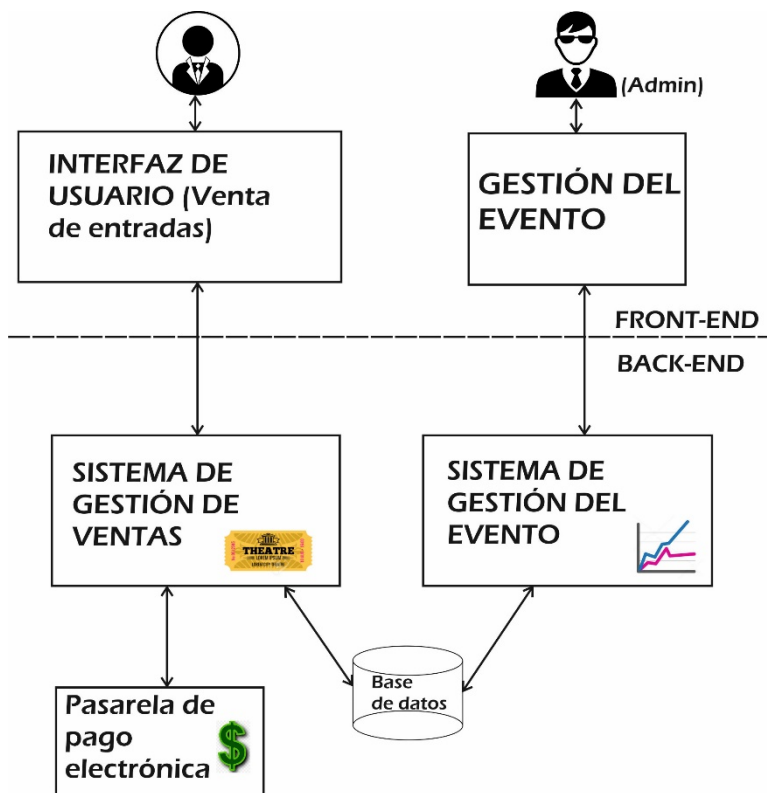


FIGURA 10. Arquitectura de la solución

y modificarlas, interactuando con la pasarela de pago. Por otro lado, el sistema de gestión de eventos, que es el que permite gestionar y controlar el evento ofreciendo resultados, como por ejemplo las gráficas. La tercera y última parte es la base de datos, con la que interactúan las dos partes anteriores.

En la Figura 10 se muestra la arquitectura de alto nivel para hacerse una idea del funcionamiento de la aplicación web. Los sistemas de gestión, tanto los de las ventas como los de los eventos, está desplegados en Heroku y la base de datos está en mLab. En la Figura 11 se puede ver en detalle la arquitectura de las tecnologías utilizadas.

A continuación, se va a ver más en detalle a nivel de la tecnología usada en la aplicación web. Para entenderlo mejor, ésta se divide en 3 partes: Cliente, Servidor Web y Servidor de datos. Esta solución se basa en dos partes principales, la parte del front-end (el cliente) y la parte del back-end (el servidor de datos y el servidor Web), de forma que, bidireccionalmente, todas se intercambian los datos y la información mediante JSON [16]. Asimismo, se muestra también, en la Figura 11 (Figura original [17]) el resultado de la arquitectura terminada con el despliegue, la del servidor web en Heroku [17] y la base de datos, MongoDB [18], en mLab [19].

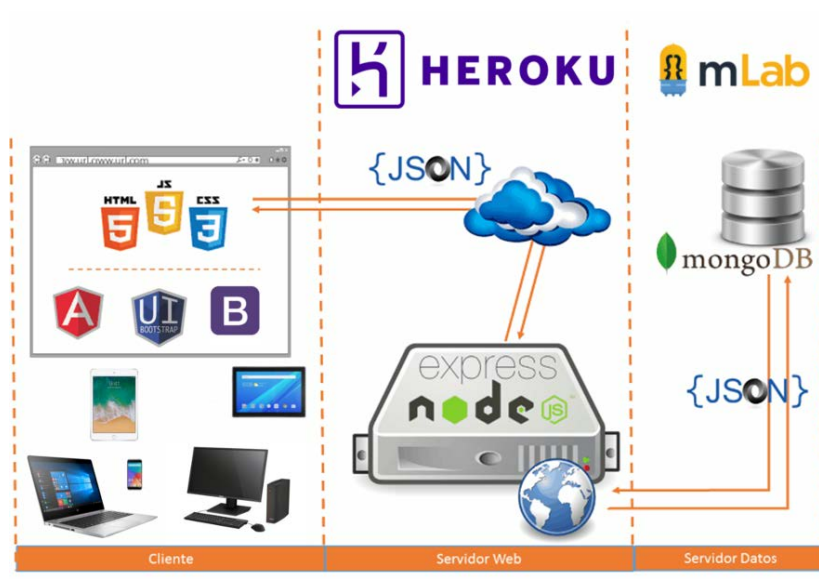


FIGURA 11. Arquitectura de las tecnologías utilizadas.

El front-end es la parte del cliente, es decir, la que hace uso de los prototipos del punto superior, de forma que, la aplicación web pueda ser utilizada por todo tipo de dispositivos, mientras que el back-end es la parte del servidor, es decir, la que se encarga de gestionar los prototipos mediante micro servicios y diversas conexiones a la Base de datos. Ambas partes se conectan mediante una base de datos NoSQL, utilizándose MongoDB, entre otras razones porque posee una serie de colecciones para gestionar la información y porque está pensada para la programación MEAN [20], utilizando asimismo NodeJS [21].

Mediante este tipo de tecnología, se puede adaptar la solución a todos los dispositivos del mercado, puesto que al ser Cliente/Servidor, lo único que habría que adaptar sería el cliente, es decir, el front-end a los distintos dispositivos. Para la realización de la aplicación, se ha utilizado AngularJS [22] y Bootstrap [23], con el objetivo de intentar conseguir que las vistas estén disponibles para la mayor parte de dispositivos, desde PC y desktop a tablet y smartphones. Que la interfaz se adapte a la mayor parte de dispositivos se debe a que la tecnología usada para el front-end es *responsive*, es decir, se adapta a las medidas del dispositivo que la esté invocando, de forma que dicha interfaz se auto-escala. La ventaja de estas tecnologías es que sólo es necesaria la conexión a Internet para hacer uso de ellas.

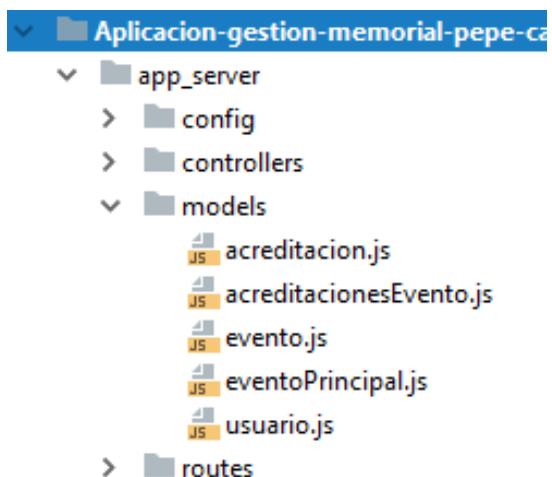
La conexión con la base de datos se consigue mediante los modelos de datos, los cuales se encargan de gestionar y ordenar la información con la que se va a trabajar. Esa información se envía del back-end al front-end de forma bidireccional para que se guarden los datos en BD, consiguiendo almacenar y trabajar con la información. Dicha comunicación interna es gestionada por el back-end, tal y como se explica anteriormente, de forma que interactúan todas las partes implicadas en el funcionamiento de la aplicación web, es decir, la base de datos, la pasarela de pago y las vistas del cliente.

3.2. Implementación

A continuación, se van a detallar los aspectos principales de la implementación del sistema.

3.2.1. Modelo de datos

El modelado de los datos de este Trabajo Fin de Grado es bastante importante para poder almacenar los datos de forma estructurada para que su posterior recuperación sea lo más eficiente posible.



Toda esta información se gestiona desde *NodeJS* con la librería “*mongoose*” [24]. Esta librería está desarrollada para trabajar con MongoDB, de forma que se facilita enormemente el uso de la base de datos. Se trata de un proceso tan sencillo como pedirle el dato y los parámetros a través de los que se desee realizar la búsqueda y ésta devuelve todas las filas que tenga el parámetro solicitado. Además, editar un dato es muy sencillo, ya que se puede realizar una edición al vuelo, o algo más complejo, obteniendo el dato y trabajando con él para luego guardarlo.

FIGURA 12. Modelos de datos de la aplicación

Los modelos de datos son la clave en el trabajo con la información. Dicha información se almacena en colecciones de datos, que son las que indican cómo está estructurada la base de datos, ya que al tratarse de una Base de Datos NoSQL no hay un modelo relacional como en las bases de datos SQL. MongoDB sólo necesita un fichero que le indique los nombres de los atributos que se van a utilizar y el tipo de dato. Dichos datos son tipos primitivos, tales como String, Number, Boolean o Array entre otros.

Según MongoDB, cada modelo de datos es una *Colección*, que permite extender el modelo de datos añadiendo atributos a los documentos sin necesidad de refactorizar apenas el código. Por este motivo, en este proyecto se trabaja con cinco modelos que se detallan a continuación.

Modelo “Acreditación”

Este modelo de datos es el que se encarga de almacenar las acreditaciones, es decir, el que guarda la información de las credenciales compradas. Almacena la información del usuario que ha comprado, así como datos sobre la compra, siempre siguiendo la información de las acreditaciones creadas por el administrador de la web.

Estos datos tienen dos identificadores, el del evento y el del usuario, de forma que haciendo uso de los dos identificadores se obtiene la acreditación. Con dicha información se facilitan enormemente las búsquedas, lo que hace que buscar acreditaciones sea más sencillo.

```
fechaCompra: Date,  
fechaModificacion: Date,
```

```
fechaConcursante: Date,  
evento: String,  
id_usuario: String,  
id_compra: String,  
nombre_usuario:String,  
concursante: Boolean,  
tipo: String,  
descripcion: String,  
comida:Boolean,  
cena: Boolean,  
camiseta:Boolean,  
talla:String,  
precioBase: Number,  
precioComida: Number,  
precioCena: Number,  
precioCamiseta: Number,  
precioTotal: Number
```

Tal y como se puede comprobar, este modelo de datos tiene tres valores *Date*, "*fechaCompra*", "*fechaModificacion*" y "*fechaConcursante*". Estos datos son usados básicamente para la parte de estadística en la parte de las gráficas del Administrador, con el objetivo de llevar un registro de las actividades diarias, así como acreditaciones por día.

Después están los valores booleanos, los cuales permiten saber si el usuario ha contratado el extra indicado; en este caso podrían ser "*comida*", "*cena*" y/o "*camiseta*". También se guarda la talla de la camiseta para que todos los usuarios que la compraron tengan su talla correspondiente.

Además, se almacenan los precios individuales y el precio total para realizar las devoluciones si fuese necesario, con el objetivo de que si el precio se modificase se le devolviese la cantidad correspondiente. También se le indica a qué evento principal corresponde la acreditación para facilitar su búsqueda. Para terminar, se almacenan dos identificadores, el de usuario (para saber a qué usuario le corresponde la acreditación) y el de compra (para saber a qué compra se corresponde la acreditación). Respecto a la compra, Stripe tiene su propia Base de Datos de compras y para realizar cualquier modificación en ésta solicita el id de la compra. Por lo demás, los datos restantes corresponden a la información de la acreditación, así como datos del concurso.

Modelo "acreditacionesEventos"

Este modelo de datos es el que se encarga de almacenar las opciones de cada acreditación creada para cada evento. Cualquier variación o cambio en las opciones de las acreditaciones se almacena aquí, creándose una acreditación por evento, lo que nos facilita tener un control total de las acreditaciones.

```
fechaCreacion: Date,  
fechaModificacion: Date,  
evento: String,  
precioNoConcursante: Number,  
concursante: Boolean,  
precioConcursante: Number,  
concursantesFaltan: Number,  
totalConcursantes: Number,
```

```
especialidadesConcurso: [],
comida: Boolean,
precioComida: Number,
cena: Boolean,
precioCena: Number,
camiseta: Boolean,
precioCamiseta: Number,
tallas:[],
lista_usuarios: [],
lista_concursantes: [],
lista_noConcursantes: []
```

Dicho modelo contiene toda la información necesaria para la realización de las inscripciones. Contiene valores *booleanos* para indicar si un complemento está activo para el evento; esto se puede ver en “*concursante*”, “*comida*”, “*cena*” y “*camiseta*” y en los precios de cada complemento. Los valores que tienen múltiples opciones, como “*especialidadesConcurso*” y “*tallas*” se almacenan en un *Array de String* con los datos que hay para elegir.

Además, tiene 3 valores que son *Array*, “*lista_usuarios*”, que contiene el correo (*id_usuario*) de todos los inscritos al evento, “*lista_concursantes*”, que contiene el email (*id_usuario*) de todos los inscritos al evento como concursantes y “*lista_noConcursantes*”, que contiene el correo (*id_usuario*) de todos los inscritos al evento de forma estándar. Gracias a dichas listas se puede facilitar el envío de avisos a los inscritos al evento, ya que simplemente hay que elegir a la lista que se desea enviar.

