



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Aplicación de reserva de hoteles

Hotels reservations app

Ernesto Echeverría González

La Laguna, 22 de marzo de 2019

D. **Alejandro Pérez Nava**, con N.I.F. 43821179S profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

D. **Fernando Pérez Nava**, con N.I.F. 42091420V profesor Titular de Universidad adscrito al Departamento de Estadística, Investigación Operativa y Computación de la Universidad de La Laguna, como cotutor

C E R T I F I C A (N)

Que la presente memoria titulada:

“Aplicación de reserva de hoteles”

ha sido realizada bajo su dirección por D. **Ernesto Echeverría González**, con N.I.F. 54064117H.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 22 de marzo de 2019

Agradecimientos

Este trabajo no habría sido posible sin mis padres, todos los profesores por cuyas manos he pasado, ni la ayuda de mis compañeros. También quiero agradecer a mi pareja, ya que si no fuera por ella, no estaría aquí en este momento.

Licencia



© Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.

Resumen

El objetivo de este trabajo es la creación de una aplicación móvil que permita, por parte del usuario, la búsqueda y reserva de hoteles. Sin embargo, como elemento innovador se ha decidido que la elección (y por tanto demanda) sea realizada por el usuario en lugar de por los hoteles. De este modo se promueve una mayor competencia entre los diferentes hoteles y el usuario no debe consumir un tiempo del que puede no disponer en la selección del mismo.

Para ello, también es necesaria una plataforma en la que los hoteles puedan inscribirse y “pujar” por los posibles clientes. Dado que la mayor parte de hoteles cuentan con un ordenador para los trámites internos (y no tanto con un smartphone o tablet), se ha decidido crear una aplicación web en lugar de una app para dicha plataforma. De este modo los hoteles podrán acceder incluso si no cuentan con un ordenador pero se favorecerá el uso de éste.

Palabras clave: hoteles, app, reservas

Abstract

This work is meant to create a mobile app which allows its users to search and compare hotels as well as to create a reservation once selected. As an innovation matter, the user shall be the one responsible to create a post that informs of the necessities. Afterwards, the hotels shall compete each other to get the lead. This way, the user will not have to waste time on comparing hotels, but those which fulfil the necessities and are interested in the client will post their response.

In order to do that, two platforms are needed: the mobile application itself and another application in which the hotels shall create an account and make bets for their leads. As many hotel managers work with personal computers instead of a smartphone or tablet, it has been decided to create a Web application which also will be accesible from such kind of devices.

Keywords: hotel, app, reservations

Índice general

Capítulo 1	Introducción.....	1
1.1	Antecedentes.....	1
1.2	Estado del arte.....	2
1.3	Marco sociocultural.....	2
1.4	Marco legal.....	2
Capítulo 2	Fases del desarrollo.....	4
2.1	Diseño.....	4
2.1.1	Diseño de la app móvil.....	4
2.1.2	Aplicación web.....	6
2.2	Diseño de datos.....	7
2.3	Programación.....	7
Capítulo 3	Tecnología aplicada.....	8
3.1	Base de datos.....	8
3.2	API Rest.....	9
3.3	Aplicación web.....	10
3.4	Aplicación móvil.....	11
Capítulo 4	Resultados.....	14
4.1	Hoteles.....	14
4.2	Aplicación móvil.....	16
Capítulo 5	Conclusiones y líneas futuras.....	25
Capítulo 6	Summary and Conclusions.....	26

Capítulo 7 Presupuesto.....27

Índice de figuras

Figura 2.1: Diseño de login en App.....	4
Figura 2.2: Diseño de registro en App.....	4
Figura 2.3: Diseño de pujas en App.....	5
Figura 2.4: Diseño de reservas en App.....	5
Figura 2.5: Diseño de buscador en App.....	5
Figura 2.6: Diseño de perfil en App.....	6
Figura 2.7: Diseño de login web.....	6
Figura 2.8: Diseño de registro web.....	6
Figura 2.9: Diseño de perfil de hotel web.....	7
Figura 2.10: Diseño de vista leads web.....	7
Figura 2.11: Diseño de pujas web.....	7
Figura 3.1: Modelo E/R.....	9
Figura 4.1: Vista login web.....	16
Figura 4.2: Vista registro web.....	16
Figura 4.3: Vista perfil de hotel web.....	17
Figura 4.4: Vista leads web.....	17
Figura 4.5: Vista open bets web.....	18
Figura 4.6: Vista closed bets web.....	18
Figura 4.7: Vista login App.....	19
Figura 4.8: Vista registro App.....	20

Figura 4.9: Vista reservas App.....	21
Figura 4.10: Vista búsqueda App.....	22
Figura 4.11: Vista bets App.....	23
Figura 4.12: Vista hotel App.....	24
Figura 4.13: Vista perfil App.....	25
Figura 4.14: Vista modificar perfil App.....	26
Figura 7.1: Gráficas de pérdidas mensuales atribuibles a la compra.....	30

Índice de tablas

Tabla 3.1: Diseño de API Rest.....	10
Tabla 3.2: Componentes web.....	12
Tabla 3.3: Servicios web.....	12
Tabla 3.4: Servicios de App.....	14
Tabla 7.1: Modelos de presupuesto.....	29

Capítulo 1

Introducción

En los últimos años, el crecimiento de la tecnología ha permitido al sector turístico ampliar sus ingresos gracias al marketing electrónico. Un caso muy notorio ha sido el de las aplicaciones móviles y webs de búsqueda, que permiten a un usuario decidir dónde desean alojarse en sus vacaciones o viajes de trabajo. En este entorno se sitúa este trabajo, pero con la intención de ofrecer una visión innovadora al paradigma.

En las aplicaciones tradicionales los hoteles publican sus ofertas y son los usuarios los que han de buscar, comparar y seleccionar la que más se adecue a sus necesidades. También existen aplicaciones que se sitúan en ese punto intermedio entre el hotel y el cliente y compara de forma automática varias ofertas publicadas en diferentes portales, pero no dejan de ser un intermediario que, en un caso ideal, no debería ser necesario.

El propósito de este trabajo es invertir las tornas, que sea el cliente el que especifique qué producto desea, evitando de este modo sobrecostos, y los hoteles compitan por el lead con el fin de que aquellos clientes que no dispongan del tiempo necesario para buscar y comparar hoteles puedan también hospedarse y eliminar sobre los hoteles las imposiciones de precios dada por otros portales. Este último punto viene a cumplirse debido a la posibilidad de dar una oferta mucho más centrada en las necesidades del usuario, sin características extras de una “oferta empaquetada” y sin la necesidad de generalizar ofertas a todos los clientes, por lo que el hotel, frente a un determinado cliente, puede desarrollar una estrategia mucho más agresiva. Además, el carácter privado de la oferta mantiene de forma segura el anonimato de la estrategia tomada por cada hotel entre ellos, provocando más competitividad por la incertidumbre, y ante la plataforma, por lo que se le impide cualquier tipo de actuación posible.

1.1 Antecedentes

La reserva de hoteles ha sido un servicio ampliamente requerido desde la creación de los mismos a finales de la primera década de 1700, durante el Periodo Colonial. Sin embargo, durante la Edad Media ya existían lugares de hospedaje para comerciantes y mercaderes. En un principio, la reserva de habitación en estos lugares se realizaba de forma presencial, pero planteaba un gran problema, ya que si el cliente no se encontraba cerca, no tenía la posibilidad de comprobar la disponibilidad. Esto se solucionó de forma rápida mediante el correo postal primero y el teléfono después.

Un tiempo después, en 1841, Thomas Cook crea la primera agencia de viajes y desde entonces los clientes dejan en sus manos la preparación del trayecto, actividades y reserva de estancia; sin embargo, esto incurre en un gran coste para el cliente. A pesar de ello, durante varios años es el método más utilizado.

Tras la globalización de Internet, se han creado varias empresas basadas en un portal online en el que los hoteles se publicitan y permiten hacer reservas desde la misma página web. Varios de estos portales obligan al hotel a poner el precio más bajo posible para cada oferta y ahora se cambia de nuevo el problema: los costes se abaratan pero el usuario de nuevo debe utilizar su tiempo para comprobar cuál de estas ofertas hoteleras satisfacen en mayor medida sus necesidades y posibilidades.

1.2 Estado del arte

Para estudiar el estado del arte tenemos que diferenciar entre el servicio de búsqueda y reserva de hoteles brindado a los clientes y el ofrecido a los hoteles al facilitar campañas de marketing y la relación con clientes previos.

En el primer caso, un gran número de empresas online se han ido consolidando a lo largo de los años utilizando la misma tecnología y prácticamente el mismo tipo de contrato. Todas ellas se fundamentan en una página web enlazada a una base de datos que registra ofertas de los diferentes hoteles que hayan creado una cuenta en ella. Para la creación de la cuenta es necesario firmar unos términos y condiciones que, como ya hemos nombrado anteriormente, suponen una gran presión para los hoteles. Algunas de estas páginas también piden exclusividad, por lo que, para facilitar el trabajo al usuario final, existen páginas web que comparan entre varias de las páginas anteriores por medio de APIs y agentes.

