Despliegue de la Arquitectura e Infraestructura, la Base de Datos y la Aplicación móvil

Índice…

[**Despliegue de la Arquitectura e Infraestructura**](#_b23rolgfk50e) **3**

[Descripción general](#_464e7cwasjnm) 3

[Cloud:](#_oud90b3z27ly) 3

[Fog:](#_4l5ok6cmy1l7) 4

[Edge:](#_2u523yxr2ddt) 4

[Software / Hardware utilizado](#_iiug5gyv2o6) 6

[Manual de usuario](#_gp1apcw159i) 6

[**La aplicación móvil**](#_gcbudbmrxbwj) **7**

[Descripción general](#_v7eno92lpqt1) 7

[Explicació software/hardware utilitzat](#_deqbv3qh8u5p) 7

[Manual de usuario](#_sr4bctbzkcer) 8

[Mapa](#_cenmagypauzj) 9

[Tarjeta de embarque](#_eo48p08zhz5u) 10

[**Base de Datos**](#_y1vhppyae6aa) **11**

[Descripción general](#_4jhtld3jlzsa) 11

[Explicación software/hardware utilizado](#_xmmuxo2ei1ou) 12

# Despliegue de la Arquitectura e Infraestructura

## Descripción general

Todos los servicios que se prestarán serán gestionados en diferentes partes de la arquitectura propuesta según el grado de inmediatez que se pide a la información pedida. Ahora se hará una explicación de cada uno de los componentes de la arquitectura.

### Cloud:

Dispondremos de un servidor web con diferentes front ends (parte del servicio al que tienen acceso directo los usuarios), uno con autentificación que nos servirá para el sistema de gestión para los operadores del aeropuerto y las aerolíneas. Otro para para los operadores y administradores para el sistema de gestión de los vuelos y la terminal 2.

El servidor web sin autentificación, que dispondrá información general para los usuarios respecto a sus vuelos, hora de salida, puerta de embarque, etc., mapa del aeropuerto, servicios ofrecidos(restaurantes, tiendas..), información y publicidad.

Tambíen en el Cloud se mantendrá el Servidor DNS, se guardarán las copias de seguridad de todos los datos de la infraestructura y se mantendrá un servidor de correo para la empresa.

La Base de Datos generales con información necesaria para los servicios que no necesiten ser instantáneas(vuelos, la seva hora de sortida, puerta de embarque, usuarios, información del coche. tiendas, servicios disponibles en la terminal, ofertas disponibles, etc..)

El servidor DNS servirá para gestionar y agilizar todas las comunicaciones que se producirán dentro de las instalaciones de la Terminal 2.

### Fog:

El fog está construido como sistema distribuido con 5 servidores (el número es orientativo)

Además, tendremos 2 servidores de backups para cubrir el trabajo en caso de que cayera.

Los 5 servidores se comunicarán mediante el algoritmo de: token ring.

Aquí tendremos los servidores con la información necesaria para los servicios que requieran

más rapidez (localización de coches, pasajeros...)

### Edge:

Tendremos cuatro tipos de dispositivos en el edge. Móviles que se podrán conectar al servidor web mediante el frontend web, o se podrán conectar con la app.

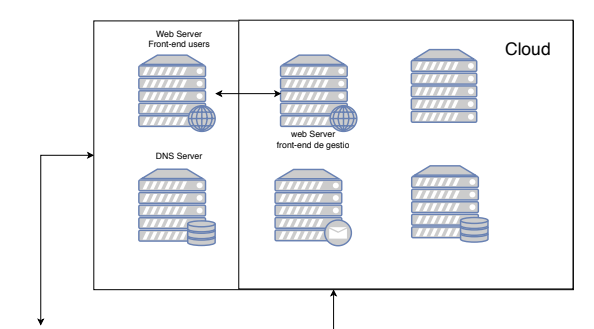
Coches que se conectarán a los servidores FOG (se encuentran en la misma red).