Modelo “evento”

Este modelo de datos se encarga de almacenar los eventos que se crean y van conectados con el evento principal. Dicho modelo guarda la información de todos los eventos que se crean en la aplicación. Su principal dato es el evento principal al que pertenecen, lo que facilita las búsquedas, siendo el modelo de datos el siguiente:

```
fecha: Date,
fechaCreacion: Date,
fechaModificacion: Date,
eventoPrincipal: String,
id_eventoPrincipal: Number,
titulo: String,
descripcion: String,
lugar: String,
precio: Number,
incluido: Boolean,
```

Este modelo almacena la información principal del evento y dos valores para una futura estadística, que de momento no tiene funcionalidad alguna en la aplicación y que son “*fechaCreacion*” y “*fechaModificacion*”, puesto que en un futuro se podrían crear estadísticas para ver la creación de eventos y si éstos fueron modificados.

Respecto a los valores importantes, se almacena “*fecha*”, que es el valor que contiene una fecha completa, es decir, el día/mes/año y la hora del evento, de forma que en una variable se guardan dos datos importantes. Cuando el Front-end reciba este valor, sólo se mostrará la parte que interese, utilizando para ello AngularJS, que tiene un gran filtro de tipo Date. Respecto al resto de datos, se usa el id del evento principal para enlazar este evento con su evento principal, utilizando para ello el id del evento principal, porque al poder modificar todos los datos del evento, el id es el único valor que no se puede modificar, mientras que el nombre puede surgir algún cambio. Se utiliza también un Booleano para “*incluido*”, que indica si este evento está incluido en la inscripción; de no ser así (actualmente todos los eventos están incluidos), sería necesario pagar un suplemento. El resto de datos son el título del evento, la descripción y el lugar donde se realiza.

Modelo “eventoPrincipal”

Este modelo de datos es el que se encarga de almacenar los eventos principales, de forma que se utiliza para guardar toda la información de dichos eventos. Este modelo de datos se identifica con un id de evento. Se ha diseñado así porque al tener el nombre del evento la posibilidad de modificarse, se podrían perder las modificaciones. Tal y como se puede observar en el modelo, no se le indica el identificador. Esto se debe a que la base de datos, MongoDB la asigna automáticamente, siendo el modelo de datos el siguiente:

```
fechaCreacion:Date,  
fechaModificacion:Date,  
inicio:Date,  
final:Date,  
duracion:Number,  
nombre:String,  
descripcion:String,  
plazas:Number,  
total_inscritos:Number,  
imagen:String
```

El evento principal tiene dos valores *Date* para la creación de futuras estadísticas, al igual que con el modelo “*Evento*”. Respecto a las fechas del evento, como tiene dos, la de inicio y la de final, éstas se guardan en dos valores distintos. También almacena la información básica del evento como es el nombre, la descripción y el número de inscritos (desde el total que se permiten hasta el total de inscritos que hay en el momento actual), así como un String para la imagen.

Es importante mencionar que lo que realmente almacena la base de datos es la dirección de la imagen, es decir, la ruta donde se encuentra la imagen en el servidor web. De este modo la Base de datos se quita el peso de ofrecer una imagen, pasándole esta carga al servidor, que es el que guarda la imagen en una carpeta pública de éste.

Modelo “usuario”

Este modelo de datos es el que se encarga de almacenar la información de todos los usuarios del sistema. Cada vez que un usuario accede a la aplicación se consulta esta colección, en la cual está la información necesaria para hacer el proceso de registro (Login). Además, contiene un listado del historial de compras que ha realizado el usuario, de forma que se consigue que cada usuario tenga una única compra actual y se pueda consultar su historial. El modelo de datos es el siguiente:


```
fechaCreacion: Date,  
fechaModificacion: Date,  
correo: String,  
contrasena: String,  
nombre_artistico: String,  
nombre: String,  
apellidos: String,  
dni: String,  
telefono: Number,  
numero_compras: Number,  
compras: [],  
eventos:[],  
activo: Boolean,  
ultimoAcceso: Date
```

El modelo de datos de “usuarios” contiene dos listas importantes, la primera es una lista de todos los eventos a los que se ha inscrito “eventos”, mientras que la segunda es una lista de todas las compras que ha realizado con sus modificaciones “compras”; gracias a esta lista se puede realizar el historial de compras, puesto que guarda toda la información de una compra. De este modo, siempre podrá modificar la inscripción actual manteniendo el histórico. También se trabaja con un dato Booleano, “activo” para indicar si ese usuario está dado de baja o no. Se usa este método porque de esta forma queda almacenado el usuario para las estadísticas, pero si se intentara conectar, se mostraría un mensaje advirtiendo que no existe dicho usuario y al volver a autenticarse (*logearse*) se activaría de nuevo y mantendría todos sus datos.

Tal y como se puede observar, el resto de datos son los básicos que hay que almacenar para trabajar con el comercio electrónico. Respecto a “*ultimoAcceso*”, es el dato que se va usar para las estadísticas futuras, consiguiendo así ver una interacción de los usuarios con respecto a los eventos que se van sacando. La “contrasena” se guarda cifrada con la librería *bcrypt* [25], que se explicará más adelante.

Este modelo de datos tiene un identificador único que corresponde al correo electrónico del usuario, puesto que es un valor que a priori no cambia y facilita las búsquedas. Además, el modelo de datos cuenta con una pequeña parte para el *token* de sesión, el cual permite al usuario navegar por la web siempre y cuando su *token* siga vivo o no se cierre sesión destruyendo dicho *token*. En tal caso, el usuario tendría que autenticarse (*logearse*) creando un nuevo *token* de sesión, con el que tendría acceso a la web nuevamente. Este *token* también es el que permite el acceso al API del servidor y con él se tiene un control total en la web haciéndola mucho más segura. El citado token también se almacena en el modelo de datos del usuario. La razón es sencilla: se permite que, al cerrar el navegador, los datos sigan activos. También contiene información del *token* que se asigna al iniciar la sesión para poder navegar por la web. Dicha información ayuda a controlar el acceso a la API, que será tratada más adelante.

Es muy importante mencionar que los modelos de datos han sido pensados para que trabajar con ellos sea sencillo. Debido a esto, se han adaptado los datos a las necesidades de las búsquedas.

3.2.2. Back-end

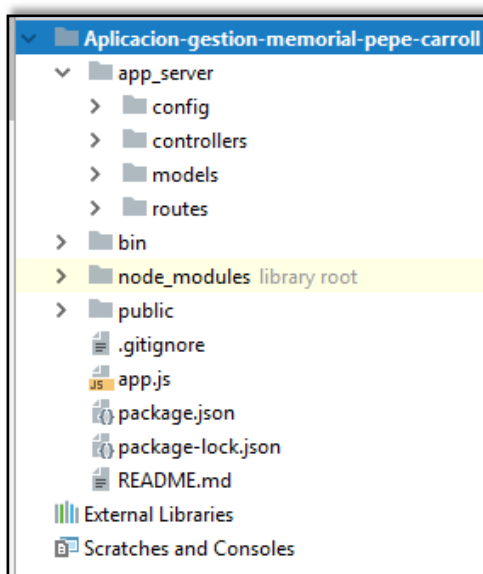


FIGURA 13. Estructura del proyecto

El Back-end, Figura 13, se ha implementado en *NodeJS* [21] es la tecnología sobre la que se va a cimentar este proyecto (proyecto *MEAN*). Se trata de una tecnología basada en *JavaScript*, la cual reduce el lenguaje JS a algo simple. Además, cuenta con sus propios repositorios, *npm* [26] es el repositorio de *NodeJS* con todas las librerías, que suelen ser oficiales y de gente anónima y que facilitan mucho su uso porque ofrecen API's a las que simplemente hay que enviarle los parámetros necesarios. Siguiendo por *Express* [27], es un framework pensado para los proyectos, es decir, su finalidad es la de configurar una aplicación web de tecnología JS.

Por ese motivo se utiliza *NodeJS*, puesto que está pensado para la programación del Servidor, mientras que *Express* está pensado para la configuración de la aplicación Servidor por un lado y Cliente por el otro. La configuración del Servidor se realiza en un fichero, que será el encargado de hacer que la aplicación se lance y configurar todo lo necesario, como son las dependencias y rutas de ficheros y direcciones, facilitando la tarea mediante el uso de librerías.

Para su uso, *Express* ofrece un fichero denominado "package.json", en el cual se le indica la configuración básica del proyecto, con las dependencias necesarias y sus versiones necesarias, pudiéndose indicar una versión concreta, la versión más nueva o la versión más nueva de una concreta.

La estructura del proyecto es el de "Modelo-Vista-Controlador (MVC)". *Express* crea una configuración predefinida distinta y se ha tenido que modificar dicha estructura para adaptarlo mediante dos carpetas principales "app_server" y "public" (ésta última es donde se encuentra la parte del front-end, las vistas y sus controladores). Dentro de la carpeta "app_server", que es la carpeta de la parte del Back-end, se pueden encontrar las tres carpetas básicas "controllers", "models" y "routes".

En la ruta "controllers" es donde se encuentran los micro-servicios de los que va a hacer uso la aplicación, es decir, aquí es donde están las funciones que serán llamadas para realizar las distintas tareas de la aplicación. O lo que es lo mismo, es la parte del servidor, es decir, desde aquí se hará uso de la base de datos, creación, lectura y modificación.

En la ruta "models" es donde se encuentran los modelos de la base de datos. Aquí está la estructura de la base de datos, como bien se ha explicado en los modelos de datos, mientras que "routes" contiene el API, es decir, donde están las direcciones con sus funciones. Aquí se encuentran todas las funciones públicas y la forma de invocarlas. Sin esta ruta y sin sus ficheros, nunca se podría acceder a los "controllers", de ahí que sea de suma importancia. Además, es el router interno que se comunica con *Express*. Estos ficheros del route son los que conectan las direcciones del API con las funciones, de forma que se facilita el uso de *Express*.

Además, se hace uso de *frameworks* que facilitan mucho el manejo de funciones del Back-end, tal y como se detalla a continuación.

NodeMailer

NodeMailer [28] es un *framework* para facilitar el envío de correos electrónicos. En el proyecto se usa este *framework* para enviar los e-mails de avisos con las listas de los inscritos, así como para enviar las contraseñas en caso de ser olvidadas por el usuario y como método de contacto con el administrador de la web.

Bcrypt

Bcrypt [25] es un *framework* que se usa para cifrar las contraseñas, dado que es necesario trabajar con la contraseña cifrada en todo momento. También posee una función que comprueba la contraseña introducida con la que ya está cifrada en la base de datos con el fin de hacer las comprobaciones pertinentes con el usuario.

Passport

Passport [29] es un *framework* que controla el acceso de usuarios, mediante el cual un usuario un usuario se autentica y se registra (*login*) y obtiene un *token*, el cual le va a permitir navegar por toda la aplicación. Una vez sale de sesión o caduca el *token* ya no puede navegar por la aplicación. Gracias a este *framework* se tiene controlado el acceso a zonas restringidas de la aplicación, teniendo que preocuparse simplemente porque el usuario se registre.

3.2.3. Front-end

El Front-end es la parte que se encarga del cliente. Se compone de código HTML [30] y hojas de estilo CSS [31], englobado en el *framework* Bootstrap. En este proyecto se ha utilizado la versión 3.3.7. El motivo por el que se usa esta versión y no la actual (v 4.0.0) es el *framework* usado para integrar AngularJS en los componentes no tiene compatibilidad para la última versión. También posee código en JavaScript para dar dinamismo a la vista, gracias a la tecnología AngularJS, que, aunque se trata de su primera versión, es una de las más usadas en la actualidad, entre otras razones porque es una de las tecnologías utilizadas en la mayoría de proyectos (no en nuevos, pero si para mantenimiento). Para la integración de AngularJS y Bootstrap se ha hecho uso del *framework* UiBootstrap [32], que hace uso de unas directivas de AngularJs predefinidas para que estos componentes posean un dinamismo, de forma que simplemente se tengan que configurar mediante la opción “ng-”. Posee múltiples opciones, desde las más básicas como “ng-show” o “ng-click” a las más complejas que sólo sirven para el componente en concreto.

La composición de las vistas consta de dos ficheros por vista. El código HTML tiene los componentes y el controladorJS, que es la parte de AngularJS, es la que le da el dinamismo a la vista y realiza las conexiones con el Back-end.

Los controladores tienen las dependencias que van a usar y una conexión directa con el *router*, que es el que se encarga de gestionar las vistas. Además, AngularJS posee un *router*, que es el que se encarga de redireccionar las vistas y controlar los ficheros que se han de cargar en cada punto. Para esto se ha usado “angular-ui-router” [33], que facilita el uso de multi página en Angular. Asimismo, se han usado varios *framework* para mejorar la vista del contenido del Front-end, detallado a continuación.