En el segundo caso, el SoA (State of Art o estado del arte) sería un CRM, Customer Relation Manager, un tipo de software que en la actualidad intenta integrar herramientas como el Big Data y la inteligencia artificial para segmentar a los clientes y aplicar marketing inteligente. En este caso, el referente en tecnología es Microsoft Dynamics, pero por su precio el más utilizado es, en su lugar Salesforce.

1.3 Marco sociocultural

Este proyecto nace de la necesidad de la sociedad moderna de conseguir un hospedaje cómodo mientras se encuentra de viaje, ya sea de vacaciones o por trabajo. En los últimos años ambos casos han aumentado drásticamente debido principalmente a las políticas laborales, que aseguran un periodo vacacional mínimo; aunque también en menor medida a la globalización de las empresas que han llegado a hacer necesario que algunos empleados tengan que viajar por la empresa. Sin embargo, el ritmo frenético de la vida moderna limita el tiempo disponible de los posibles clientes de los hoteles.

1.4 Marco legal

El ámbito legal de este tipo de aplicaciones está vastamente estudiado por el mercado: principalmente este tipo de aplicaciones debe tener en cuenta el Reglamento General de Protección de Datos (RGPD) y establecer un claro contrato tanto con el usuario como con el hotel. Cada uno de estos contratos deben ser mostrados en los Términos y Condiciones de Uso de la plataforma adecuada al medio; esto es, si la plataforma está destinada a los hoteles, los términos y condiciones de uso relativos a los hoteles deberán ser mostrados en esta así como con los usuarios en su respectiva plataforma.

Muchas de estas aplicaciones incurren también en publicidad en la misma página, por lo que añaden a sus términos y condiciones el permiso por parte del usuario de registrar sus accesos con herramientas analíticas propias del marketing inteligente tales como Google Analytics, Google Tag Manager, eTracker, Hotjar, Taboola, Intent Media o Radar. El uso de complementos de reedes sociales también puede ser utilizado para recopilar aún más información del usuario, especialmente Twitter, Facebook, Instagram y Youtube. El usuario además, en estos casos, puede o debe (dependiendo de la plataforma) permitir que se le envíen correos de publicidad y visualizar la misma dentro de la página.

Capítulo 2

Fases del desarrollo

En este capítulo especificaremos punto a punto cómo será el transcurso del desarrollo sin profundizar en tecnologías y utilizando

2.1 Diseño

El primer paso para desarrollar el proyecto se basa en analizar las necesidades de los usuarios y hoteles y desarrollar vistas que permitan a estos satisfacer sus necesidades de forma sencilla. El diseño de ambas plataformas tenderá a ser oscuro, con colores blancos y negros para mostrar seriedad y compromiso, pero no completamente blanco y negro (#FFFFFF y #000000) para no molestar al ojo.

2.1.1 Diseño de la app móvil



Figura 2.1:
Diseño de login
en App



Figura 2.2:
Diseño de
registro en App

En las vistas anteriores tenemos un rápido boceto del login y el registro. En ambos casos se incluye, dentro de una carta, el título y un par de campos de texto seguidos de un botón. El diseño es ampliamente utilizado, por lo que ningún usuario tendrá problemas para entender qué debe hacer.

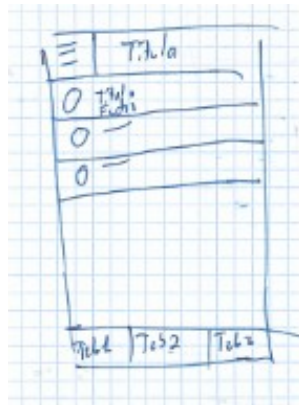


Figura 2.4: Diseño de reservas en App



Figura 2.3: Diseño de pujas en App

Posteriormente accedería a una lista clickable con sus reservas y las respectivas pujas. La lista de puja será visualizada tras clickar sobre la reserva y desde ella se podrá seleccionar el hotel.



Figura 2.5:
Diseño de
buscador en
App

Para añadir una búsqueda, el usuario deberá acceder a una vista que le permita introducir un nombre identificador, fechas de entrada y salida y características mínimas. Todo ello debe ser lo suficientemente descriptivo, ya que este es el núcleo de la aplicación.

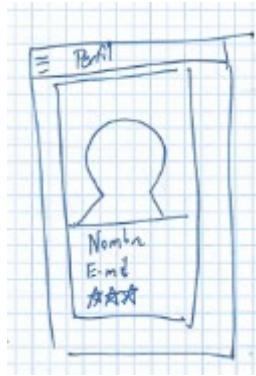


Figura 2.6:
Diseño de perfil
en App

También permitimos al usuario modificar su perfil, así que es necesaria una vista para representarla. Hemos elegido una carta con una imagen como cabecera.

2.1.2 Aplicación web

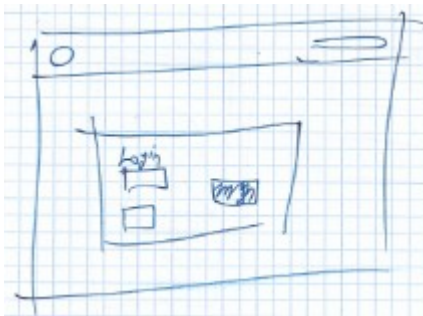


Figura 2.7: Diseño de login
web

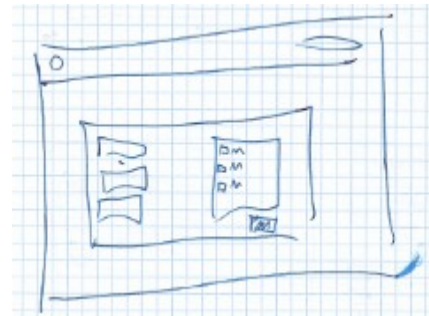


Figura 2.8: Diseño de registro
web

Al igual que en la aplicación móvil, será necesario un registro y un login, pero en esta ocasión el registro ha de añadir la información del hotel. Por ello dejaremos una pequeña caja a rellenar con estos datos.

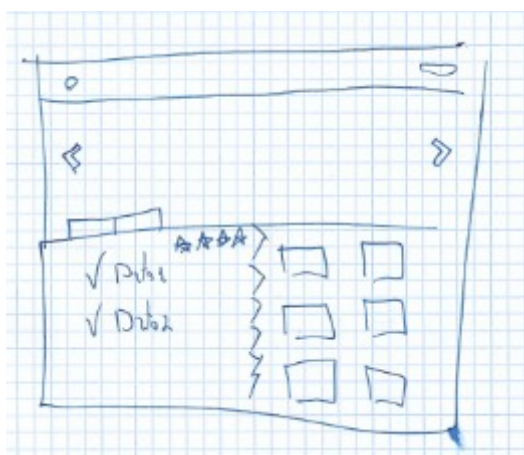


Figura 2.9: Diseño de perfil de hotel web

Una vez identificado, el hotel podrá ver una vista de perfil, visible en modo móvil para la aplicación, en la que se represente de forma visual la información del hotel y algunas fotos del mismo, siguiendo el estilo genérico de las webs de hoteles.

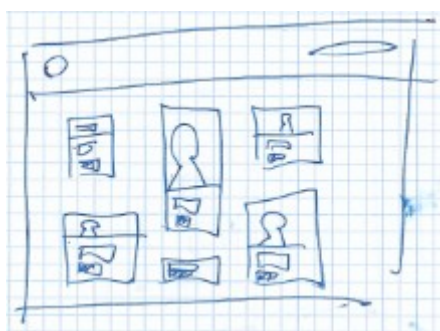


Figura 2.10: Diseño de vista leads web

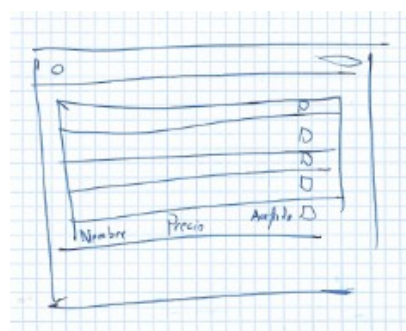


Figura 2.11: Diseño de pujas web

El hotel después deberá tener, de forma igual de visual, una lista de los posibles clientes a los que ofertar y otra, menos visual con las ofertas realizadas.

2.2 Diseño de datos

Para almacenar toda la información necesaria para ambas aplicaciones, primero hemos de identificar los objetos básicos de éstas. Es innegable que los usuarios y hoteles que han de registrarse son dos objetos totalmente necesarios. Por tanto, profundicemos en ellos:

- El usuario: Posee un nombre identificativo en la plataforma, así como una contraseña. Este nombre indicativo puede ser el correo electrónico y, por sencillez para el usuario, utilizaremos éste. Además, posee un nombre y apellidos reales que figurarán en la reserva del hotel. También hemos decidido darle una imagen y, para terminar de conectarlo con el hotel, pediremos también su teléfono. Estos datos

serán nuestra base, pero siempre será ampliable.

- El hotel: También posee un correo electrónico y una contraseña para identificarse en la plataforma. Todo hotel tiene un nombre propio, una localización, un número de estrellas y ciertas características nombrables, como si tiene piscina o no, spa, caja fuerte, calefacción, aire acondicionado,... Durante el desarrollo real, tras seleccionar la tecnología, crearemos una lista inicial, que podrá ser ampliada en producción.

