



# Documento de Arquitectura

**Integrantes:**

Mathías Fernández  
Eduardo Flores  
Federico Acosta  
Gabriel Suárez  
Pablo Perdomo  
Alexander Rodriguez

**Tutor:**

Fernando Arrieta

# Índice

|  |           |
|--|-----------|
| <b>Índice.....</b>   | <b>2</b>  |
| <b>Introducción.....</b>   | <b>3</b>  |
| Propósito.....   | 3         |
| Alcance.....   | 3         |
| Definiciones, siglas y abreviaturas.....                                     | 3         |
| Referencias.....   | 3         |
| <b>Vista de modelo de Casos de uso.....</b>                                  | <b>3</b>  |
| Diagramas de casos de uso críticos.....                                      | 3         |
| Administrador.....   | 4         |
| Cliente.....   | 5         |
| Vendedor.....  | 6         |
| Todos los usuarios.....  | 7         |
| <b>Trazabilidad desde el modelo de casos de uso al modelo de diseño.....</b> | <b>7</b>  |
| Administrador.....   | 7         |
| Cliente.....   | 7         |
| Vendedor.....  | 8         |
| <b>Vista de Modelo de Implementación.....</b>                                | <b>9</b>  |
| Capa de Presentación.....  | 10        |
| Capa de Negocio.....   | 10        |
| Capa de Datos.....   | 10        |
| <b>Vista de Modelo de Despliegue.....</b>                                    | <b>11</b> |

# Introducción

## Propósito

El presente documento tiene como objetivo plasmar una visión general sobre el modelado de arquitectura a realizar en función de los casos de uso presentados.

## Alcance

En el documento se presentan varios diagramas que permiten visualizar qué componentes conforman el sistema y cómo se comunican entre ellos.

## Definiciones, siglas y abreviaturas

En caso de encontrarse términos técnicos desconocidos, se recomienda referirse al glosario con el fin de facilitar la comprensión completa del documento.

## Referencias

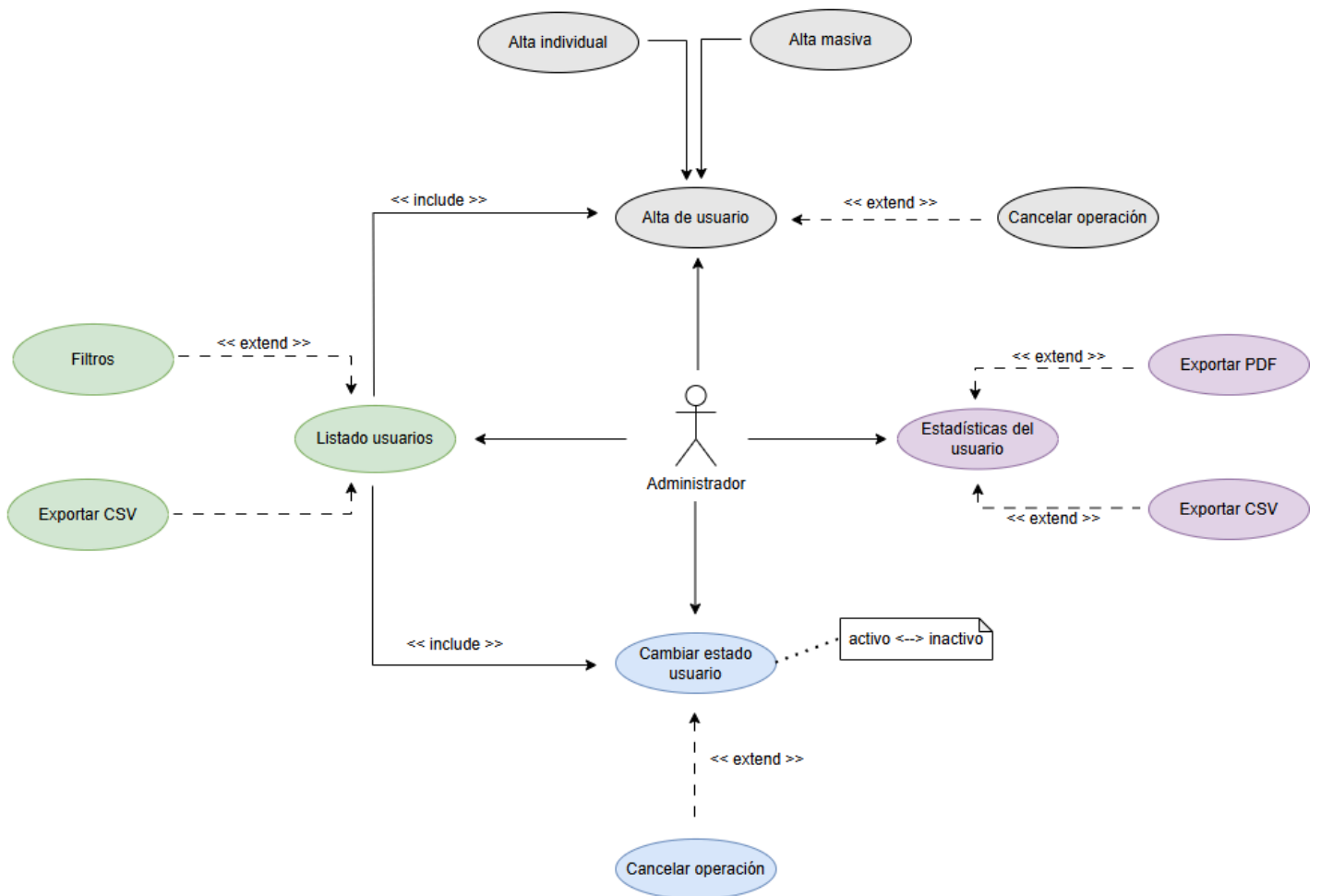
Se recomienda realizar una lectura previa del documento de requerimientos y del documento de casos de uso para lograr un mayor entendimiento del presente documento.