NgTable

NgTable [34] es un *framework* especializado en el trabajo con datos con tablas. Además de mostrar los datos organizados en columnas y filas, permite utilizar filtros y ordenar los datos. Estas tablas permiten filtrar y ordenar los datos resultantes de múltiples maneras, permitiendo configurar como se desee que se muestren los datos de forma predefinida y ofreciendo la opción de modificar esas características sobre la marcha en tiempo real. Gracias a este *framework*, se ha facilitado el trabajo con los datos de forma muy fácil y dinámica. En contraposición a *AngularJS*, *Bootstrap* y *UiBootstrap*, que ofrecen tablas de datos, no son tan buenas como las que nos ofrece *NgTable*, permite filtrar, ordenar y trabajar con hasta cientos de datos.

chartJS

ChartJS [35] es el *framework* que se encarga de la creación de las gráficas. Para su uso simplemente hay que enviarle los datos con los que se quiere que se cree la gráfica, así como indicarle el tipo de gráfica que se va a mostrar y con la configuración de los colores se consigue crear graficas como las mostradas en los prototipos. Asimismo, se ha hecho uso de componentes extra para mejorar la comprensión de los citados datos.

Fonts

Para obtener unas vistas más simples reduciendo texto, se han usado los iconos que ofrece una web [36]. Estos iconos son de múltiples tipos y formas, siendo la mayoría muy reconocibles y fáciles de asimilar con la función que desempeñan en la vista. Gracias a ellos se ha conseguido una reducción de carga del texto en las vistas sustituyéndolo por estos iconos.

3.2.4. Pasarela de pago

En lo referente a la pasarela de pago, se ha realizado un pequeño estudio de las posibilidades que hay en el mercado, habiendo estudiado cuatro plataformas de pago importantes, Tabla 1. Tanto Brian Team como Pay Pal pertenecen a la misma compañía, pero Brian Team es la parte de Pay Pal que se encarga de realizar los pagos en aplicaciones web y Pay Pal es para pagos en general.

	TEFPAY	STRIPE	BRIAN TEAM	PAY PAL
Pago Transacción	20€/mes + 0.35 €	1.4% + 0.25 €	2.9% + 0,3 \$	3.4% + 0,35 €
API NODEJS		✓	✓	✓
Devoluciones		✓	✓	✓

TABLA 1. Pasarelas de pago estudiadas

TefPay [37] es una plataforma usada por el grupo cines Palafox que requiere un pago mensual para ser usada, además de 0.35€ por transacción realizada. Por este motivo se decidió desechar como una opción factible, ya que al ser una asociación se buscaba lo más económico y en este caso no se quiere tener un gasto innecesario cada mes, siendo éste el motivo principal por el cual se ha desechado como una posible pasarela de pago.

Por otro lado, están las dos plataformas del grupo Pay Pal [38], que no requieren ningún tipo de suscripción y son las pasarelas de pago más importante de internet, siendo la pega de éstas su elevada comisión. Haciendo hincapié en “Brian Team” [39], es la pasarela de pago de Pay Pal especializada en la tecnología web, estando pensada e ideada para facilitar las compras por la red, por lo que sería, a priori una de las más indicadas para este proyecto, puesto que brinda la opción de realizar devoluciones y cuenta con una API en *NodeJS*. Pese a que esta opción es una de las que más se adaptan a lo buscado, no es la elegida por el coste, ya que es elevado. A modo de ejemplo, suponiendo un 2,9% más una comisión del 0.3\$ (a día de hoy, 21/11/2018, el cambio corresponde a 0,26€), lo que supondría en una compra de 100€ una comisión de 3,16€ extras.

Respecto a *Pay Pal*, es una empresa especializada en comercio electrónico. Es la más fiable y la mejor del mercado a cambio de una gran comisión, 3,4% de la compra más 0,35€, lo que supone una comisión mucho más elevada que la anterior. Esto hace que sea descartada al instante porque el coste en comisiones en una transacción de **100€** llegaría a ser de **3,75€ extras**, precio que se piensa que es demasiado elevado frente a la opción de “*Stripe*”, que se detalla a continuación.

“*Stripe*” [40] es la más baja en lo que a comisiones se refiere, con 1,4% de la compra total más 0,25€, lo que dejaría una compra de **100€** con un coste de **1,65€ extras**. Asimismo, también tiene un API para *NodeJS*, lenguaje utilizado en el back-end y una gran disponibilidad, siendo del 99,999% en los últimos 90 días [41].

Asimismo, ofrece la posibilidad de realizar devoluciones, pudiendo ser de varias formas: la compra íntegra, parte de la compra o la compra íntegra por partes. Eso sí, solo devuelve el total de la compra sin devolver la comisión. A pesar de esto, que es lógico puesto que la transacción ya está hecha, “*Stripe*” ha sido pasarela de pago elegida, por ser una de las que se adaptaba a las necesidades reales del proyecto, por su bajo coste en comisiones y su gran disponibilidad.

Pese a que las otras pasarelas de pago del grupo *Pay Pal* también se adaptan a las necesidades de la aplicación, no se han elegido por sus elevadas comisiones y cuantías económicas.

Stripe

Stripe es una solución de pago de terceros que permite aceptar pagos con tarjeta de débito y crédito en un comercio o tienda online. Se trata de un sistema seguro y rápido que permite procesar pagos prácticamente al instante, proporcionando API's que los desarrolladores pueden utilizar para integrar procesamiento de pago en sus sitios web y aplicaciones móviles y no móviles.

Tal y como se ha mencionado, posee una API [42] para muchos lenguajes de programación como "Curl", "Ruby", "Python", "PHP", "Java", "NodeJS", "Go" y ".NET". Además, no solo está en el campo de los pagos (como pasarela de pago), sino que al ser "open source" [43], se encarga de apoyar otros campos como pueden ser el "Machine Learning", las bases de datos, infraestructuras, herramientas empresariales y grandes proyectos.

Stripe tiene un portal web que permite realizar todas las consultas necesarias, así como el estado actual de la pasarela de pago, tales como estadísticas gráficas. Además, Stripe también permite una integración total con sus servicios, lo que permite poder utilizar si se desea el API y nada más sin necesidad de conectarse al Dashboard.

Respecto al Dashboard, se trata de una vista muy sencilla en la que se muestran los datos actuales del estado de la pasarela de pago con las últimas acciones realizadas, Figura 14. También

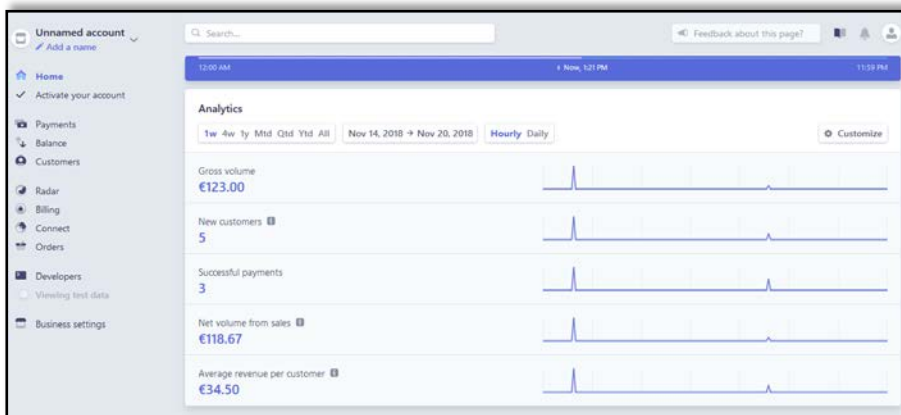


FIGURA 14. Dashboard de Stripe

hay una gráfica que muestra el resultado del día y que permite filtrar por semanas o incluso meses. Tiene un panel en el lateral a mano izquierda desde la que permite desplazarse por todas las opciones que ofrece.

La API de Stripe está organizada y clasificada en secciones (al igual que el dashboard), de forma que, según la información que se desee consultar, hay que ir a una sección o a otra.

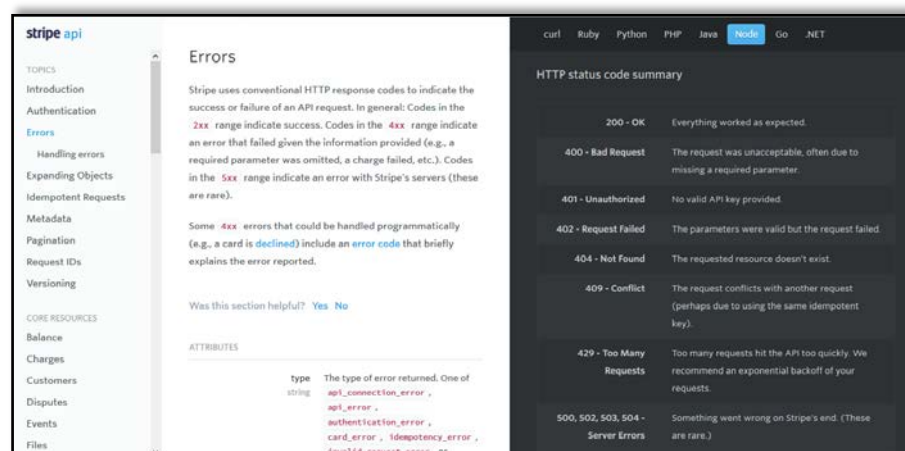


FIGURA 15. API de Stripe

En cada sección se encuentra una explicación y un ejemplo de ejecución, tal y como se puede ver en la Figura 15, donde hay tres secciones. La primera, la de la izquierda, tiene la clasificación de temas para buscar por el tema que interese. La segunda, la central tiene una explicación del funcionamiento. Y la tercera parte, la de la derecha, es donde se encuentra el ejemplo de uso. A modo de ejemplo, la Figura 15 ofrece una vista sobre la sección de errores que devuelve la API de Stripe.

A continuación, se detalla el modo de uso del API de Stripe en este proyecto. Lo primero de todo es cargar el API de Stripe con la clave pública. Esto se debe a que Stripe cifra la información mediante clave pública - clave privada. Una vez se tiene la clave pública con Stripe, ya se puede trabajar con la pasarela de pago. Es importante mencionar que en este proyecto sólo se utiliza el API para realizar los cobros y las devoluciones.

La realización del cobro en este proyecto se hace en dos partes, front-end y el back-end. Una vez se va a realizar el pago en el front-end, salta un modal que pide los datos de la tarjeta con la que se va a realizar el pago, comprobándose automáticamente si esos valores son válidos; si son válidos, este modal envía la información de la tarjeta y la compra al back-end, que envía los datos a Stripe mediante el API para realizar el cobro y éste le devuelve el resultado de la transacción al front-end. Dado que se está trabajando con comercio electrónico y datos reales, toda la información está cifrada y no se puede obtener el número de la tarjeta ni ningún otro dato de la forma de pago, entre otras razones porque Stripe genera un *token* con toda esa información.

Respecto a las devoluciones, es el mismo proceso, mencionando que Stripe sólo va a devolver el dinero que se ha pagado por la inscripción. La acción de devolver se hace en las modificaciones, ya que se comprueba y compara el coste anterior con el coste de la nueva modificación y si la nueva obtiene un resultado positivo, se le devuelve la diferencia. Para realizar una devolución Stripe necesita el id de la compra que hay que devolver y de esa compra devolverá lo que se le indique. El proceso es el mismo que si sale a pagar: a esa compra se le incrementará el coste nuevo. Todos estos procesos se realizan siempre con los datos de la tarjeta de la primera compra.

A continuación, se muestra en detalle técnico la compra y la devolución mediante el API. Para realizar la compra hay que crear un "*customers*", que ha de tener la información del *token* generado por Stripe y que necesita el correo y el id de la tarjeta. Posteriormente se crea el "*charge*", que es la solicitud de cobrar, indicándole el importe y esperando una respuesta de Stripe con el resultado. En el caso de las devoluciones, la acción es muy similar, solo que en lugar de usar "*charge*", se hace uso de "*refunds*", en la cual hay que indicarle el id de la compra en el parámetro "*charge*" y el importe a devolver. En el caso de que no se le indique ningún importe se hará una devolución íntegra de la compra sin las comisiones de éstas.

3.3. Prototipos de la interfaz de usuario

Los prototipos, diseñados para la interfaz de usuario, se han diseñado utilizando Bootstrap [23] y código HTML [30] ya que se trata de una aplicación web. De esta forma se le otorga a cada vista los componentes (color, tipo de fuente...) que serán utilizados directamente cuando la aplicación funcione. Además, para que el prototipo adquiera mayor dinamismo se hace uso de AngularJS [22] para aprovechar el potencial que forman el binomio AngularJS-Bootstrap. Para unir el *framework* de AngularJS y de Bootstrap se unen mediante el framework “uiBootstrap” [32].

En este subapartado, primero se mostrarán los mapas de navegación por los que el usuario se moverá cuando utilice la aplicación y después se mostrarán cada uno de los prototipos de las diferentes pantallas.