Para comunicar ambos objetos estudiados hemos de añadir el pivote en el que se centra la aplicación: la búsqueda.

- La búsqueda: Tiene un nombre identificativo, un usuario, una fecha y unos requisitos mínimos a cumplir. También debemos discernir si está abierta o no, esto es, si hay un hotel al que se le haya asignado o no. Sin embargo, este objeto no nos da la información de una respuesta del hotel, por lo que crearemos el siguiente objeto.
- La puja: La puja almacena una referencia a la búsqueda a la que se refiere, el hotel que la ha realizado y la oferta en euros.

2.3 Programación

Para explicar la programación es necesario antes seleccionar la tecnología a utilizar. Es por esto que este punto será tratado en el capítulo 3.

Capítulo 3

Tecnología aplicada

Como hemos nombrado anteriormente, el proyecto se centra en una aplicación móvil y otra web. Por tanto, necesitamos alguna tecnología que haga ambos entornos compatibles. Esta tarea no es complicada siempre que utilicemos una API Rest en un servidor con una base de datos compartida entre ambas aplicaciones. Sin embargo podremos facilitar el desarrollo si utilizamos tecnologías con una base en común. Por ello hemos decidido utilizar el MEAN Stack junto con Ionic 4, todo ello basado en JavaScript.

3.1 Base de datos

Para almacenar toda la información necesaria, el MEAN Stack nos ofrece una herramienta: una base de datos no relacional documental llamada MongoDB. En este Gestor de Base de Datos no se almacenan tablas sino colecciones, puesto que el concepto de columna no existe; asimismo, cada entrada es un objeto en formato BSON, un derivado del JSON (JavaScript Object Notation). Sin embargo, aunque Mongo no es un gestor relacional permite establecer relaciones entre sus colecciones. Por ello, y porque favorece nuestro caso particular, diseñaremos una base de datos basada en relaciones.

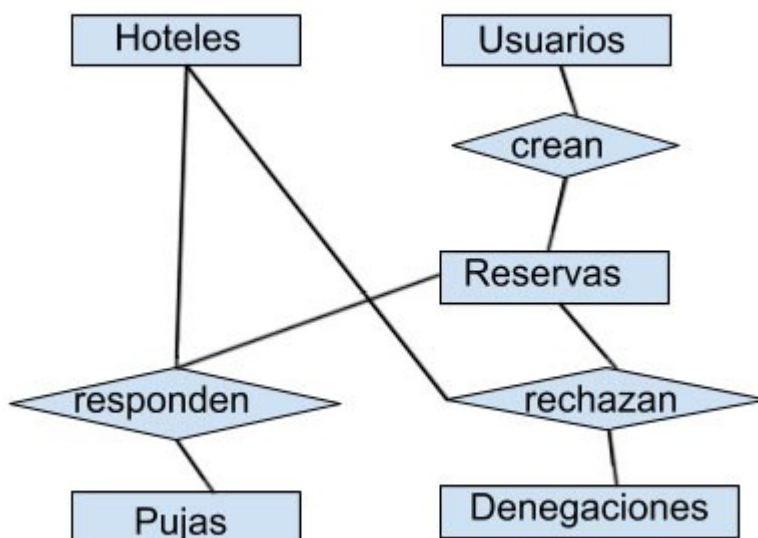


Figura 3.1: Modelo E/R

3.2 API Rest

Para conectar ambas aplicaciones entre sí y con la base de datos de MongoDB, el MEAN Stack nos ofrece la posibilidad de crear un API Rest utilizando Express y NodeJS. Esta tecnología nos permite crear un servidor al que realizar llamadas GET, POST, PUT, DELETE y OPTIONS para insertar, modificar, eliminar u obtener información. Nuestra API Rest se divide en varias rutas que dirigen a un controlador cada uno que, a su vez, define varias funciones como describe la siguiente tabla.

Tabla 3.1: Diseño de API Rest

Ruta	Controlador	URL	Método
Bets	Bets	/register	POST
		/getopenbyreservation	POST
		/ourbets	POST
		/highest	POST
		/remove	POST
		/updateprice	POST
		/getopenbyreservation	POST
		/close	POST
		/getclosedbyreservation	POST
		/findtoclose	POST
ClosedBets	ClosedBets	/register	POST
		/getbyhotel	POST
Denegated	Denegated	/register	POST
Hotels	Hotels	/stars	POST
		/register	POST
		/login	POST
		/profile	POST

		/publicprofile	POST
		/data	POST
		/getimage/:image	GET
		/setimage	POST
Reservations	Reservations	/reserve	POST
		/availables	POST
		/myreservations	POST
		/getreservation	POST
Resolutions	Resolutions	/register	POST
Users	Users	/stars	POST
		/register	POST
		/login	POST
		/update	POST
		/profile	POST
		/publicprofile	POST
		/getimage/:image	GET
		/test	GET

3.3 Aplicación web

Actualmente existe un gran número de herramientas que permiten la implementación de una aplicación web basadas en JavaScript, aunque las más conocidas son AngularJS, Angular 2+ y VueJS. Sin embargo, la recomendada por el Stack MEAN es Angular 2+.

Angular 2+, también llamado Angular a secas, es un framework que utiliza Typescript, un lenguaje basado en Javascript pero que utiliza el paradigma de programación orientada a objetos, para su desarrollo. Este paradigma guía por completo el funcionamiento de

Angular, que plantea crear un componente mediante tres ficheros; un CSS, un TypeScript y un HTML. De este modo los componentes creados pueden ser reutilizados en cualquier página de la aplicación. Así mismo, cada página compone a su vez otro componente que puede incluir a los anteriores. Por ello, hemos de diferenciar entre componentes internos a una página y componentes como página. Nuestra aplicación posee los componentes utilizados como página descritos en la siguiente figura.

Tabla 3.2: Componentes web

Componente	Función
home	Página de inicio, se encarga de redireccionar tras comprobar si el usuario* está loggeado o no.
login	Permite al usuario* identificarse dentro de la aplicación
register	Proporciona al posible futuro usuario una herramienta para crear una cuenta a la que acceder posteriormente.
hotel-profile	Ofrece una vista previa y la posibilidad de personalizar el perfil del usuario*.
leads	Muestra una lista de los posibles clientes del hotel, es decir, aquellos que hayan realizado búsquedas cuyo perfil coincida con el del usuario*.
bets	Lista las pujas realizadas por el usuario* sobre búsquedas de posibles clientes.
clients	Nombra aquellos clientes que ya han seleccionado al usuario*.

*El usuario representa en este caso al hotel inscrito en la aplicación web

Para comunicarse con la API Rest creada en NodeJS Angular nos utiliza un -tipo de clase especial llamada servicio, que no es de forma funcional sino un conjunto de métodos que realizan llamadas asíncronas al servidor con formato de promesa para que los componentes puedan suscribirse a dicha llamada. Los servicios se corresponden con las llamadas como sigue:

Tabla 3.3: Servicios web

Servicio	Método del servicio	Función de la API Rest
BetsService	create	competition/register

	highest	competition/highest
	remove	competition/remove
	updatePrice	competition/updatePrice
	closedByHotel	assigned/getbyhotel
DenegationsService	create	/register
HotelsService	register	hotels/register
	getData	hotels/profile
	login	hotels/login
	leads	reservations/availables
	betsByReservation	competition/getopenbyreservation
	myBets	competition/ourbets
	setImage	hotels/setimage
ReservationsService	reservationById	reservations/getreservation
UsersService	publicProfile	people/publicprofile

3.4 Aplicación móvil

La aplicación móvil se planteó realizar en su inicio en Ionic 3, pero tras un tiempo, y cuando la aplicación estaba creada, un par de bugs solucionados por un entonces Ionic 4 en beta afectaban a la aplicación. Es por esto que se decidió volver a crear la aplicación móvil desde cero; sin embargo, debido a que como hemos dicho Ionic 4 estaba en beta, la documentación era muy escasa y deficiente, por lo que el desarrollo se trató, en gran medida, de un trabajo de ensayo y error.

El funcionamiento de Ionic es bastante similar al de Angular salvo por la existencia de algunos módulos específicos que adaptan el código al funcionamiento de un dispositivo móvil para dejarnos acceder a cámara, GPS, etcétera. Los componentes que hacen las veces de página se llaman, en este caso, páginas y no componentes, por lo que el término de componentes se limita a aquellos que no hacen las veces de una pantalla de la aplicación.