# Vista de modelo de Casos de uso

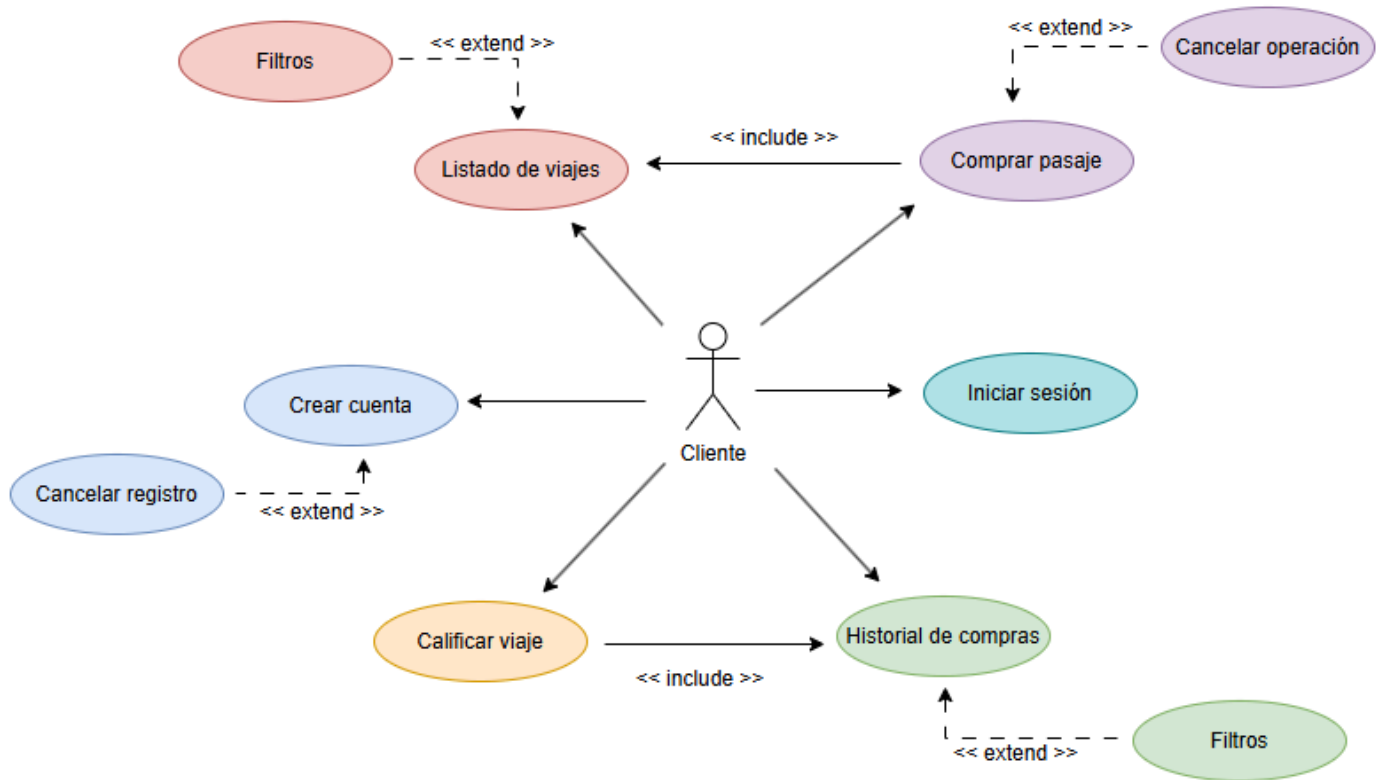
## Diagramas de casos de uso críticos

A continuación, se enumeran los casos de uso considerados críticos, los cuales recibirán una prioridad más alta que los demás. Estos están organizados según los roles que ejecutan cada caso de uso. Una descripción más detallada de cada uno puede encontrarse en el documento de casos de uso, donde se presenta el flujo paso a paso correspondiente.