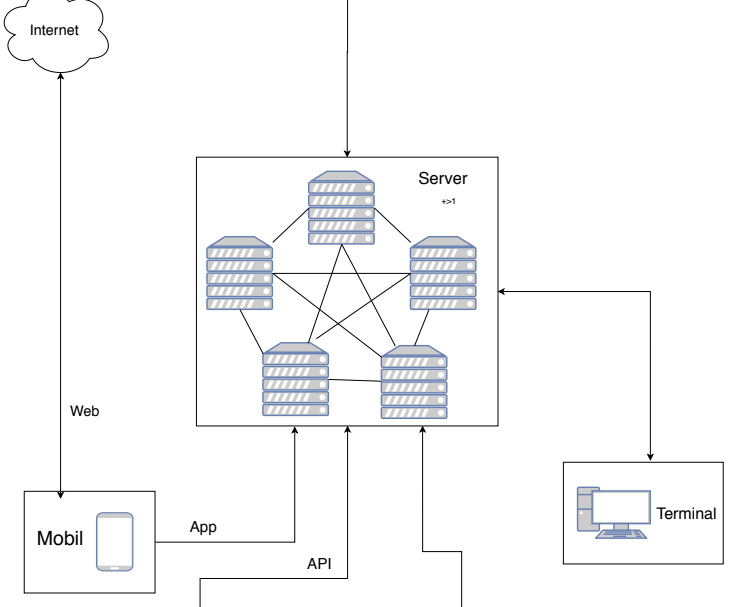
Sensores que se conectaran a los coches y a los servidores FOG.

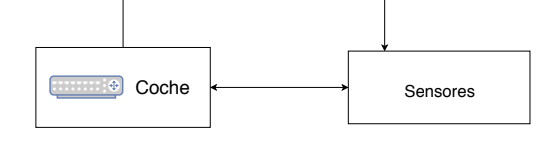
Terminales que tendremos dos tipos:

-Terminales para los trabajadores, los cuales podrán acceder a los dos frontends.

-Terminales para los pasajeros que permitirán pedir coches de forma remota.







## 

## Software / Hardware utilizado

El software utilizado está por concretar, se seguirán usando las tecnologías propuestas ( RESTful API para las comunicaciones entre componentes y la base de datos, Python 3.7 para los componentes programables, y Service dockerization(Docker) para hacer el despliegue de los componentes en el el servidor) En cuanto al hardware, se utilizará la máquina virtual presentada por los clientes para hacer el despliegue de los recursos en el Cloud. La configuración de servidores que se presenta és una configuración orientativa para cuando se quiera ofrecer el servicio dentro de las instalaciones del Aeropuerto.

## Manual de usuario

Todo el apartado de arquitectura será sólo visible para los administradores, y cuando se tenga una primera implementación, se redactarán los manuales pertinentes.

# 

# La aplicación móvil

## Descripción general

MyPassport es una herramienta en forma de aplicación para teléfonos móviles que permite organizar y gestionar el paso por la terminal 2 del aeropuerto.

Creando una cuenta en MyPassport se podrá consultar los detalles del vuelo de los clientes, acceder la tarjeta de embarque, consultar todas las opciones comerciales de la terminal, y entre otras cosas, pedir que uno de los coches autónomos les lleve hasta la puerta de embarque. Todo esto desde un teléfono móvil.

Como se acordó en los términos iniciales, el objetivo principal de esta herramienta es

conseguir que el cliente que asiste a la terminal encuentre toda la información accesible

en todo momento.

Buscamos que el cliente sienta la tranquilidad de saber que tiene todo bajo control.

Esta tranquilidad la vamos a traducir en un aumento de la actividad comercial, y por ello

la oferta comercial tiene una presencia importante en la app.

Le ofrecemos al cliente toda una variedad de actividades comerciales que puede llevar

a cabo en la terminal mientras espera su vuelo, sin miedo a llegar tarde a éste.