En este caso, las páginas desarrolladas han sido:

- **Home**. Permite visualizar el estado de las pujas divididas en cuatro categorías. **Sin respuesta** segmenta aquellas búsquedas creadas por el usuario que no han tenido respuesta por parte de ningún hotel. **En decisión** muestra aquellas búsquedas que han sido respondidas por al menos un hotel pero para las que no el usuario no ha elegido en cuál de ellos hospedarse; al pulsar sobre cualquiera de ellas se abrirá la página modal List explicada más adelante. **Reservados** separa aquellas pujas que han sido respondidas por el usuario, esto es, donde ha decidido en qué hotel hospedarse, pero cuya fecha no ha finalizado. **Finalizados** nos muestra una lista de aquellos hoteles que ya han sido visitados para futuras referencias.
- **Hotel**. Muestra el perfil público de el hotel sobre el que clicke el usuario. De este modo permitimos que pueda discernir entre varios hoteles que respondan a su búsqueda. El perfil mostrado será personalizado desde la plataforma para los hoteles realizada en Angular.
- **List**. Esta página modal se abre al clickar sobre una de las búsquedas en decisión y muestra una lista de los hoteles que hayan respondido a la búsqueda en cuestión con sus respectivas ofertas. Desde esta vista el usuario puede clickar en un hotel y visualizar el perfil ya explicado en la página Hotel.
- **Profile**. Aquí el usuario encuentra una previsualización de su perfil y el método para modificarlo. Este perfil será el que se le muestre al hotel desde la sección de leads de la plataforma desarrollada en Angular.
- **Search**. La vista search permite al usuario crear una nueva búsqueda mediante la introducción de un nombre, un lugar para hospedarse, una fecha y un conjunto de características mínimas.

La aplicación móvil también ha de contar con servicios como si de Angular se tratase para comunicarse con la API Rest de NodeJS. En este caso, los métodos son los que se definen en la siguiente tabla.

Tabla 3.4: Servicios de App

Servicio	Método	Endpoint
Bets	0	0
	0	0
	0	0
	0	0
	0	0
ClosedBets	0	0

Hotels	0	0
	0	0
People	0	0
	0	0
	0	0
	0	0
	0	0
Reservation	0	0
	0	0
	0	0
<hr/>		

Capítulo 4

Resultados

4.1 Hoteles

Cuando un hotel acceda por primera vez a la plataforma, se encontrará directamente ante la pantalla de login.



Figura 4.1: Vista login web

Como dicho hotel no tiene ninguna cuenta, el único paso posible será proceder a registrarse.

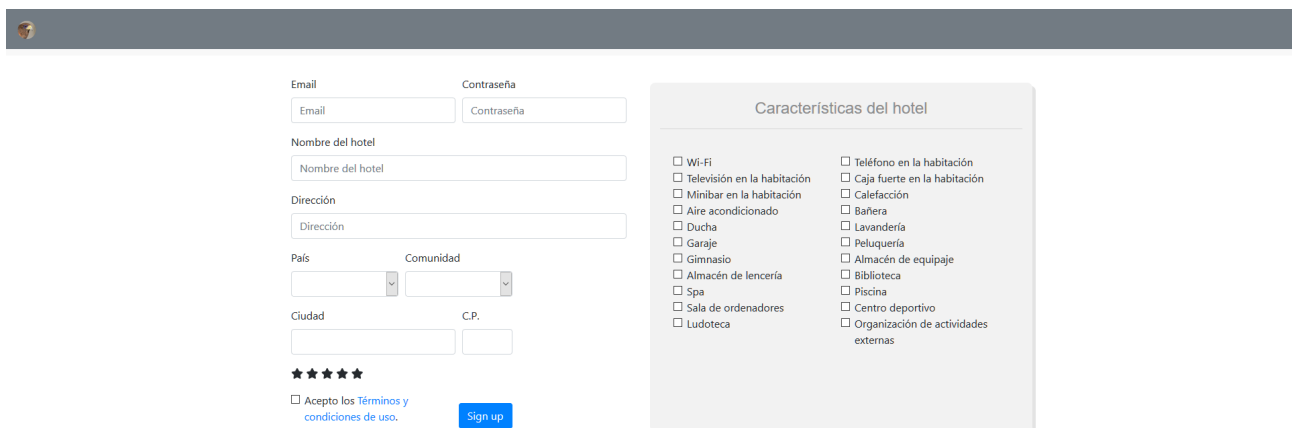
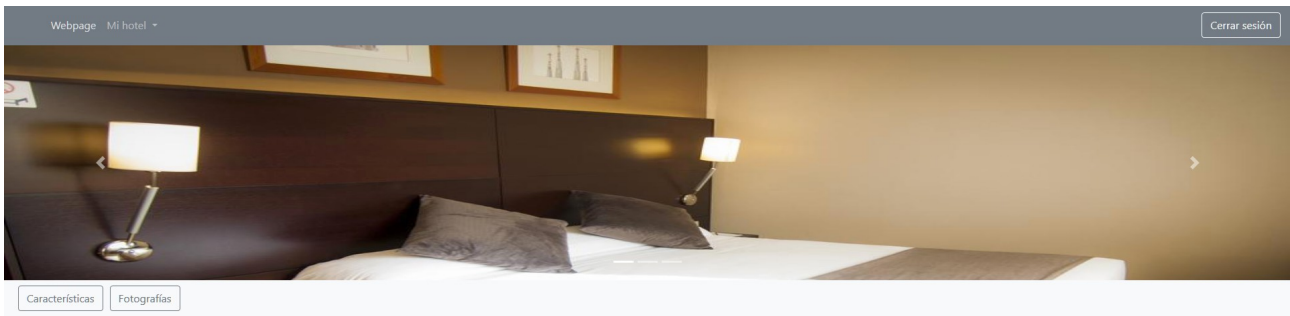


Figura 4.2: Vista registro web

Una vez el usuario se haya registrado, se redireccionará a la pantalla de login donde, ahora sí, podrá identificarse y acceder a la pantalla principal.



Terrassa Park

★★★★★

Servicios de habitación

- ✓ Aire acondicionado.
- ✓ Aseo con ducha.
- ✓ Minibar de pago.

Otros servicios

- ✓ Garaje.

Primera imagen de la presentación

Examinar... No se ha seleccionado ningún archivo.

Guardar

Segunda imagen de la presentación

Examinar... No se ha seleccionado ningún archivo.

Guardar

Tercera imagen de la presentación

Examinar... No se ha seleccionado ningún archivo.

Guardar

Imagen de la galería

Examinar... No se ha seleccionado ningún archivo.

Guardar

Añadir

Figura 4.3: Vista perfil de hotel web

Desde esta pantalla el hotel puede ver sus dotaciones así como sus imágenes. También podrá introducir nuevas fotos y cambiar las tres fotos principales. La primera de ellas representará al hotel y, entre las tres, se formará una marquesina visible desde el perfil en la aplicación móvil del usuario.

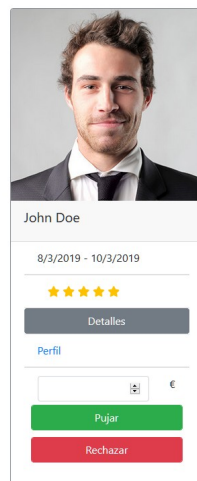
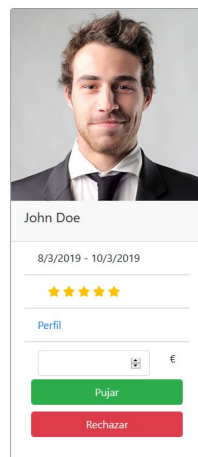


Figura 4.4: Vista leads web

Desde el menú se podrá acceder a la pestaña de leads. Aquí se encuentran todos los usuarios que hayan realizado una búsqueda cuyos requisitos mínimos sean afines al hotel. Cada búsqueda está representada en una tarjeta con la imagen del cliente, su nombre, un botón que permite visualizar las características pedidas por el cliente, un campo para introducir la puja, un botón de aceptación y otro de rechazo.



#	Nombre	Puntuación	Fechas	Puja	Mejor puja	Editar	Eliminar
1	John Doe	★★★★★	26/12/2019 - 29/12/2019	350€	200€		

Figura 4.5: Vista open bets web

Una vez la puja es realizada, se puede observar, eliminar y modificar desde la pestaña pujas; esto es, por supuesto, sólo posible si el usuario final no la ha aceptado previamente, caso en el que pasa a la siguiente sección y pestaña, clientes.



#	Nombre	Fechas	Puja	Telephone
1	John Doe	10/4/2019 - 12/4/2019	200€	123456789
2	John Doe	26/11/2019 - 29/11/2019	200€	123456789
3	John Doe	26/11/2019 - 29/11/2019	200€	123456789

Figura 4.6: Vista closed bets web

4.2 Aplicación móvil

El usuario, al igual que el hotel, se encontrará frente a una vista de login al abrir por primera vez la aplicación.