3.3.1. Mapa de navegación

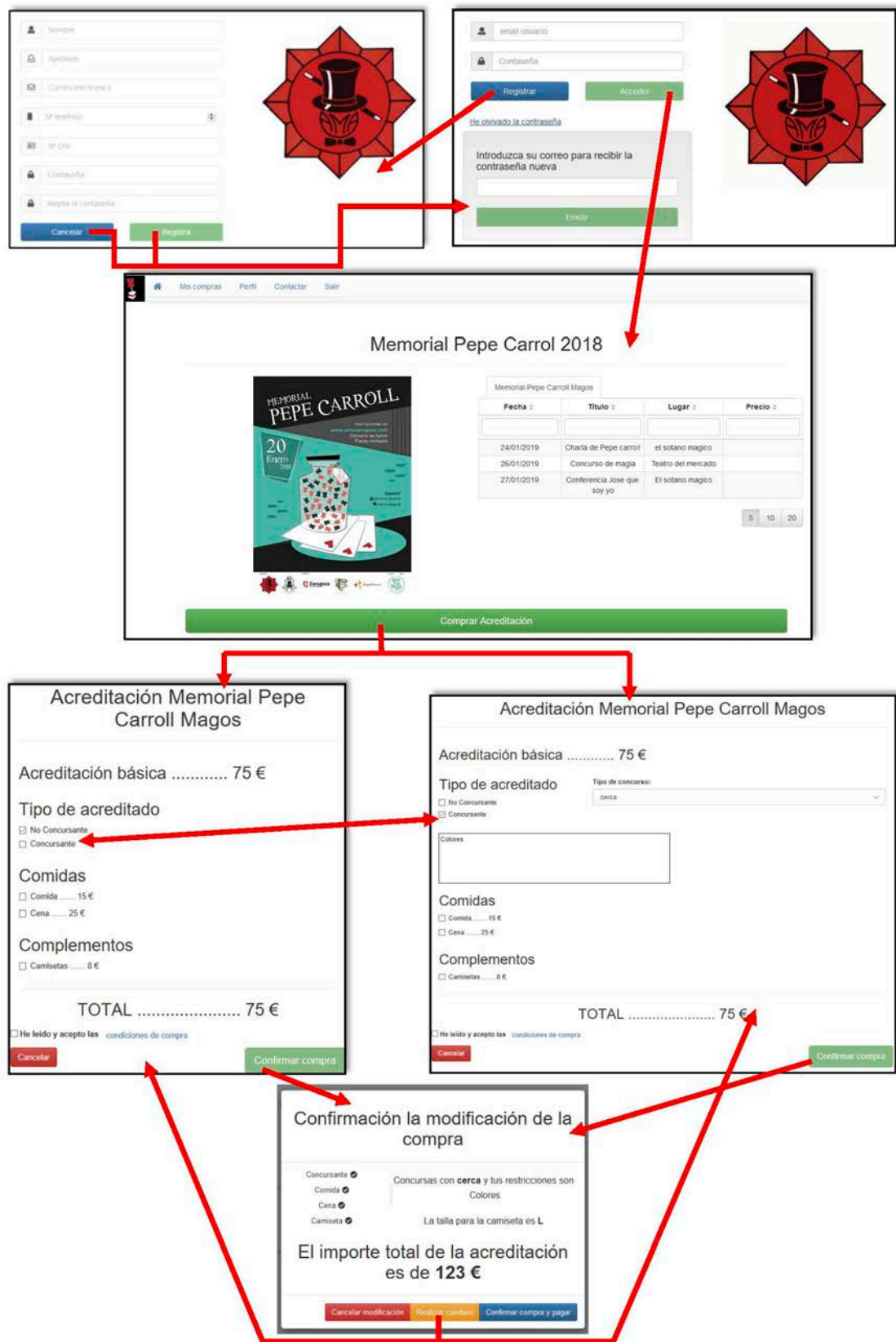


FIGURA 16. Mapa de navegación de Registro, Login y Compra

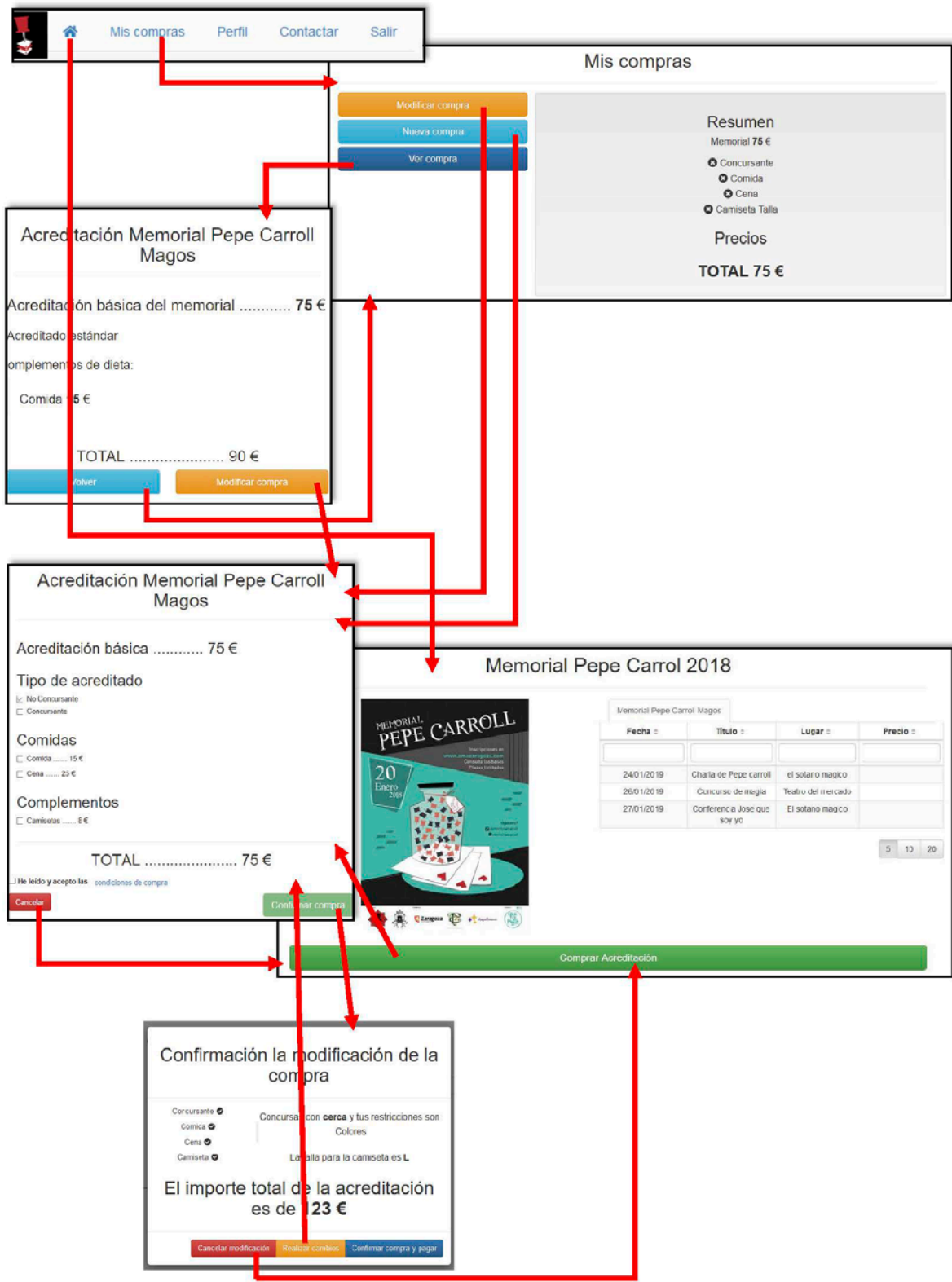


FIGURA 17. Mapa de navegación de "Mis compras"