## Explicació software/hardware utilitzat

Nuestro equipo técnico ha decidido que, ya que existe el potencial y la posibilidad entre

los miembros del equipo de desarrollar una app en un framework multiplataforma, ésta

va a ser desarrollada en React Native.

React Native es el framework de desarrollo de apps multiplataforma más popular y

utilizado a día de hoy por las startups, y cobra mucho sentido cuando se lleva a cabo un

proyecto con un equipo de personas limitado.

Desarrollar una app en este framework te permite tener una app para iOS y para

Android sin tener que dedicar dos equipos paralelos a ello. Es suficiente con un proyecto

y un equipo dedicado.

Lo mejor de todo es que la experiencia de usuario no se ve afectada, ya que la app luce

igual que una aplicación nativa del sistema operativo, a diferencia de lo que pasa con

otros frameworks.

## Manual de usuario

Cada usuario tendrá acceso a las prestaciones descritas a continuación.

La app se centra en 5 apartados principales: mapa, comercio, tarjeta de embarque,

avisos y perfil.

• Mapa: Se trata de un mapa a escala con la localización aproximada a tiempo real

del usuario, así como la localización de puntos importantes como establecimientos comerciales, puertas de embarque y coches autónomos.

• Comercio: La parte de la app que pretende mostrar al usuario todo el abanico

de opciones comerciales de las que dispone en la terminal. Desde restaurantes, a

tiendas, actividades o zonas reservadas.

• Tarjeta de embarque: El cliente tiene acceso a su tarjeta de embarque en todo

momento desde este apartado de la app. Podrá usarla para entrar a la terminal

mediante un código QR.

• Avisos: Canal directo de comunicación entre las aerolíneas y los usuarios. El

usuario de la app dispondrá de avisos relacionados con su vuelo, y alertas sobre

las horas y actualizaciones sobre el vuelo. Las aerolíneas podrán mandar avisos a

usuarios en específico.

• Perfil: Aquí es desde donde el cliente gestiona sus datos de usuario.

Se dará prioridad a una interfaz sencilla, intuitiva y agradable a la vista, para conseguir

la mejor experiencia de usuario posible. Estos 5 apartados estarán disponibles en una

barra de navegación inferior, que es la parte más accesible por el dedo del usuario

cuando utiliza el teléfono. Se evitarán menús laterales, o pantallas innecesarias para que

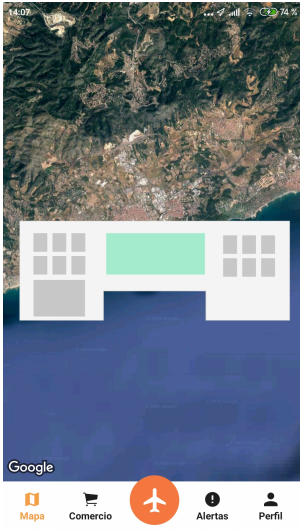
el usuario tenga siempre a la vista todas las opciones. Todo se llevará a cabo de la forma

más gráfica posible, evitando textos y exceso de información en la app.

Ahora se presentarán diferentes diseños de cómo se vería la aplicación, desde el punto de vista de un usuario.

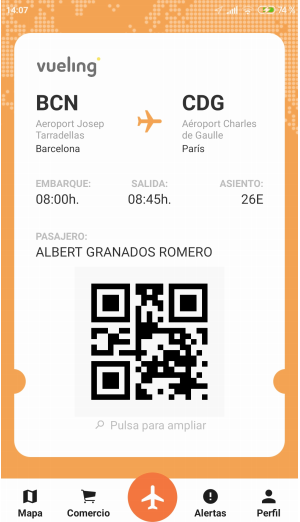
### Mapa

Se ha comenzado con el desarrollo del mapa de la terminal, que consistirá en aprovechar los recursos de la API de los mapas de Google, para dibujar sobre éste los modelos del mapa de la terminal.