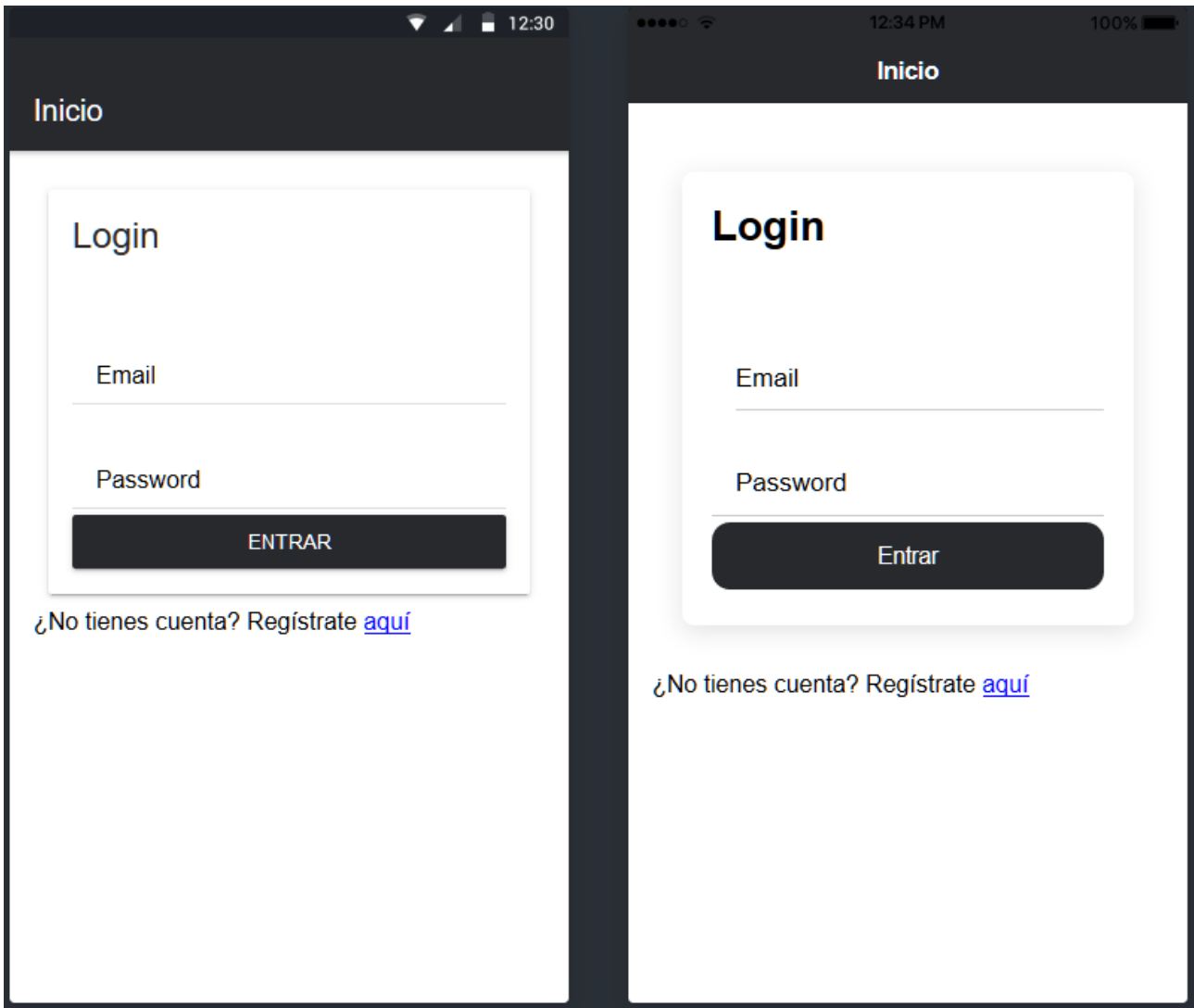


Figura 4.7: Vista login App

Y, al igual que ocurre con el hotel, su única opción inicial es registrarse para ser redireccionado al login.

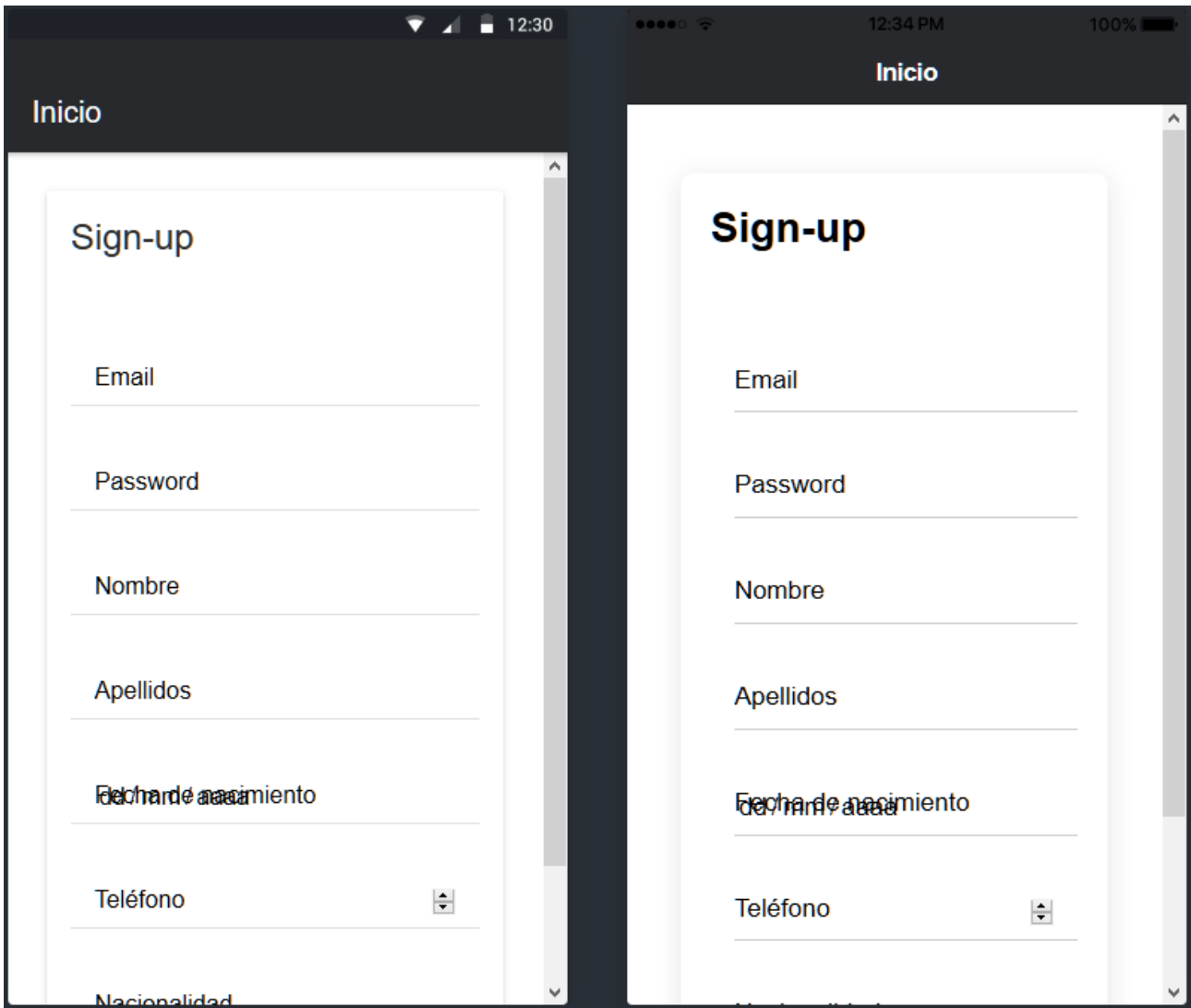


Figura 4.8: Vista registro App

Una vez loggeado, su vista inicial será la que contiene el registro de búsquedas y pujas, que en su inicio mostrará un mensaje explicando que está vacío debido a la ausencia de búsquedas e instando al usuario a crear una.