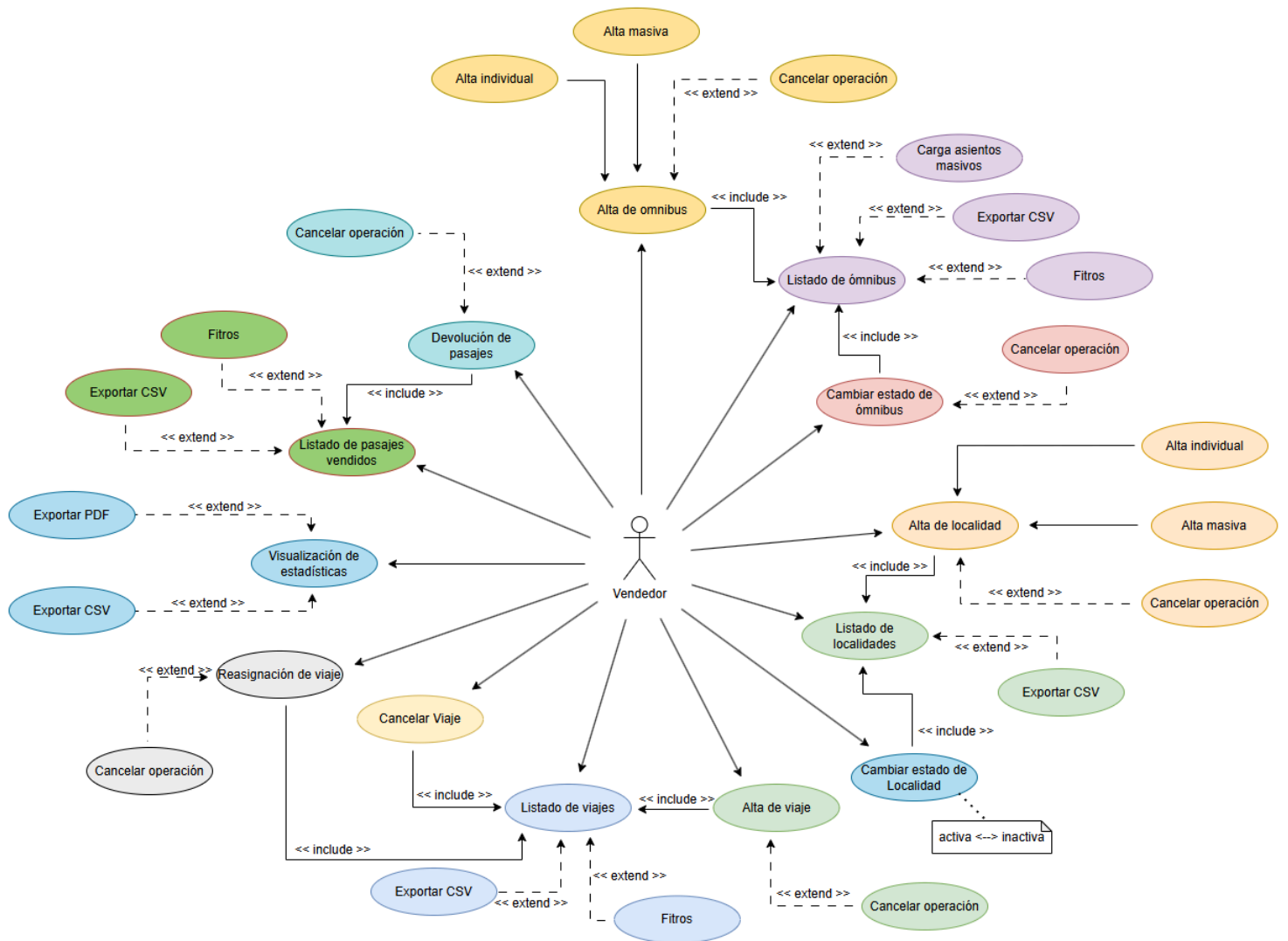
## Administrador



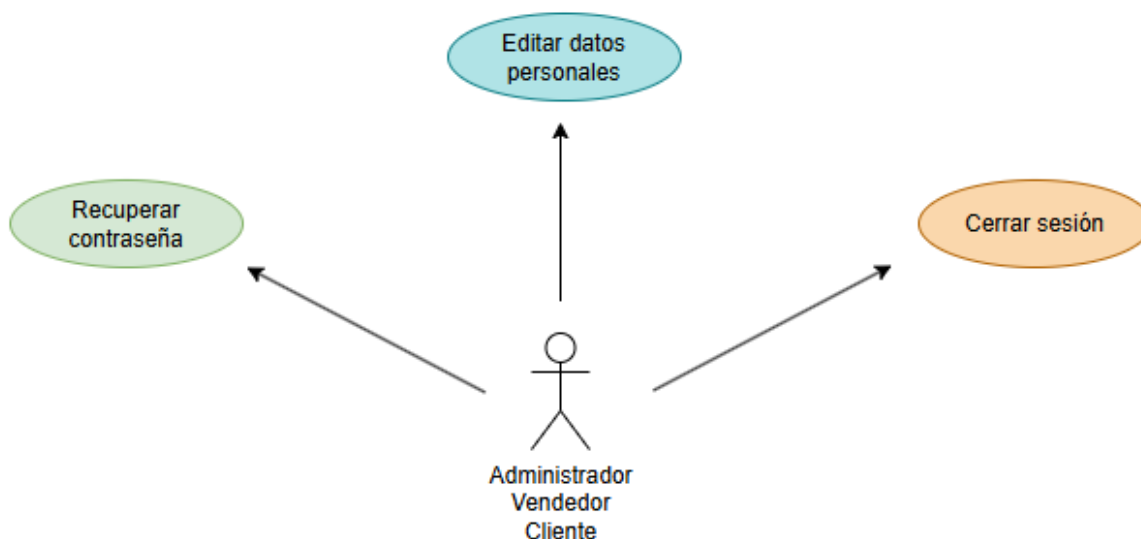
## Cliente



Vendor



Todos los usuarios



Trazabilidad desde el modelo de casos de uso al modelo de diseño

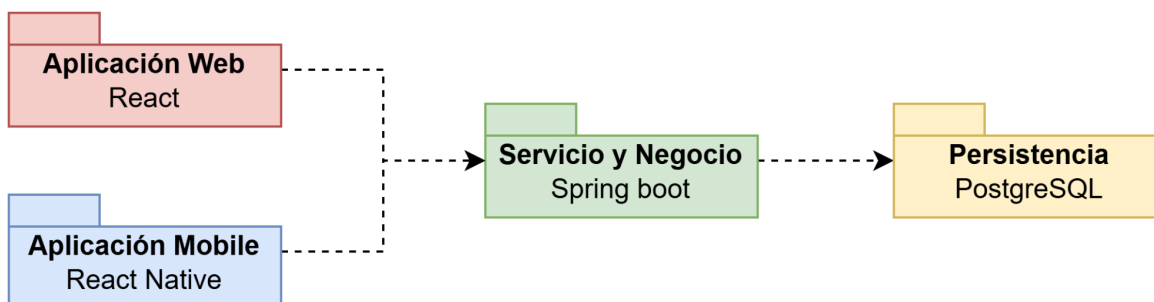
## Administrador

Alta de Usuario, Baja de Usuario, Listado de Usuarios y Estadísticas de Usuario



## Cliente

Crear Cuenta, Historial de Compras y Calificar Viajes