### Tarjeta de embarque

Ya está listo el diseño y la implementación de las tarjetas de embarque a modo local.



# **Base de Datos**

## **Descripción general**

Hemos decidido por optar a un diseño de la base de datos básico. En el esquema que adjuntamos más abajo detallamos como tenemos planeado almacenar la información imprescindible y básica de nuestra Terminal 2 para los diferentes usuarios.

**Conceptos**

Explicación de lo que simboliza cada concepto dentro del diagrama conceptual que posteriormente se convertirán en información que se almacenará en nuestra base de datos.

**Usuario**: Persona común que utiliza un sistema informático.

**Administrador**: Usuario que tiene control absoluto de nuestro sistema inteligente.

**Operador**: Usuario encargado de gestionar los vuelos de las diferentes aerolíneas.

**Pasajero**: Usuario cliente al que facilitamos información de sus vuelos y datos.

**Vuelo**: Trayecto que realiza un avión de un sitio a otro.

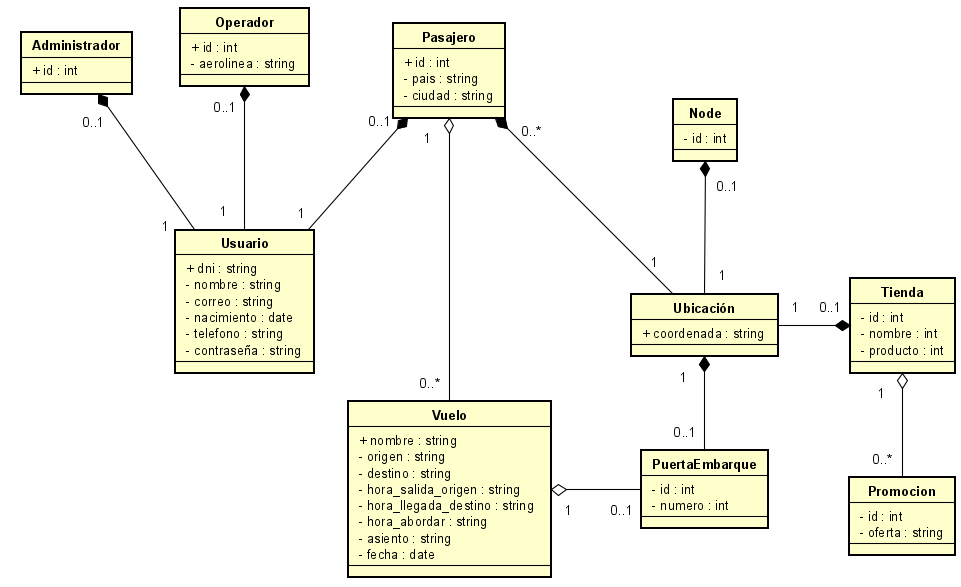
**Puerta\_embarque**: Información del lugar donde se accede al avión.

**Ubicación**: Lugar donde se encuentra alguna cosa(pasajero, puerta de embarque, tiendas, servicio)

**Node**: Concepto de coche

**Tienda**: Establecimiento donde se vende algún producto.

**Promoción**: Publicidad que se da de un determinado producto.



**Descripción modelo conceptual no relacional.**

Al ser un modelo conceptual no relacional, las líneas que hay entre dos conceptos no indican la relación que hay entre ellos sino como se se desglosa, las flechas con la punta negra indican obligación y las blancas indican que se puede desglosar o no.

Por ejemplo el concepto pasajero se desglosa en tres conceptos o entidades, pasajero se compone de sus propios atributos, debe tener un usuario, debe tener una ubicación y puede tener un vuelo asignado.

## **Explicación software/hardware utilizado**

El software que utilizaremos para el almacenamiento de datos es MongoDB, mongo viene de la palabra inglesa humongus(“enorme”), es un sistema de base de datos NOSQL(No relacional).