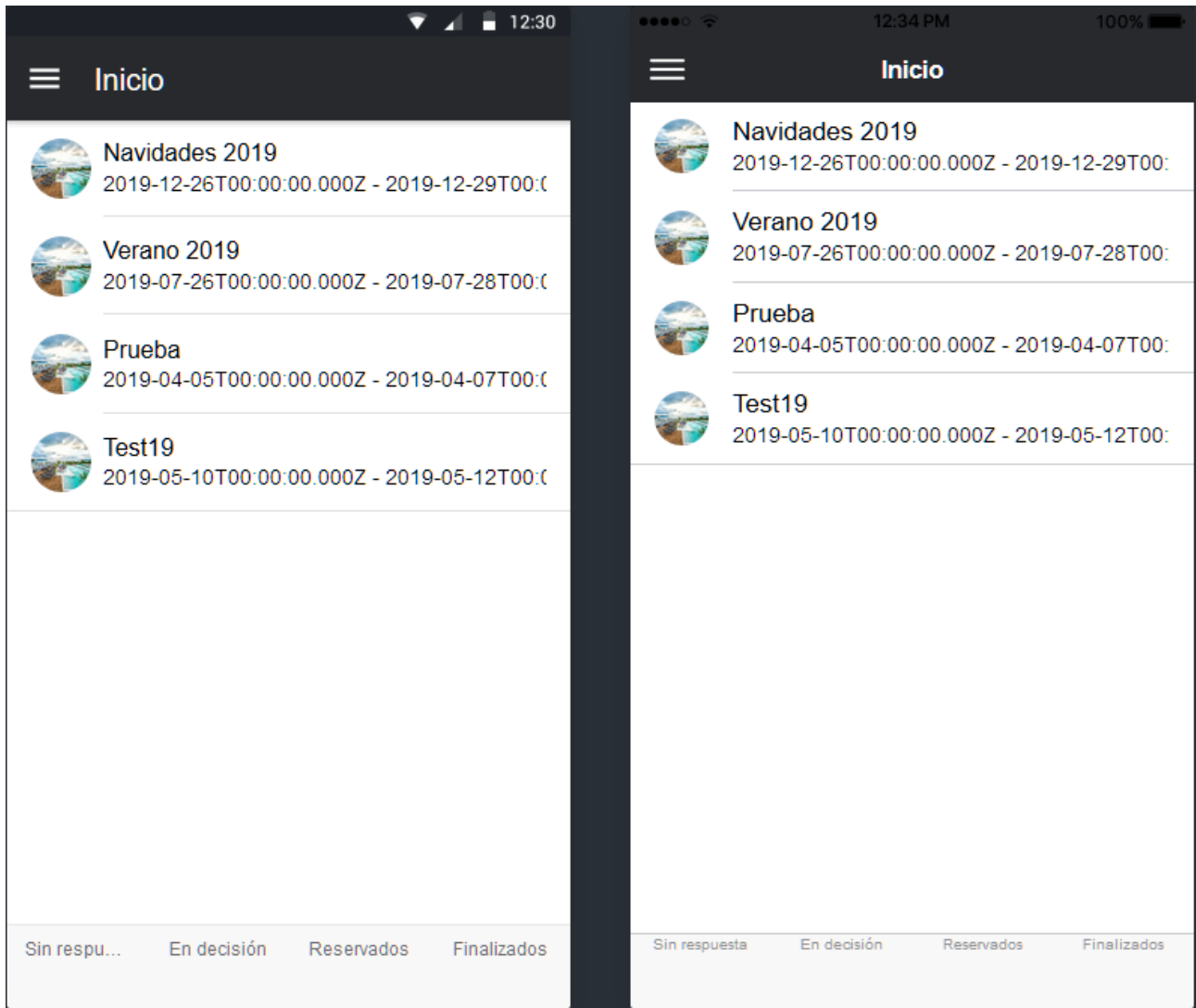


Figura 4.9: Vista reservas App

Si el usuario procede a realizar una búsqueda desde el menú, se encontrará con la vista adecuada. Al almacenar una búsqueda se mostrará un mensaje de confirmación.

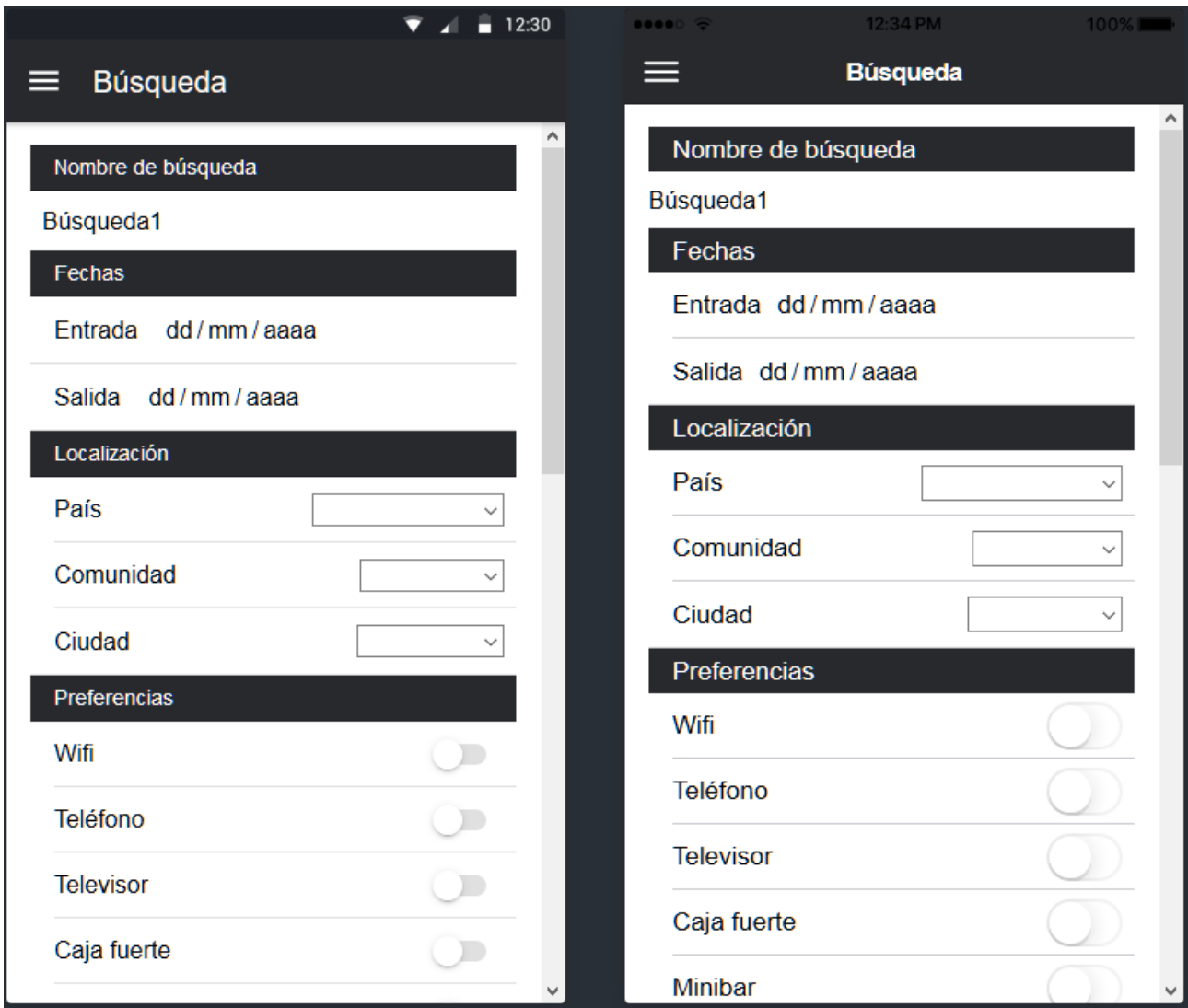


Figura 4.10: Vista búsqueda App

Tras realizar la búsqueda quedará esperar y comprobar de vez en cuando la pestaña En decisión. Cuando esta pestaña muestre una búsqueda, el usuario podrá pulsar sobre ella para observar una lista de pujas.

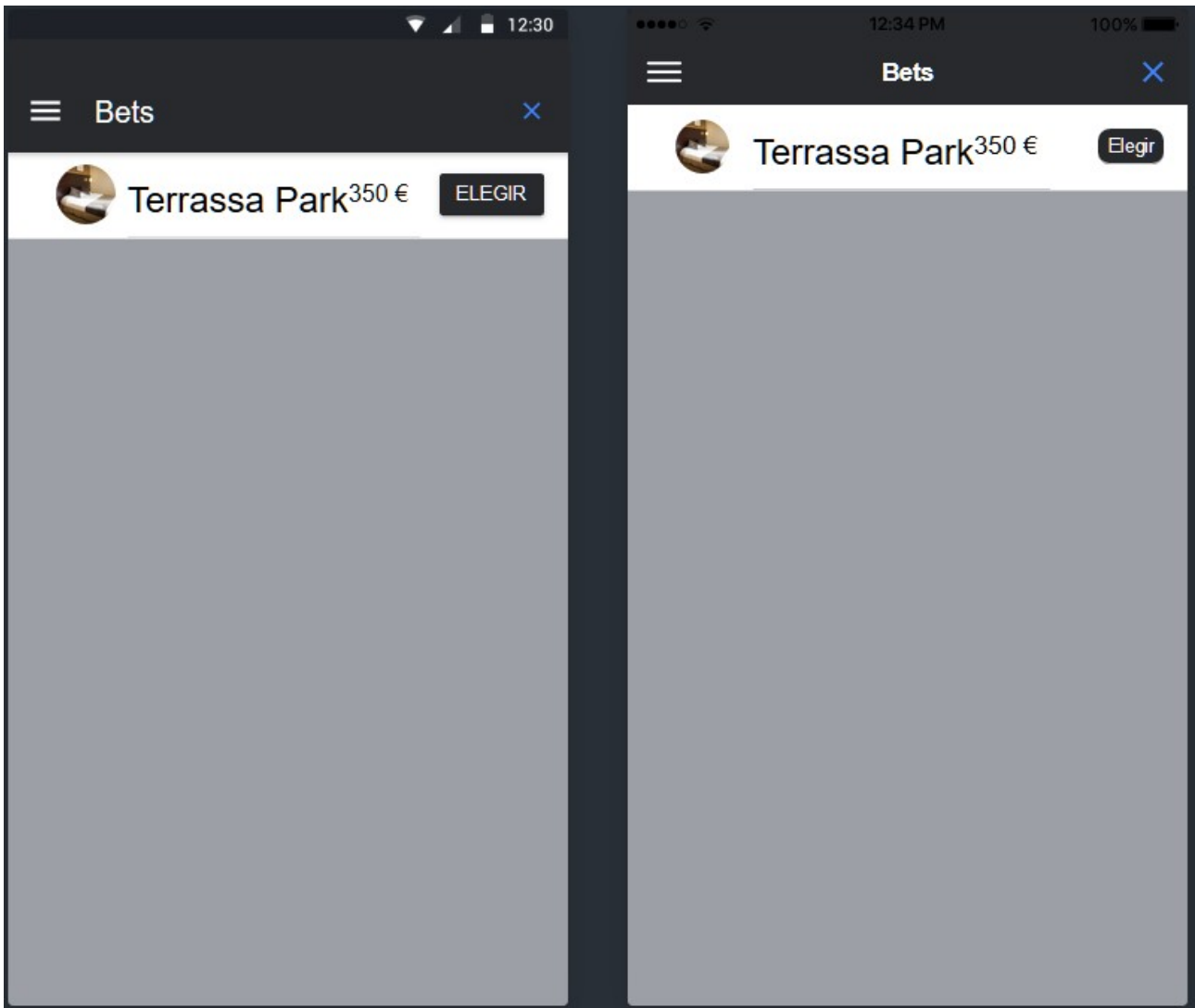


Figura 4.11: Vista bets App

Si, de nuevo, pulsa sobre la puja, podrá ver el perfil del hotel en versión móvil.