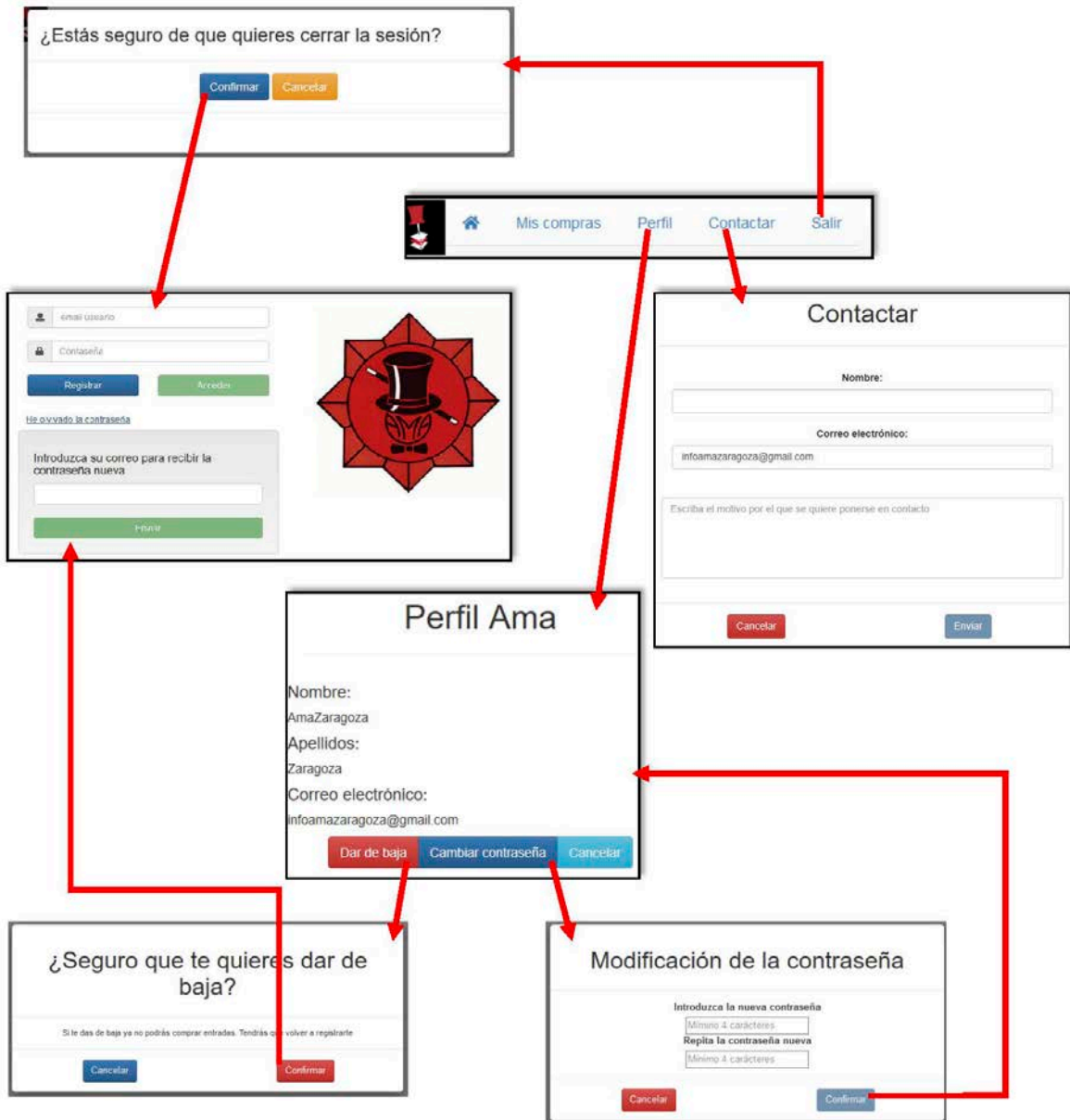


FIGURA 18. Mapa de navegación de Perfil, Contacto y Salir

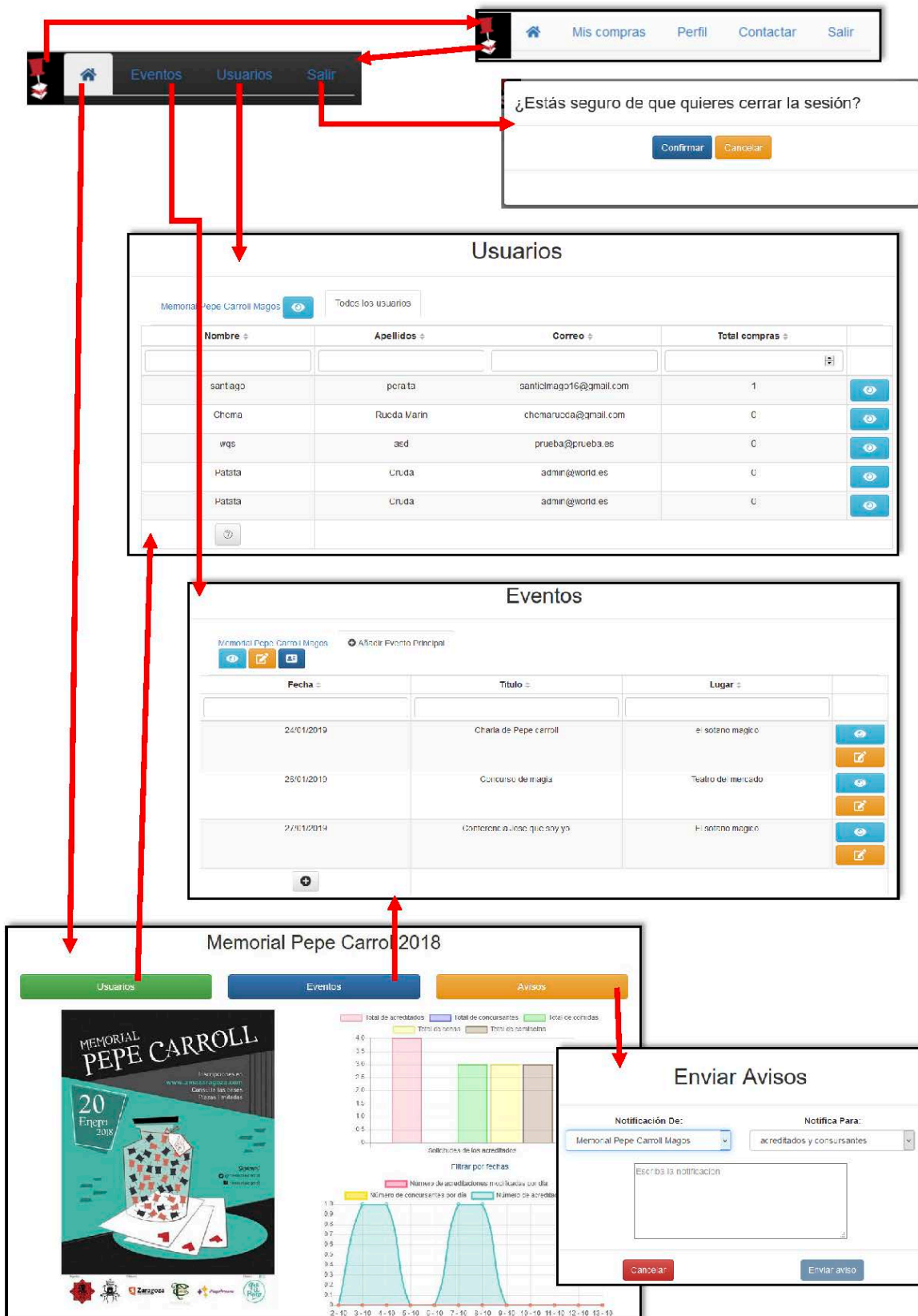


FIGURA 19. Mapa navegación lado Administrador principal

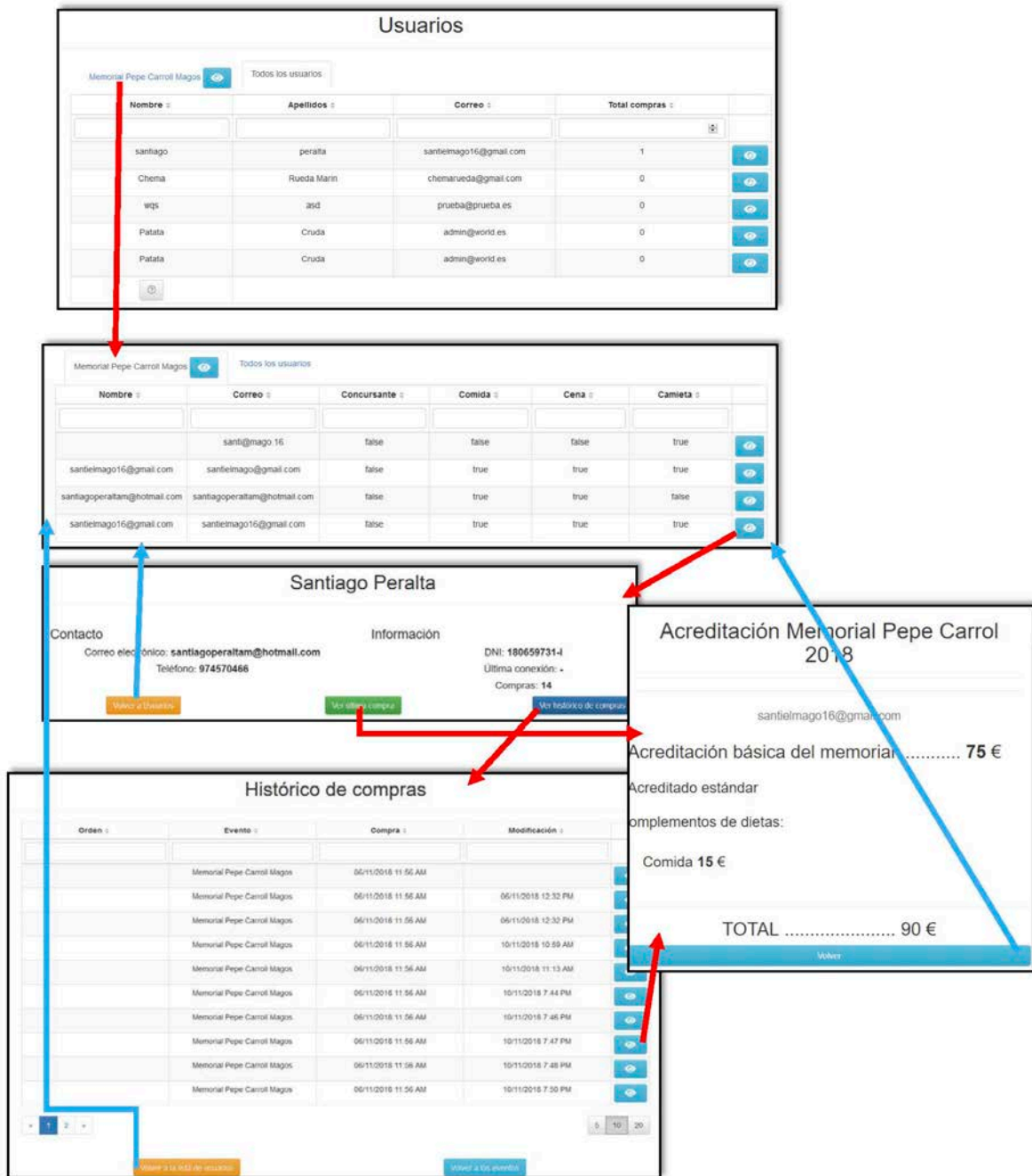


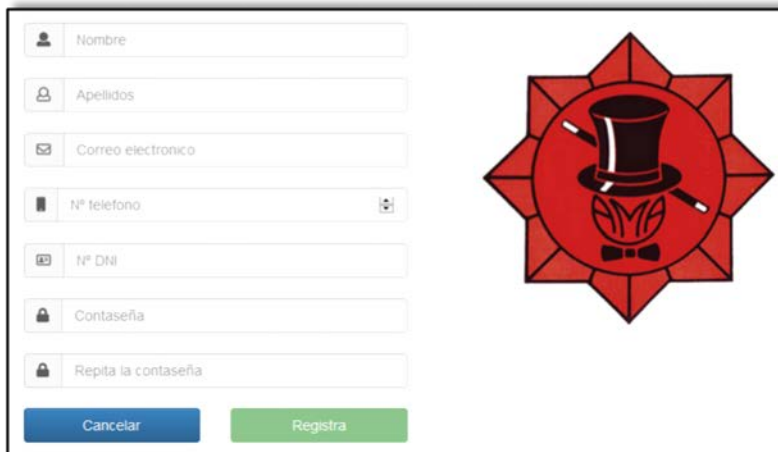
FIGURA 20. Mapa de navegación lado Administrador - Usuario



FIGURA 21. Mapa de navegación de Eventos

3.3.2. Prototipos

Registro de usuarios



Este prototipo muestra un formulario de registro de usuarios. El formulario está dividido en dos columnas. La columna izquierda contiene siete campos de entrada de texto, cada uno con un ícono representativo: un usuario para 'Nombre', un usuario para 'Apellidos', un correo electrónico para 'Correo electrónico', un teléfono para 'N° teléfono', un documento para 'N° DNI', un candado para 'Contraseña' y otro candado para 'Repita la contraseña'. Debajo de los campos hay dos botones: 'Cancelar' (azul) y 'Registra' (verde). La columna derecha contiene un logo circular rojo con un sombrero negro y un lazo negro.

FIGURA 22. Registro de usuarios

La vista del *Registro* permite la gestión de las altas de nuevos usuarios en la aplicación, rellendo el formulario de la Figura 22. Este formulario solicita la información básica necesaria para la realización de compras electrónicas, como son el Nombre, Apellidos, Correo electrónico, teléfono, DNI y la contraseña (la cual se pide dos veces para evitar un posible fallo del usuario).

Login



Este prototipo muestra la pantalla de Login y recuperación de contraseña. El formulario está dividido en dos columnas. La columna izquierda contiene dos campos de entrada de texto: 'email usuario' y 'Contraseña'. Debajo de los campos hay dos botones: 'Registrar' (azul) y 'Acceder' (verde). Debajo de los botones hay un link 'He olvidado la contraseña'. Debajo del link hay un campo de entrada de texto y un botón 'Enviar' (verde). La columna derecha contiene un logo circular rojo con un sombrero negro y un lazo negro.

FIGURA 23. Login y recuperación de contraseña

La vista de *Login* permite al usuario iniciar la sesión. Para iniciar sesión ha de introducir el correo electrónico y la contraseña Figura 23. También permite recuperar la contraseña pinchando en el link "He olvidado la contraseña", desplegando una opción, en la que se ha de introducir el correo electrónico en el que se desea recibir la contraseña.

Vista del Administrador



FIGURA 24. Barra superior del Administrador

Esta es la barra superior del administrador, desde la que se puede acceder a cualquier parte. Esta barra se encuentra en todas las vistas del administrador Figura

24.

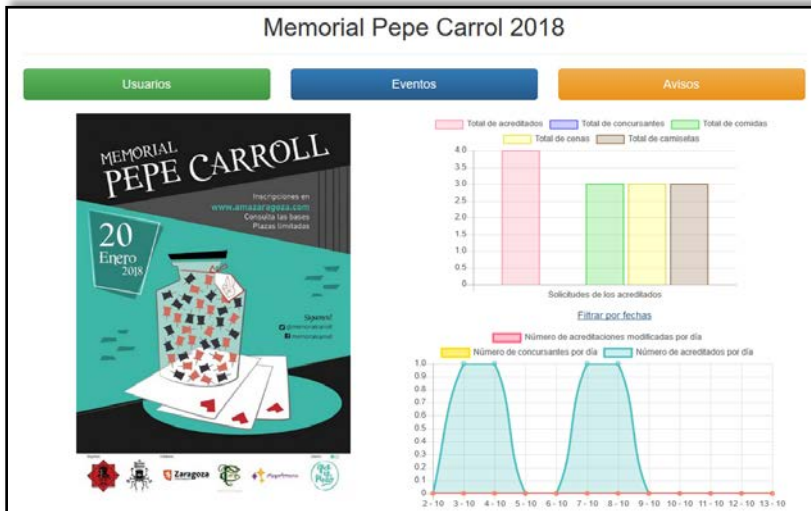


FIGURA 25. Vista principal del Administrador

La Figura 25 muestra la interfaz principal del usuario Administrador. Desde aquí el administrador tiene acceso a todo. Mediante los botones superiores puede acceder a "Usuarios", "Eventos" y "Avisos". La parte inferior tiene información del evento principal. La imagen del evento y las gráficas correspondientes a los datos que resumen de las acreditaciones del evento



FIGURA 26. Filtro de fechas

Esta segunda gráfica, ofrece un filtro que permite, filtrar por fechas como se puede ver en la Figura 26.

Vista Usuarios

Nombre	Apellidos	Correo	Total compras
santiago	peralta	santiemago16@gmail.com	1
Chema	Rueda Marin	chemarueda@gmail.com	0
wqs	asd	prueba@prueba.es	0
Patata	Cruda	admin@world.es	0
Patata	Cruda	admin@world.es	0

FIGURA 27. Lista de todos los usuarios del sistema

La vista de los Usuarios permite ver la lista de usuarios. En esta lista se pueden ver 4 columnas con la información del usuario como el nombre, los apellidos, el correo y el número de compras, Figura 27. Esta lista es de todos los usuarios registrados en la aplicación.

Nombre	Correo	Concursante	Comida	Cena	Camiseta	
	santi@magos.16	false	false	false	true	
santeimago16@gmail.com	santeimago@gmail.com	false	true	true	true	
santiagoperaltam@hotmail.com	santiagoperaltam@hotmail.com	false	true	true	false	
santeimago16@gmail.com	santeimago16@gmail.com	false	true	true	true	

Seleccionando la pestaña del evento, se obtiene una lista de los usuarios inscritos en dicho evento Figura 28.

FIGURA 28. Vista de los usuarios inscritos en el evento

En esta tabla se muestran 6 columnas con los datos de la acreditación del usuario, datos como el nombre, el correo, si es concursante, si tiene la comida, si tiene la cena y si tiene camiseta. Además, en ambas tablas se puede ampliar la información del usuario pinchando en el botón para ampliar la información del evento o del usuario respectivamente. También permite filtra la información de ambas tablas por nombre o por valor numérico, también ordenar ascendentes y descendentes los datos.

Detalle de un usuario

Santiago Peralta

Contacto

Correo electrónico: santiagoperaltam@hotmail.com
Teléfono: 974570466

[Volver a Usuarios](#)

Información

DNI: 180659731-i
Última conexión: -
Compras: 14

[Ver última compra](#) [Ver histórico de compras](#)

FIGURA 29. Detalle sobre un usuario concreto

En la vista mostrada en la Figura 29 del usuario ampliado en detalle se puede ver la información básica del usuario. Y luego mediante los botones se puede *ver la última compra* o *el historial de compras*, Figura 31.