Lo hemos escogido porque es escalable, flexible, fácil de aprender, usar y sobretodo que

la información que almacenaremos será la de un aeropuerto internacional, lo que nos lleva a almacenar grandes cantidades de datos que además irá creciendo constantemente.

# 

**Descripción modelo lógico no relacional en MongoDB**

En las bases de datos no relaciones, el concepto de tabla y su relación mediante claves foráneas son muy diferentes, en mongoDB no existe una estructura de relaciones, en lugar de tablas se utilizan las colecciones(collections) que están formadas con atributos igual que las tablas de las bases de datos relacionales, pero lo que hace que sean no relaciones es que están orientadas a objetos, es decir que dentro de una colección se pueden añadir objetos, es como si en una base de datos SQL se pudiesen añadir tablas dentro de otras tablas, en este apartado se explicará la estructura de cada colección de nuestra base de datos.

**Colecciones**

**passengers**



Esta sería una colección de pasajeros, con un id(su documento de identidad). un nombre, el email, su fecha de nacimiento, el país y la ciudad por si en un futuro necesitáramos generar estadísticas de los pasajeros. También podemos observar que la contraseña de nuestro pasajero está cifrada con un hash y así mantener la privacidad de su contraseña, la ubicación es la posición actual donde se encuentra.

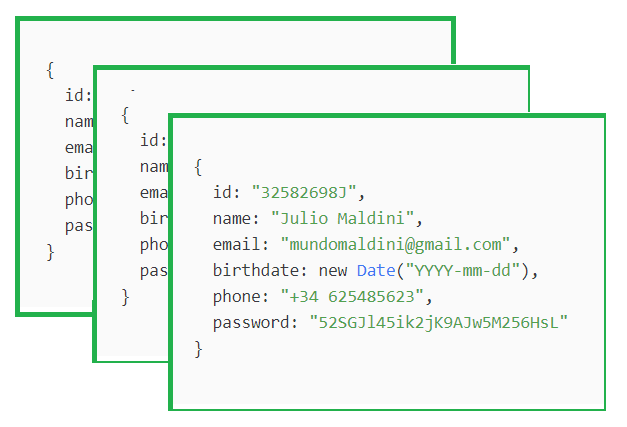
También tiene información respecto a su vuelo y este a su vez tiene información de su puerta de embarque.

**Operators**



En el caso de los operadores tendría los mismo atributos del concepto Usuario de nuestro MC, solo que en este le añadimos el atributo aerolínea que es para la compañía que trabaja.

**Administrators**



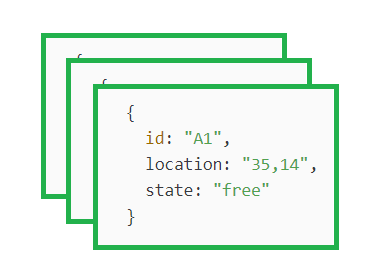
La colección de Administrators tiene los mismo atributos que el concepto usuario porque hemos supuesto que no habrá muchos administradores por lo tanto no hace falta que tengamos demasiada información de ellos en nuestro sistema para estadísticas o similares.

**Shops**



Estructura básica de las tiendas, contiene información del nombre, el producto que vende, su localización para guiar a los pasajeros hacia las tiendas cercanas, un atributo llamado type que nos indicará de qué será este comercio, que pueden ser tiendas, restaurantes, de ocio y por último una lista de las ofertas que promocionan.

**Node**



La colección Node tiene solo estos atributos porque aún no sabemos qué datos relevantes necesitamos.

**Flights**



Estructura básica de la colección vuelo que guarda información del nombre del vuelo, el origen, destino del vuelo, la hora de embarque, hora a la que llegará a su destino como la hora a la que sale del origen, también tenemos la fecha del vuelo y por último tenemos la información de la puerta de embarque con su nombre y la localización.