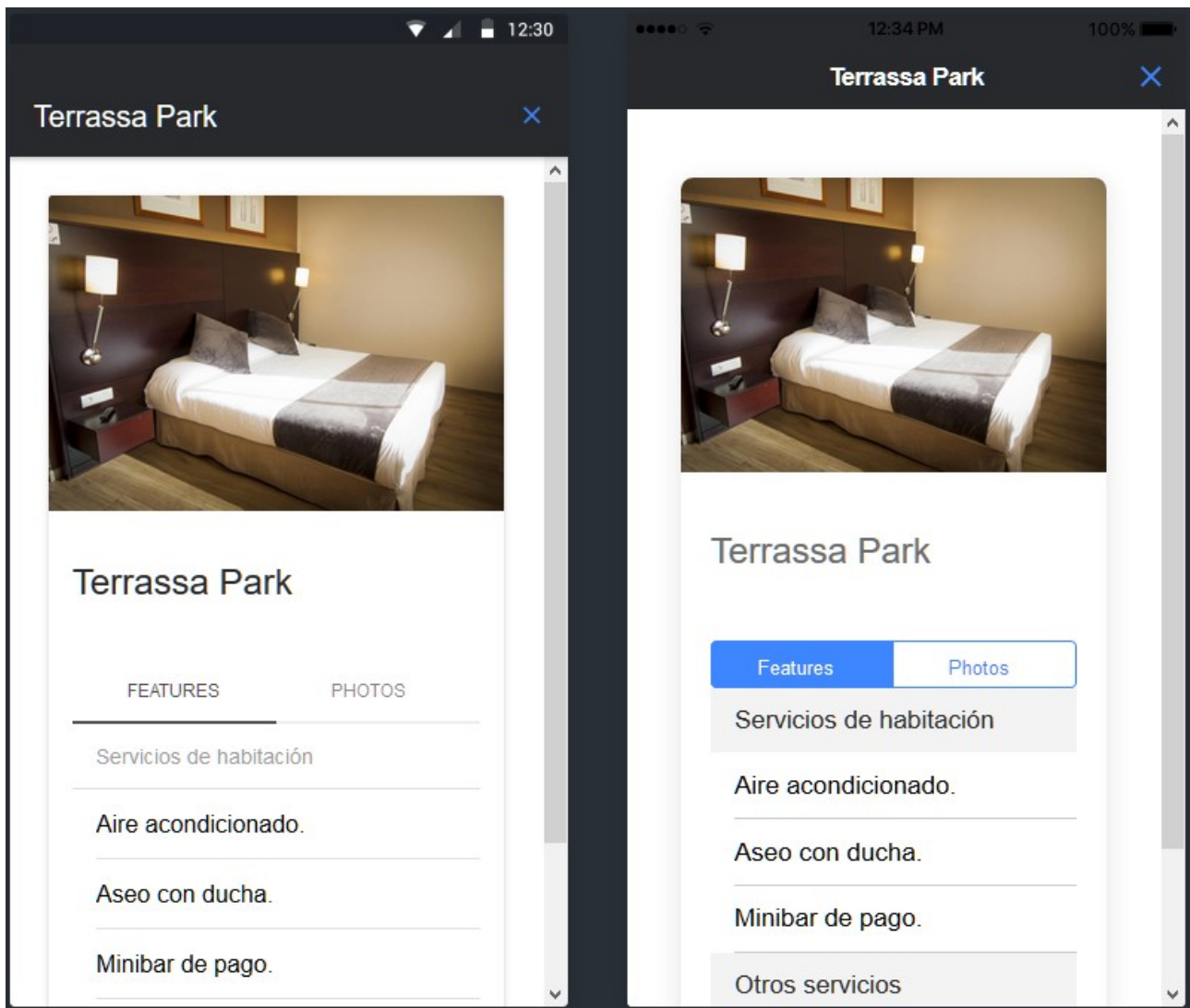


Figura 4.12: Vista hotel App

Al usuario sólo le queda por comprobar algo en la aplicación: su perfil. Si bien no es obligatorio para el proceso, es altamente recomendable que el usuario lo modifique, ya que será su carta de presentación frente al hotel.

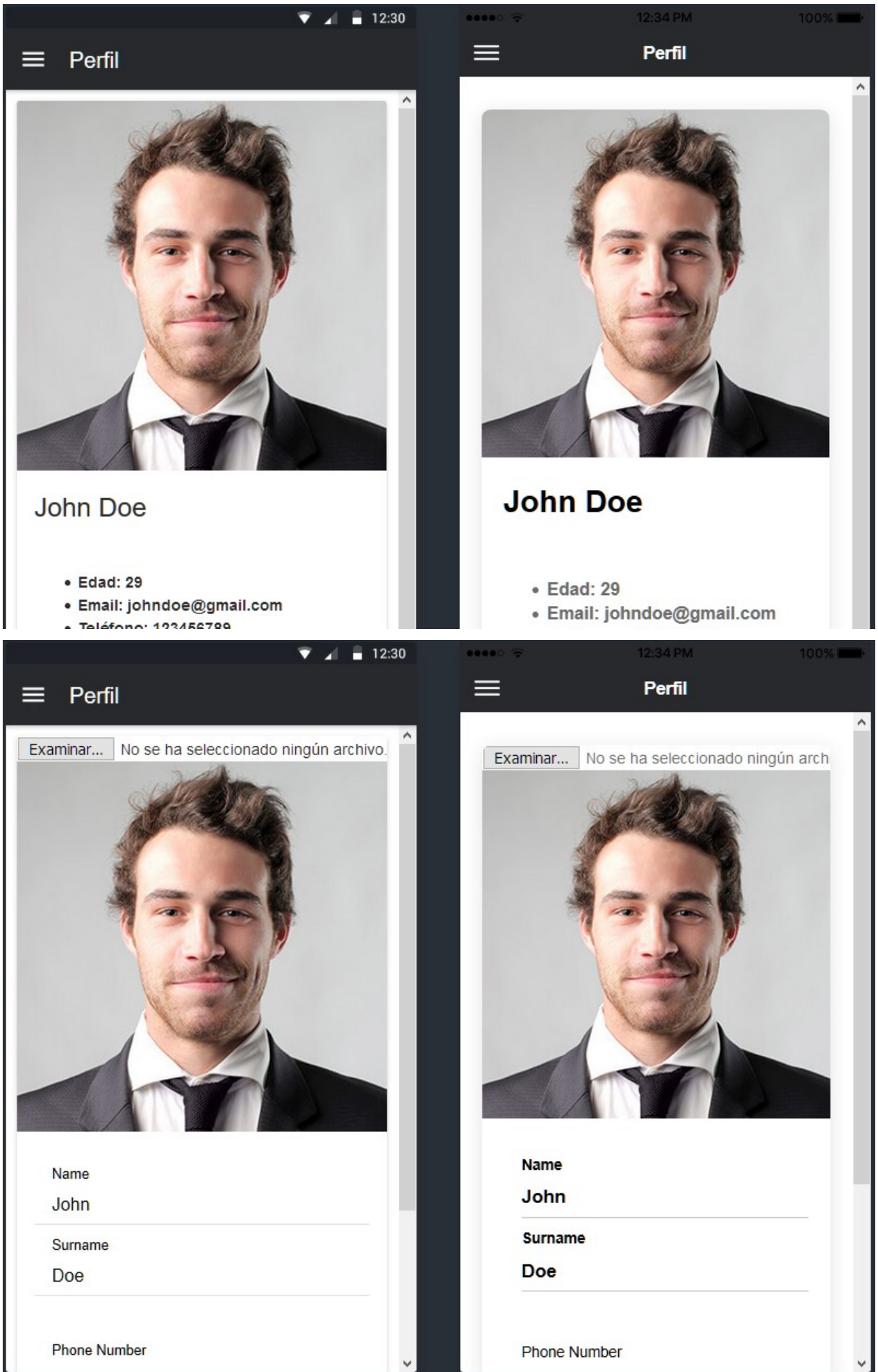


Figura 4.14: Vista modificar perfil App

Capítulo 5

Conclusiones y líneas futuras

Este trabajo ha sido una gran forma de expresar al máximo las posibilidades brindadas por la tecnología elegida. Sin embargo, el hecho de que Ionic 4 fuera una tecnología experimental durante el desarrollo dificultó la existencia de un resultado adecuado de acuerdo a sus patrones de diseño para algunas de las funciones requeridas. El proyecto por sí mismo no requería de una tecnología en exceso innovadora, sino que lo novedoso del proyecto se basa en el modelo de negocio. Por lo tanto, el verdadero reto era su adaptación y el cumplimiento de las bases de la aplicación especificadas durante el capítulo introductorio.