Acreditación Memorial Pepe Carrol 2018

santiagoperaltam@hotmail.com

Acreditación básica del memorial 75 €

Acreditado estándar

Complementos de dietas:

- Comida 15 €
- Cena 25 €

TOTAL 115 €

[Volver](#)

En la Figura 30 se ve la compra del usuario ampliada, es decir, la compra detallada. En esta compra sale la información básica, el precio de la inscripción y el coste de los complementos adquiridos.

FIGURA 30. Detalle de la compra realizada

Histórico de compras

Orden	Evento	Compra	Modificación	
	Memorial Pepe Carroll Magos	06/11/2018 11:56 AM		
	Memorial Pepe Carroll Magos	06/11/2018 11:56 AM	06/11/2018 12:32 PM	
	Memorial Pepe Carroll Magos	06/11/2018 11:56 AM	06/11/2018 12:32 PM	
	Memorial Pepe Carroll Magos	06/11/2018 11:56 AM	10/11/2018 10:59 AM	
	Memorial Pepe Carroll Magos	06/11/2018 11:56 AM	10/11/2018 11:13 AM	
	Memorial Pepe Carroll Magos	06/11/2018 11:56 AM	10/11/2018 7:44 PM	
	Memorial Pepe Carroll Magos	06/11/2018 11:56 AM	10/11/2018 7:46 PM	
	Memorial Pepe Carroll Magos	06/11/2018 11:56 AM	10/11/2018 7:47 PM	
	Memorial Pepe Carroll Magos	06/11/2018 11:56 AM	10/11/2018 7:48 PM	
	Memorial Pepe Carroll Magos	06/11/2018 11:56 AM	10/11/2018 7:50 PM	

« 1 2 » 5 10 20

[Volver a la lista de usuarios](#) [Volver a los eventos](#)

FIGURA 31. Histórico de compras

La vista del *histórico de compras* Figura 31 muestra un listado del histórico de todas las modificaciones que el usuario ha realizado. En esta tabla se puede ver la fecha de la primera compra y las fechas en las que se han realizado las modificaciones. Además se puede ampliar el detalle de la compra modificada pinchando en el botón . Esta vista en compra en detalle se muestra como la vista superior.

Eventos

Fecha	Título	Lugar	
24/01/2019	Charla de Pepe Carroll	el sotano magico	
26/01/2019	Concurso de magia	Teatro del mercado	
27/01/2019	Conferencia José que soy yo	El sotano magico	

« 5 10 20 »

FIGURA 32. Vista de los eventos

Esta es la vista de los *eventos*, Figura 32. Aquí se muestran todos los Eventos Principales con sus eventos. Como se puede apreciar la lista de los eventos incluidos en un evento principal muestra la fecha, el título y el lugar de dicho evento. Con estos eventos se puede ampliar el detalle del evento pinchando en el botón o modificar dicho evento pinchando en . Estos botones tienen la misma funcionalidad para el evento principal que para el evento. Además el evento principal tiene un botón más, el de las acreditaciones , el cual lleva a las acreditaciones, donde se trabaja con ellas.

Creación de un nuevo evento principal

The screenshot shows a form for creating a new event. It features two calendar pickers for 'Fecha de inicio' (start date) and 'Fecha de finalización' (end date). The start date is set to Wednesday, November 14, 2018, and the end date is Thursday, November 15, 2018. Below the calendars, it indicates 'Duración del evento 1 días'. The 'Información del evento' section includes a 'Nombre y descripción' field with a 'Nombre' input and a description area, and an 'Información adicional' section with a 'Plazas disponibles' input and an 'Examinar...' button for file uploads. A 'Cancelar' button is on the left and a 'Crear evento' button is on the right.

FIGURA 33. Creación de un evento principal

La Figura 33 muestra la vista de la *creación de un evento principal*. Aquí se crea el evento principal, con la información necesaria como fechas de inicio y de finalización, duración, el nombre y la descripción del evento, las plazas de las que dispone el evento y se le puede adjuntar una imagen.

The screenshot displays the details for the 'Memorial Pepe Carroll Magos' event. It includes a calendar for January 2019 with the start date (24/01/2019) marked in green and the end date (27/01/2019) marked in red. The event duration is '3 días de evento'. The title is '3º Edición del Memorial Pepe Carroll' with '100 plazas disponibles'. A poster for the event is shown on the right, featuring a deck of cards and the text 'MEMORIAL PEPE CARROLL', '20 Enero 2018', and 'Inscripciones en www.ajmzaragoza.com'. At the bottom, there are 'Ver' and 'Modificar' buttons.

FIGURA 34. Vista del evento principal

La Figura 34 muestra el evento principal en detalle. Aquí se muestra la duración del evento en un único calendario marcado en verde el inicio y en rojo la finalización. También se muestra el título, la descripción del evento y las plazas que dispone. Mientras que a la derecha de esta información se muestra el cartel del evento.

Modificación del evento

Fecha de inicio: Thursday, January 24, 2019

Fecha de finalización: Sunday, January 27, 2019

Duración del evento 3 días

Información del evento

Nombre y descripción: Memorial Pepe Carroll Magos

3ª Edición del Memorial Pepe Carroll

Información adicional: 100 plazas

Examinar... No se ha seleccionado ningún archivo.

Cancelar Modificar evento

FIGURA 35. Modificación de un evento

La vista de la *modificación del evento* Figura 35 es muy similar a la de creación, solo que éste ya viene con todos los valores añadidos. Es decir, esta vista se carga con la información del evento y se puede modificar los valores que se muestran.

Acreditación

Acreditaciones Memorial Pepe Carroll Magos

Datos de las acreditaciones

Los acreditados pueden ser concursantes

Precio de las acreditaciones: 75 €

Datos de los concursantes: 12

Especialidad del concurso

Cerca Salón Grandes Ilusiones Micromagia

EXTRAS

Datos de dietas

Comida: 15 €

Cena: 25 €

Merchandising

Camiseta: 8 €

Tallas disponibles: S M L XL XXL

Cancelar Aceptar

FIGURA 36. Configuración de la acreditación

La Figura 36 muestra la creación y modificación de las acreditaciones. Para las acreditaciones hay que indicar el precio de las inscripciones, indicando si hay concurso o no, y el precio de inscripción a este en caso de que lo haya, también hay que indicar el número máximo de concursantes. También se puede indicar la especialidad del concurso, con unos valores predefinidos de las disciplinas más comunes en concursos.

Por otro lado, en la parte de extras se ofrece la opción de añadirlos y de indicar el precio, tanto de la comida como de la cena. Respecto al merchandising, sigue el mismo estilo que con las dietas, solo que se puede elegir las tallas de las camisetas, pudiendo así actualizar las tallas en cualquier momento, permitiendo que se puedan ir variando los precios y modificando las ofertas de los distintos productos, en tiempo real. Lo que se verá reflejado en la compra de acreditaciones Figura 44.

Evento dentro de un evento principal

The screenshot shows a web form for creating a sub-event. At the top left is a calendar for January 2019 with dates 01 through 10 highlighted. To the right are time selection controls for HH and MM, and a 'Hora:' field set to '12H / 24H'. Below the calendar is a 'Fecha:' label. The main form area includes a 'Tipo:' dropdown menu with 'Memorial Pepe Carroll Magos' selected, a 'Lugar:' text input, a 'Titulo' text input, and a larger text area for 'Escriba la descripción del evento'. At the bottom are two buttons: 'Cancelar' (red) and 'Crear evento' (green).

FIGURA 37. Creación de un evento dentro de un evento principal

La Figura 37 muestra la vista de la *creación de un evento* para un evento principal. Como se puede ver en la elección de la fecha de este evento hay un calendario, el cual está delimitado con la duración del evento principal, es decir, solo se puede elegir una fecha que este dentro del margen del evento principal. Además de ello se le ha de indicar un horario de inicio (pudiendo mostrarse en 12 o 24 horas), el título, una descripción del evento y el lugar donde se celebra.

Resumen del evento

The screenshot shows the summary page for an event titled 'Concurso de magia' under the category 'Memorial Pepe Carroll Magos'. It features a calendar for January 2019 with the date 26 highlighted. To the right, the event details are displayed: 'Evento 26/01/2019 a las 10:00 AM', 'Se verán los 10 numeros de los concursantes del evento', and 'Actividad incluida en la inscripción'. At the bottom are two buttons: 'Volver' (blue) and 'Modificar' (orange).

FIGURA 38. Resumen de un evento

La Figura 38 muestra la información del evento. En la que se puede ver la fecha del evento y los datos de este.

Modificación del evento

January 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
31	01	02	03	04	05	06
07	08	09	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	01	02	03
04	05	06	07	08	09	10

Fecha: Thursday, January 24, 2019

Tipo: Memorial Pepe Carroli Magos

Charla de magos Aragonesés

Los magos Aragoneses nos hablaran de sus vivencia con Pepe Carroli

El sótano mágico

Esta actividad esta incluida en el precio del memorial

Cancelar Modificar evento

FIGURA 39. Modificación de un evento

La Figura 39 muestra la vista de *modificación del evento*. Esta vista es cargada con la información actual del evento, la cual puede ser modificada. Esta vista es como la de Creación del evento.

Avisos

Enviar Avisos

Notificación De: Memorial Pepe Carroli Magos

Notifica Para: acreditados y concursantes

Escriba la notificacion

Cancelar Enviar aviso

FIGURA 40. Modal para enviar notificaciones

La Figura 40 muestra cómo se pueden enviar avisos a los inscritos a un evento principal. Para enviar el aviso simplemente sea de indicar al evento que se desea notificar, y luego a qué tipo de inscritos, concursantes, no concursantes o a todos los inscritos. Escribiendo el mensaje deseado en el cuadro de texto.

Todas estas vistas son las del lado del administrador, el que crea y gestiona los eventos y todos los derivados de los eventos como las inscripciones y los usuarios inscritos.

Vista del Usuario

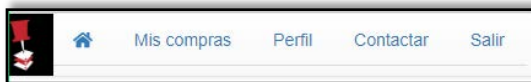


FIGURA 41. Barra superior del usuario estándar

Esta es la barra superior del usuario estándar Figura 41, desde la que se puede acceder a cualquier parte del usuario. Esta barra se encuentra en todas las vistas del usuario.

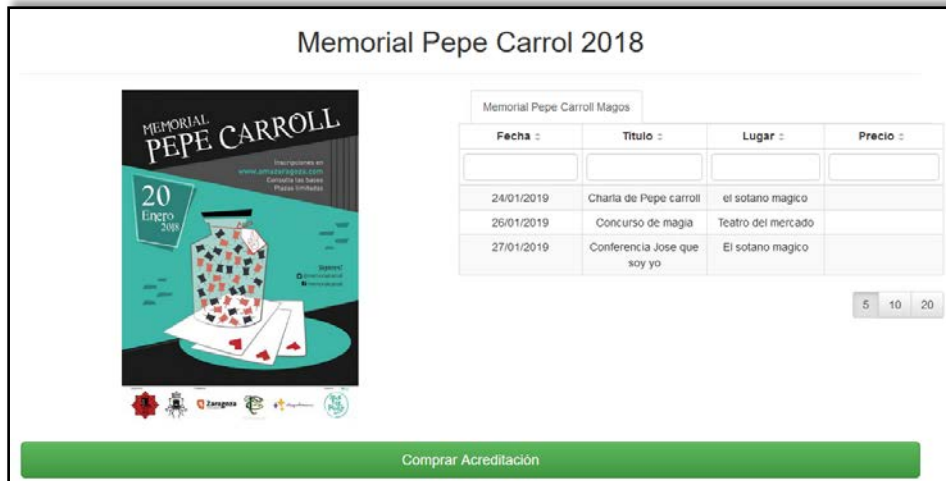


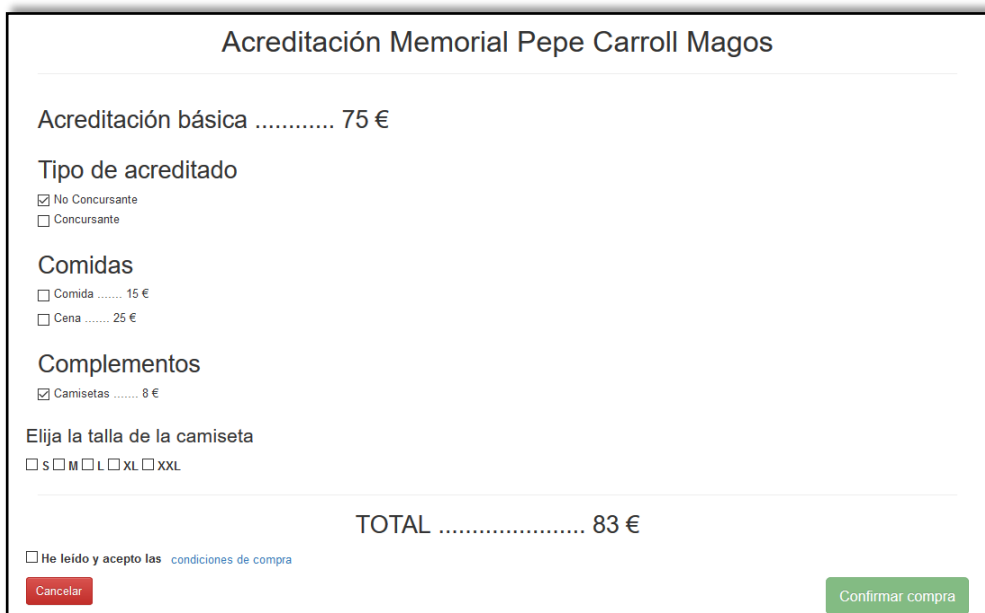
FIGURA 42. Vista principal del usuario estándar

La Figura 42 muestra la vista principal del usuario. El usuario tendrá una vista del evento con sus actividades, la cual muestra la fecha, título, descripción y precio, en caso de que lo tuviera, del evento.

Además, se le da acceso a la compra mediante el botón "Comprar acreditación", con el objetivo de realizar la compra.

Compra de acreditación normal

La Figura 43 muestra la compra de acreditación para el evento. Es importante mencionar que una compra no se puede aceptar hasta que se confirmen las condiciones de compra. En la acreditación se puede elegir entre todos los complementos disponibles por el administrador, el cual puede modificar las opciones desde su vista, como se muestra en la Figura 36. En la acreditación se puede elegir entre todos los complementos disponibles por el administrador, el cual puede modificar las opciones desde su vista.



Acreditación Memorial Pepe Carroll Magos

Acreditación básica 75 €

Tipo de acreditado

No Concursante
 Concursante

Comidas

Comida 15 €
 Cena 25 €

Complementos

Camisetas 8 €

Elija la talla de la camiseta

S M L XL XXL

TOTAL 83 €

He leído y acepto las condiciones de compra

FIGURA 43. Compra de acreditación

Compra de acreditación con opción de concursante

Acreditación Memorial Pepe Carroll Magos

Acreditación básica 75 €

Tipo de acreditado

No Concursante
 Concursante

Tipo de concurso:

cerca
cerca
sala
maratón
grande fusión

Indique las necesidades para concursar y un resumen del mismo

Comidas

Comida 15 €
 Cena 25 €

Complementos

Camisetas 9 €

Elija la talla de la camiseta

S M L XL XXL

TOTAL 83 €

He leído y acepto las condiciones de compra

En caso de que se desee ser concursante se le pedirán unos datos auxiliares, como se observa en vista Figura 44. Estos datos son en la modalidad que desea competir y una ligera descripción del número a realizar, esta última opción es recomendable.

FIGURA 44. Compra de acreditación como concursante

Mis compras

La Figura 45 muestra un resumen rápido de la compra actual, ofreciendo además las opciones en los botones laterales. Estas opciones son la de ver o modificar la compra actual o la de crear una nueva compra.

Mis compras

Modificar compra

Nueva compra

Ver compra

Resumen

Memorial 75 €

- ✖ Concursante
- ✖ Comida
- ✖ Cena
- ✖ Camiseta Talla

Precios

TOTAL 75 €

FIGURA 45. Vista de "Mis compras"

Vista de la compra

Acreditación Memorial Pepe Carroll Magos

Acreditación básica del memorial 75 €

Acreditado estándar

Complementos de dieta:

Cena 25 €

Complementos merchandising:

Camiseta 8 €

Talla L

TOTAL 108 €

Volver Modificar compra

FIGURA 46. Vista del detalle de una compra

En esta vista, Figura 46 se puede *ver la compra* que el usuario acaba de realizar.

Mostrando los complementos adquiridos con su coste y dando la opción de modificar esta mediante el botón inferior.

Vista de confirmación y modificación de compra

Confirmación la modificación de la compra

Concursante ✓
Comida ✓
Cena ✓
Camiseta ✓

Concursas con **cerca** y tus restricciones son Colores

La talla para la camiseta es L

El importe total de la acreditación es de **123 €**

Cancelar modificación Realizar cambios Confirmar compra y pagar

FIGURA 47. Modal de confirmación de compra y modificación

Al igual que con el resto de modificaciones, la modificación de la compra es completamente igual que el proceso de compra. Cuando el usuario efectúa una compra o una modificación, el modal que aparece en la Figura 47 aparece un resumen de la compra. Este aparece en la compra también. Se muestra antes de confirmar una compra o una modificación, ya que este contiene un resumen de la compra actual y te da la opción de realizar algún cambio antes de confirmar la compra, esto se hace mediante los botones inferiores.

Perfil del usuario

Perfil santiago

Nombre:
santiagoperalta

Apellidos:
peralta

Correo electrónico:
santielmago16@gmail.com

Dar de baja Cambiar contraseña Cancelar

FIGURA 48. Vista del perfil de usuario

La vista del *perfil de usuario*, Figura 48, muestra la información al usuario. Además, le permite que cambie la contraseña y darse de baja, gracias a los botones inferiores.

¿Seguro que te quieres dar de baja?

Si te das de baja ya no podrás comprar entradas. Tendrás que volver a registrarte

Cancelar Confirmar

FIGURA 49. Modal de confirmación de baja de usuario

En la vista de darse de baja, Figura 49, se interactúa con dos botones, el de confirmación, que da de baja al perfil de usuario y el de cancelar.

Modificación de la contraseña

Introduzca la nueva contraseña

Mínimo 4 caracteres

Repita la contraseña nueva

Mínimo 4 caracteres

Cancelar Confirmar

FIGURA 50. Modal de cambio de contraseña

Mientras que en la vista del cambio de contraseña, Figura 50, contiene dos campos de texto en los que hay que introducir la nueva contraseña para comprobar si es la misma contraseña y no se ha equivocado el usuario, si la contraseña es igual en ambos campos se admitirá la nueva contraseña.

Contacto

Contactar

Nombre:

Correo electrónico:
santielmago16@gmail.com

Escriba el motivo por el que se quiere ponerse en contacto

Cancelar Enviar

FIGURA 51. Pestaña de contacto

En la Figura 51 se muestra la vista de *contacto*, donde se envía un email al administrador de la web para ponerse en contacto por cualquier motivo. Tiene tres campos disponibles, el del nombre del usuario, el correo electrónico al que deseas recibir la respuesta y el campo donde va el motivo por el que se contacta.

3.4. Despliegue

Para el despliegue de la aplicación se han usado dos plataformas online que permiten realizar despliegues. El despliegue de la aplicación se realiza a través de dos plataformas. En la primera se hará trabajar a la base de datos y en la segunda la aplicación propiamente dicha.

En la aplicación se utiliza “Heroku” [17], que es una plataforma web que permite realizar el despliegue del proyecto de forma manual y automática. Asimismo, también permite enlazar un proyecto de GitHub con la cuenta de “Heroku” que se haya creado, de forma que cada vez que se realicen modificaciones en el repositorio GitHub, éstas se actualizan en Heroku, que realiza el despliegue de forma automática.

Una desventaja de esta plataforma es que no ofrece servicio para la base de datos MongoDB [18]. Por ello, es necesario usar otra plataforma. Para la aplicación se ha optado por “mLab” [19], tal y como se describe a continuación.

Se comienza por la base de datos y se termina con el despliegue de la app en “Heroku”, cuya URL se encuentra en <https://memorialpepecarroll.herokuapp.com/>

mLab

Para el despliegue de la base de datos se ha usado “mLab”. Se trata de una web que permite realizar el despliegue de una base de datos NoSql, como es MongoDB. Admite el almacenamiento de datos gratuitamente hasta 500 MBytes, pudiendo elegir entre las tres grandes empresas del Cloud mundial, Amazon (AWS) [44], GoogleCloud [45] y Microsoft Azure [46].

El despliegue de la base de datos es muy sencillo ya que sólo hay que crear una base de datos en cualquiera de los servidores de las empresas Cloud, para después introducir la URL suministrada en el código. De esta forma se consigue conectar la Base de datos de *mLab* con nuestra aplicación web.

En la Figura 52 (Figura original [47]) se puede comprobar cómo quedaría la arquitectura del despliegue de mLab.

Heroku

El despliegue de la aplicación, Figura 53 (Figura original [48]) y, se va a realizar a través de “Heroku”, tal y como se ha descrito anteriormente. Se trata de una plataforma que permite realizar despliegues de aplicaciones web. Dichos despliegues son gratuitos y se basan en “Dynos”, de forma que suministran un “Dyno”, que es básicamente el nodo que ofrecen para poder trabajar.

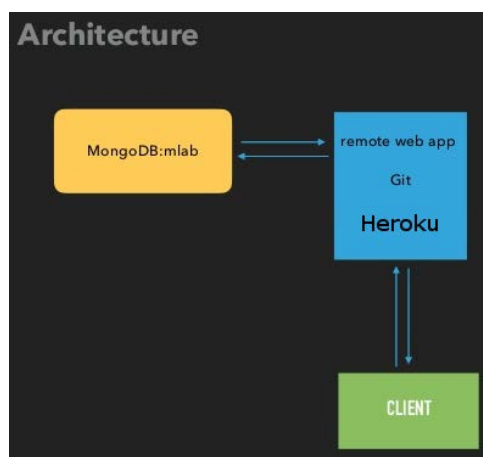


FIGURA 52. Arquitectura del despliegue.

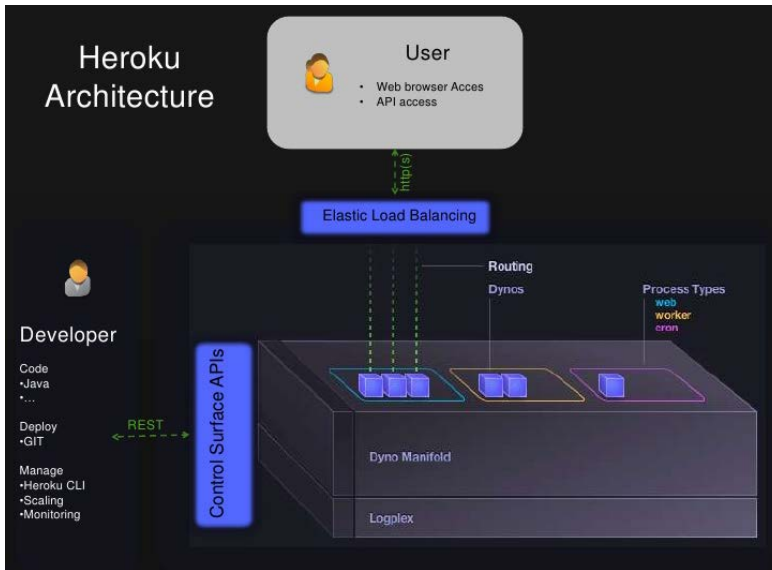


FIGURA 53. Arquitectura del despliegue de Heroku

“Heroku” permite realizar auto-escalados de forma automática. Dichos escalados se realizan añadiendo “Dynos”, con el objetivo de evitar que la aplicación web deje de prestar servicio. Otra cosa que ofrece “Heroku” es que la aplicación no está en funcionamiento continuo, es decir, ésta se encuentra suspendida mientras no recibe ninguna petición; En el momento en el que recibe una petición, ésta se levanta y queda lista para ser usada. Una vez usada, si en treinta minutos no ha habido actividad, se vuelve a suspender.

Como inconveniente a este tipo de despliegues gratuitos: sólo se dispone de un “Dyno”, el cual no tiene escalado. La ventaja de “Heroku” es que permite el despliegue sobre Amazon, AWS. Esto es una ventaja porque la base de datos está desplegada también en AWS, lo que hace que tanto la aplicación web como la base de datos estén muy cerca, reduciéndose la latencia. Además, AWS ofrece una disponibilidad del 99.999%.

El despliegue de la aplicación web se puede realizar de dos maneras posibles: desde la web o desde la consola de comandos del usuario. En el despliegue desde la web lo primero que hay que hacer es crear el “Dyno”, indicándole el nombre, que será la URL para acceder a la aplicación web y elegir la región donde se desee hacer el despliegue. Seguidamente hay que indicarle el repositorio “Git” y sincronizarlo. Una vez hecho este paso, hay que pulsar en el botón de “Despliegue” para lanzarlo al servidor. “Heroku” ofrece la opción de actualizar el repositorio “Heroku” cada vez que detecte un cambio en el repositorio indicado de “Git”.

Por otro lado, si se hace un despliegue a través de la consola de comandos del usuario, ésta ha de ser en base UNIX. El “Dyno” es creado automáticamente por el sistema y lo que se sincroniza es el directorio que se encuentre en el ordenador desde el que se está invocando a la aplicación.

4. COSTES Y PLANIFICACIONES

El proyecto se concretó a finales de junio, principios de julio. Desde el principio se habló con la Asociación Mágica Aragonesa para ver y analizar cuáles eran sus necesidades. De esta forma, durante todo el mes de agosto se analizaron las distintas necesidades para después realizar los distintos prototipos en papel.

Tras los exámenes de septiembre se comenzó con la implementación del código, resolviendo dudas y corrigiendo errores. Por último, se planificó realizar la memoria explicativa del proyecto durante el mes de noviembre.

Este proyecto ha seguido un modelo de diseño en cascada con retroalimentación ya que el presidente del AMA ha estado al corriente de los distintos avances, lo que ha hecho tener varias iteraciones mejorando o añadiendo detalles.