En un futuro, el proyecto es bastante ampliable. Por un lado, se puede hacer un mayor estudio de mercado y aumentar el número de características posibles sobre un hotel, por ejemplo, ofreciendo diferentes tipos de alimentación al usuario o con un mayor número de comercios en el hotel, sin limitarse a una peluquería como en su estado actual; por otro lado, se podría implementar un sistema de votos en el que tanto los usuarios como los hoteles voten uno al otro tras la estancia en cuestión de servicio y comportamiento respectivamente. De este modo también se favorece al cliente cuyo comportamiento sea destacable, ya que su valoración media será visualizable para los hoteles en el caso de los usuarios y viceversa. Para finalizar, el proyecto podría unirse al CRM del hotel, ya sea utilizando las APIs del proyecto como un módulo del que ya tengan instalado o ampliando los servicios de éste.

Capítulo 6

Summary and Conclusions

This work has been a great way to explore the possibilities that the technology chosen has to offer. However, the fact that Ionic 4 was in a beta state during the development of this project has made it more difficult since there was not a right way to implement certain tools needed for the work amongst the design pattern. The project nevertheless did not require an outstanding technology, but the innovation was based in the business level. This is why, the challenge lied in the adaptation of it and the achievement of the basis for the application as explained in the introduction chapter.

In the future, the project could get quite an advance. For example, a better business study could be done in order to increase the features that segment the hotels. A vote system could also be implemented, in which the users would evaluate the service of the hotels and they would evaluate the client's behaviour. This way, a well-mannered client would get more offers from different hotels. Lastly, this project could be turned into the hotel's CRM with enough work or get fused with the hotel's current CRM using the APIs created in the development of this work.

Capítulo 7

Presupuesto

Teniendo en cuenta que el trabajo se ha desarrollado a lo largo de 17 semanas, con 40 horas semanales, a una media de 36 euros la hora para un equipo de desarrolladores y 60 para el equipo de análisis; el trabajo quedaría valorado en 35000 €. Si añadimos los costes de mantenimiento del servidor, tenemos dos opciones, el alquiler de uno o la compra del mismo, como se muestra en la siguiente tabla.

Tabla 7.1: Modelos de presupuesto

Servidor propio	Servidor propio con backup	Hosting de azure
<ul style="list-style-type: none"> • Compra de servidor medio • Mantenimiento del DNS 	<ul style="list-style-type: none"> • Compra de dos servidores medios para garantizar disponibilidad • Rack • Mantenimiento del DNS 	<ul style="list-style-type: none"> • Alquiler de Azure
<p>1500 € pago único + 5 € anuales.</p> <p>* Descontar pérdidas por caídas y mantenimiento</p>	<p>3500 € pago único + 5 € anuales.</p> <p>*Descontar pérdidas por mantenimiento</p>	<p>1500€ mensuales</p>

** Todos los precios han sido redondeados tras hallar una media de las especificaciones tirando a la baja.

Recomendamos la opción de Hosting de Azure para un bajo número de hoteles al comenzar, ya que para tener mantenimiento se ha de contratar a un técnico que rondará el precio del mismo. Sin embargo, si la empresa ya cuenta con un técnico especializado en sistemas informáticos, el servidor propio con backup parece la mejor opción. Una vez tomada esta decisión, el precio final sería de 38000 euros de desembolso inicial para el servidor propio con backup. Si queremos iniciar una estrategia de marketing para recuperar dicho desembolso, la estrategia podría ser de cobro por hotel o por comisión de puja. En el primer caso, se aconsejaría un cobro de 150€ por hotel y en el segundo una comisión de 20€ por cliente. Según la estimación, durante los dos primeros años el proyecto estaría en

déficit económico, pero con un número suficiente de clientes generaría beneficios tras este periodo.

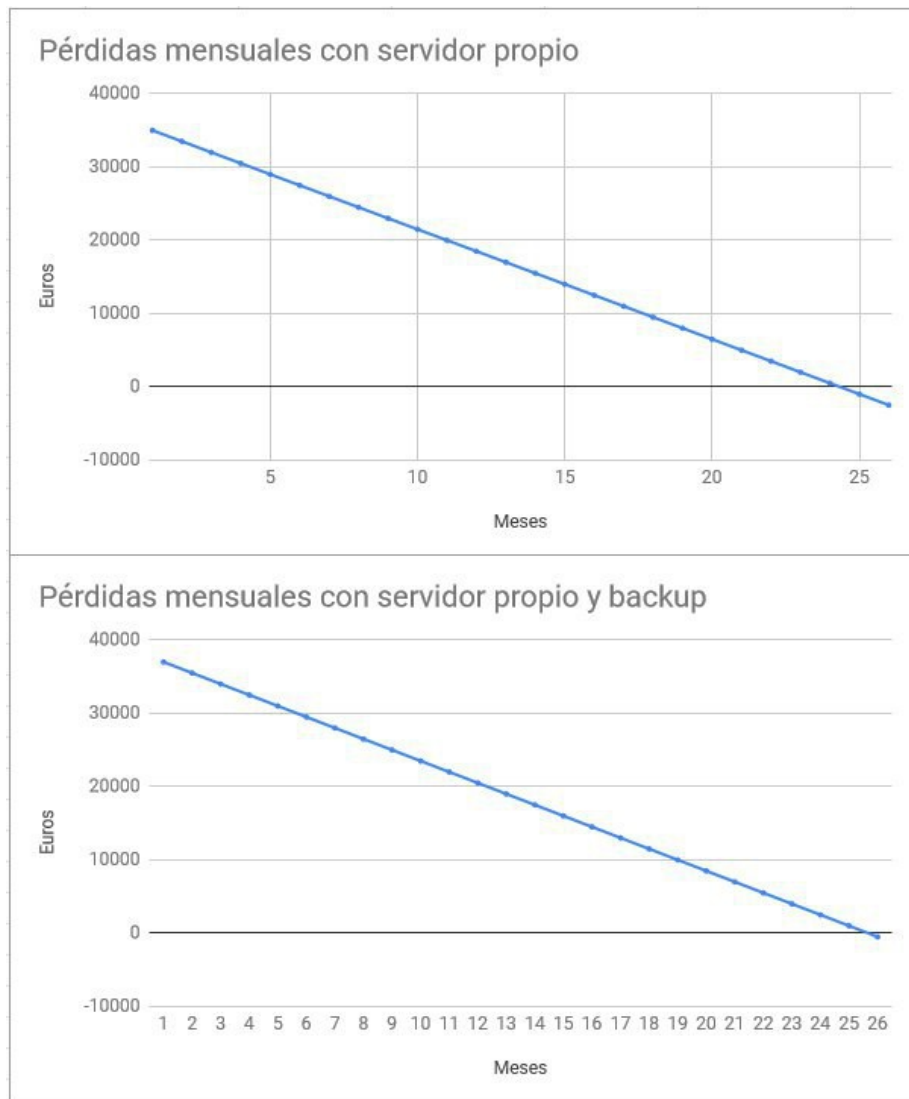


Figura 7.1: Gráficas de pérdidas mensuales atribuibles a la compra

Capítulo 8

Debido a la extensión del código desarrollado, en lugar de estar anexado a este documento, será descargable desde el link <https://github.com/AkaiBF/rrhtfg/>

Bibliografía

1. Ionic, “Documentation”, 2019 [Online]. Available: <https://ionicframework.com/docs>
2. Herrera, Fernando, “Angular: De cero a experto creando aplicaciones (Angular 7+)”, 2019 [Online]. Available: <https://www.udemy.com/angular-2-fernando-herrera/>
3. Trivago, “Página web oficial”, 2019 [Online]. Available: <https://www.trivago.es>
4. Azure, “Pricing Calculator”, 2019 [Online]. Available: <https://azure.microsoft.com/es-es/pricing/calculator/>
5. Booking, “Página web oficial”, 2019 [Online]. Available: <https://www.booking.com>
6. Wikipedia, “Thomas Cook”, 2018 [Online]. Available: https://es.wikipedia.org/wiki/Thomas_Cook
7. Tostado, Francisco Javier, “Thomas Cook y el origen de las agencias de viajes”, 2016 [Online]. Available: <https://franciscojaviertostado.com/2016/07/04/thomas-cook-y-el-origen-de-las-agencias-de-viaje/>
8. NPM, “Mongoose”, 2018 [Online]. Available: <https://www.npmjs.com/package/mongoose/>
9. Angular, “Página web oficial”, 2018 [Online]. Available: <https://angular.io>