Con esta explicación, la distribución de tiempo, Tabla 2 , ha sido la siguiente:

	Análisis	Diseño	Implementación	Memoria	Horas totales
Agosto	15	16			31
Septiembre	10	11	63	5	89
Octubre	6	9	90	11	116
Noviembre			55	52	107
TOTAL	31	36	208	68	343

TABLA 2. Planificación final

5. MEJORAS FUTURAS

El prototipo desarrollado cumple con todos los objetivos que se habían marcado para su realización, aunque es posible añadir nuevas funcionalidades y mejorar las ya existentes.

- Añadir funcionalidades para la creación de códigos QR [47] con la información relativa a la inscripción del evento. De esta forma, ésta sería leída por los organizadores del evento y podrían actualizar los datos de participación en tiempo real.
- Sería útil agregar encuestas de satisfacción al finalizar el evento para obtener datos de complacencia de los inscritos y poder mejorar cada evento para el siguiente.
- Si la aplicación tiene éxito, se podría desarrollar la aplicación para los sistemas operativos móviles Android e iOS mediante tecnologías como Cordova [48]. Se trata de un framework multiplataforma que permite crear aplicaciones móviles para los sistemas operativos mencionados mediante código JavaScript, lo que hace que se pueda convertir un proyecto MEAN como este en una aplicación móvil.
- Sería útil añadir soporte multi idioma con el fin de que la aplicación pueda ser utilizada en mercados extranjeros y por usuarios que no hablen el castellano.
- Como en cualquier aplicación de venta de entradas, por ejemplo, un cine, sería útil una funcionalidad para poder elegir los asientos o butacas del lugar donde se vaya a celebrar el evento.

6. CONCLUSIONES

Gracias a la tecnología Stack MEAN, es mucho más sencillo crear un proyecto adaptado a las necesidades del cliente con muy poco coste. Este motivo por el que se ha utilizado esta tecnología es la libertad brindada para crear grandes cosas partiendo de pequeñas ideas. Además de toda la infraestructura que tiene por detrás, la gran comunidad que se encarga de crear y mejorar los distintos frameworks y librerías que facilitan la vida del desarrollador. También gracias a las empresas que ofrecen infraestructura gratuita y que permite que pequeñas empresas y asociaciones sin demasiado capital pueden invertir en este tipo de tecnologías, a sabiendas de que se les ofrece infraestructura gratuita.

Utilizar MEAN ha sido un gran acierto, puesto que se ha conseguido crear esta aplicación de la nada en poco más de tres meses, creando una solución que se ajusta al máximo a las necesidades iniciales, ***“Poder modificar inscripciones y facilitar la gestión de las entradas del Memorial Pepe Carroll”***. Además, tratándose de una asociación, gracias a las empresas Heroku y mLab, se ha conseguido desplegar la aplicación web en el mundo real completamente gratis, ofreciendo un dominio bajo el nombre de Heroku, con la ventaja de ser una aplicación web permite que ésta sea usada por todos los organizadores del evento sin importar la plataforma desde la que se invoque: desde móvil con Android hasta portátiles con Linux, pasando por IOS o Microsoft.

La aplicación final es totalmente funcional, de fácil implantación, ya que sólo se necesita un dispositivo conectado a Internet y tenerla desplegada, siendo de fácil mantenimiento gracias a las tecnologías utilizadas. También se ha intentado que sea lo más genérica posible con el fin de que pueda ser utilizada por esta asociación u otras asociaciones si lo necesitaran.

Por tanto, este proyecto ha servido también para constatar la dificultad de la tarea de planificación de un proyecto de software de estas características con el tiempo limitado para su defensa en el mes de diciembre, dado que se empezó a finales del mes de agosto. Resulta muy difícil anticipar los distintos problemas y dificultades que pueden surgir a lo largo de las distintas etapas, por lo que siempre es conveniente estimar algunos días más de los inicialmente calculados para poder disponer de cierta holgura que satisfaga la resolución de esos problemas que siempre aparecen de la nada.

Por último, es necesario mencionar, más de forma personal, que el desarrollo de este Trabajo Fin de Grado ha servido para repasar y afianzar muchos de los conceptos y conocimientos impartidos a lo largo de esta etapa universitaria, así como para asumir el reto de realizar un proyecto ingenieril para la vida real, en este caso, una asociación de magia. A pesar de haber entrañado algunas dificultades, también ha sido gratificante ver cómo se han podido superar uno a uno los distintos problemas que han ido surgiendo. De igual forma, todo el trabajo de investigación que ha sido necesario ha terminado siendo muy enriquecedor.

Como culmen a todo lo explicado en el párrafo actual, la realización de este proyecto ha supuesto superar un hándicap, puesto que siempre había sido reacio al tema del comercio electrónico y las compras por Internet, por lo que ha sido necesario familiarizarse con los métodos de pago y con la forma de obrar de este tipo de plataformas.

7. BIBLIOGRAFÍA

- [1] R. Mayrata, «fronterad,» 8 julio 201. [En línea]. Available: <http://www.fronterad.com/index.php?q=bitacoras/ramonmayrata/cuando-pepe-carroll-sono-vivir-en-pais-maravillas>. [Último acceso: 21 11 2018].
- [2] A. M. Aragonesa, «AMA,» [En línea]. Available: <https://amazaragoza.com/>. [Último acceso: 21 11 2018].
- [3] S. d. I. Española, «SEI,» [En línea]. Available: <http://web.seimadrid.es/>.
- [4] «Heraldo,» [En línea]. Available: <https://www.heraldo.es/noticias/ocio-cultura/2018/01/18/la-magia-escapa-del-sotano-segunda-edicion-del-memorial-pepe-carroll-1219668-1361024.html>. [Último acceso: 21 11 2018].
- [5] «Google form,» [En línea]. Available: <https://www.google.es/intl/es/forms/about/>. [Último acceso: 21 11 2018].
- [6] «Tiketttailor,» [En línea]. Available: <https://www.tiketttailor.com/>. [Último acceso: 21 11 2018].
- [7] «binaryis,» [En línea]. Available: <http://www.binaryis.com/Ticketing.aspx>. [Último acceso: 21 11 2018].
- [8] «hiberus IACPOS,» [En línea]. Available: <https://www.hiberus.com/crecemos-contigo/funcionalidades-que-debe-tener-un-software-de-gestion-de-venta-de-entradas-online/>. [Último acceso: 21 11 2018].
- [9] «Eventboost,» [En línea]. Available: <http://www.eventboost.com/es-ES/>. [Último acceso: 21 11 2018].
- [10] «Shopify,» [En línea]. Available: <https://www.shopify.es/>. [Último acceso: 21 11 2018].
- [11] «Hooh,» [En línea]. Available: <http://www.hoohsolution.com/la-aplicacion/>. [Último acceso: 21 11 2018].
- [12] «Gestion Entradas,» [En línea]. Available: <https://es.gestionentradas.com/>. [Último acceso: 21 11 2018].
- [13] «TicketMaster,» [En línea]. Available: <https://www.ticketmaster.es/>. [Último acceso: 21 11 2018].
- [14] «Tikettea,» [En línea]. Available: <https://www.tikettea.com/>. [Último acceso: 21 11 2018].
- [15] «MEAN.org,» [En línea]. Available: <http://meanjs.org/>. [Último acceso: 21 11 2018].
- [16] «Json.org,» [En línea]. Available: <https://www.json.org/>. [Último acceso: 21 11 2018].

- [17] «Heroku,» [En línea]. Available: <https://www.heroku.com/>. [Último acceso: 21 11 2018].
- [18] «MongoDB,» [En línea]. Available: <https://www.mongodb.com/es>. [Último acceso: 21 11 2018].
- [19] «mLab,» [En línea]. Available: <https://mlab.com/>. [Último acceso: 21 11 2018].
- [20] J. M. Alarcón, «campusmvp,» 19 1 2015. [En línea]. Available: <https://www.campusmvp.es/recursos/post/Que-es-el-stack-MEAN-y-como-escoger-el-mejor-para-ti.aspx>. [Último acceso: 21 11 2018].
- [21] «NodeJs.org,» [En línea]. Available: <https://nodejs.org/es/>. [Último acceso: 21 11 2018].
- [22] «AngularJS.org,» [En línea]. Available: <https://angularjs.org/>. [Último acceso: 21 11 2018].
- [23] «Bootstrap,» [En línea]. Available: <https://getbootstrap.com/docs/3.3/>. [Último acceso: 21 11 2018].
- [24] «mongooseJS,» [En línea]. Available: <https://mongoosejs.com/>. [Último acceso: 21 11 2018].
- [25] «npm,» [En línea]. Available: <https://www.npmjs.com/package/bcrypt>. [Último acceso: 21 11 2018].
- [26] «npm,» [En línea]. Available: <https://www.npmjs.com/>. [Último acceso: 21 11 2018].
- [27] «ExpressJS,» [En línea]. Available: <https://expressjs.com/es/>. [Último acceso: 21 11 2018].
- [28] «nodeMailer,» [En línea]. Available: <https://nodemailer.com/about/>. [Último acceso: 21 11 2018].
- [29] «PassportJS,» [En línea]. Available: <http://www.passportjs.org/>. [Último acceso: 21 11 2018].
- [30] «w3schools,» [En línea]. Available: <https://www.w3schools.com/>. [Último acceso: 21 11 2018].
- [31] «w3schools,» [En línea]. Available: <https://www.w3schools.com/css/>. [Último acceso: 21 11 2018].
- [32] «angular-ui,» [En línea]. Available: <https://angular-ui.github.io/bootstrap/>. [Último acceso: 21 11 2018].
- [33] «ui-router,» [En línea]. Available: <https://ui-router.github.io/>. [Último acceso: 21 11 2018].
- [34] «ngTable,» [En línea]. Available: <http://ng-table.com/#/>. [Último acceso: 21 11 2018].
- [35] «ChartJS.org,» [En línea]. Available: <https://www.chartjs.org/>. [Último acceso: 21 11 2018].
- [36] «fontawesome,» [En línea]. Available: <https://origin.fontawesome.com/>. [Último acceso: 21 11 2018].
- [37] «PayTef,» [En línea]. Available: <http://www.paytef.es/>. [Último acceso: 21 11 2018].

- [38] «Pay-Pal,» [En línea]. Available: <https://www.paypal.com/es/home>. [Último acceso: 21 11 2018].
- [39] «Pay-Pal,» [En línea]. Available: <https://www.braintreepayments.com/es?locale=es>. [Último acceso: 21 11 2018].
- [40] «Stripe,» [En línea]. Available: <https://stripe.com/es>. [Último acceso: 21 11 2018].
- [41] «Stripe,» [En línea]. Available: <https://status.stripe.com/>. [Último acceso: 21 11 2018].
- [42] «Stripe,» [En línea]. Available: <https://stripe.com/docs/api>. [Último acceso: 21 11 2018].
- [43] «Stripe,» [En línea]. Available: <https://stripe.com/open-source>. [Último acceso: 21 11 2018].
- [44] «AWS,» [En línea]. Available: https://aws.amazon.com/es/free/?sc_channel=PS&sc_campaign=acquisition_ES&sc_publisher=google&sc_medium=ACQ-P%7CPS-GO%7CBrand%7CDesktop%7CSU%7CCore%7CCore%7CES%7CEN%7CText&sc_content=Brand_Core_HV_e&sc_detail=aws&sc_category=Core&sc_segment=293642817806&sc. [Último acceso: 21 11 2018].
- [45] «Google Cloud,» [En línea]. Available: https://cloud.google.com/gcp/?hl=es&utm_source=google&utm_medium=cpc&utm_campaign=emea-es-all-es-dr-bkws-all-all-trial-e-gcp-1003963&utm_content=text-ad-none-any-DEV_c-CRE_253514829485-ADGP_Hybrid%20%7C%20AW%20SEM%20%7C%20BKWS%20~%20EXA_1:1_ES_ES_General_. [Último acceso: 21 11 2018].
- [46] «Microsoft Azure,» [En línea]. Available: https://azure.microsoft.com/es-es/free/search/?&OCID=AID719820_SEM_YwgWH4Ay&Inkd=Google_Azure_Brand&dclid=CjkKEQiAxNnfBRDy77KlI8z84uABEiQAtletMo9i3Y3NnEb6FkwJHVbJkqmWjPD8GHM2vol-cCfFkO-Lw_wcB. [Último acceso: 21 11 2018].
- [47] «codigos Qr,» [En línea]. Available: <http://www.codigos-qr.com/>. [Último acceso: 21 11 2018].
- [48] «Apache,» [En línea]. Available: <https://cordova.apache.org/>. [Último acceso: 21 11 2018].
- [49] P. Sanil, «Slideshare,» 1 8 2017. [En línea]. Available: <https://www.slideshare.net/PremSanil/deploy-web-app-heroku>. [Último acceso: 21 11 2018].
- [50] A. B. Narasimhan, «Slideshare,» 31 5 2012. [En línea]. Available: <https://www.slideshare.net/AnandBN/social-ent-with-java-on-heroku>. [Último acceso: 21 11 2018].
- [51] J. M. Alarcón, «campusmvp,» 19 1 2015. [En línea]. Available: <https://www.campusmvp.es/recursos/post/Que-es-el-stack-MEAN-y-como-escoger-el-mejor-para-ti.aspx>. [Último acceso: 21 11 2018].