## Vendedor

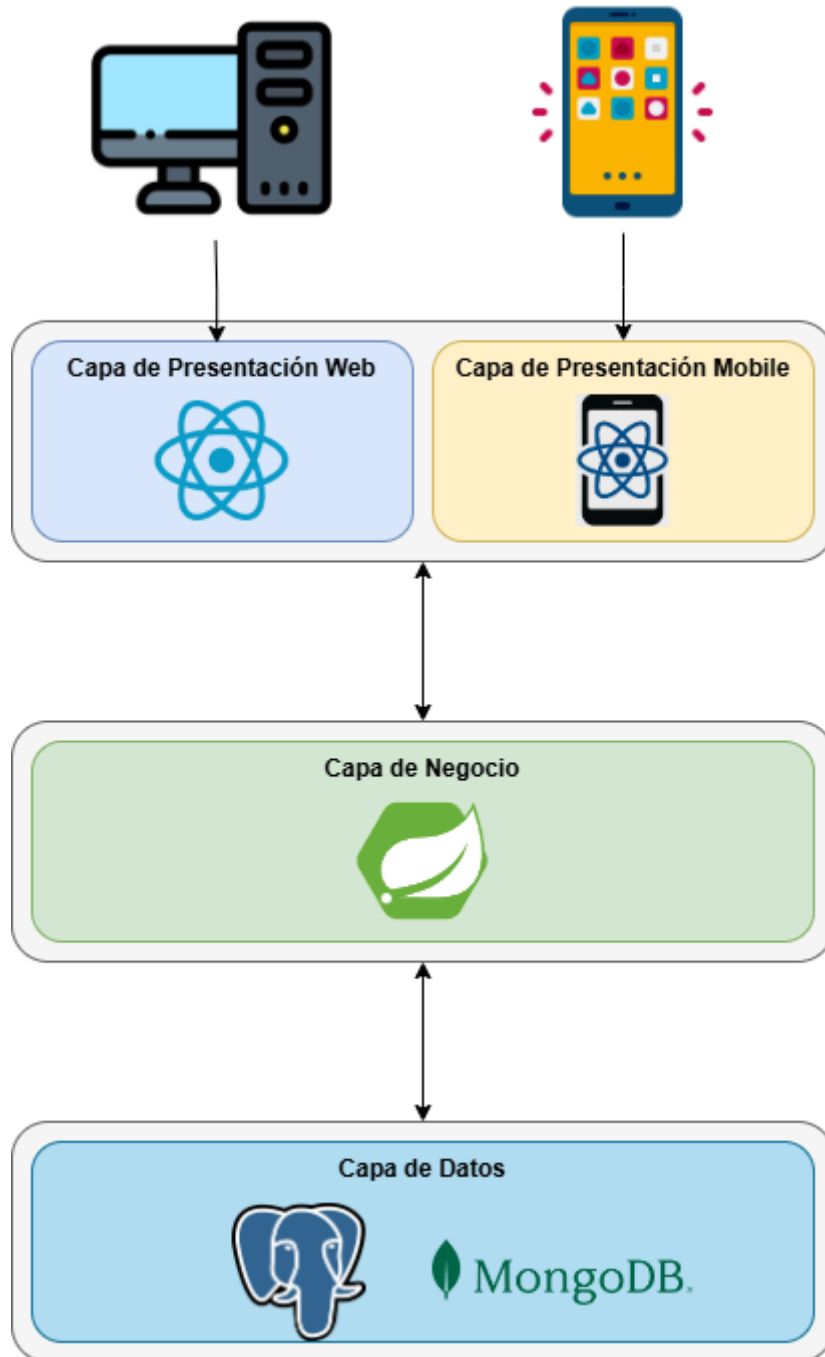
Alta de Viaje, Reasignación de Viaje, Listado de Viajes, Cierre de Venta de Pasajes, Devolución de Pasaje, Lista Pasajes Vendidos, Alta Localidad, Alta Ómnibus, Listado Ómnibus, Desactivación de Ómnibus, Reactivación de Ómnibus y Visualización de Estadísticas





## Vista de Modelo de Implementación

En esta representación gráfica se ofrece una visión general del sistema en conjunto, siendo representadas las distintas capas que la componen.



## Capa de Presentación

Esta capa es responsable de proporcionar las interfaces gráficas a los distintos tipos de usuarios.

Incluye tres componentes:

- Una **aplicación web**, desarrollada con **React** y servida con **Node.js**, orientada a los clientes finales (**FRONTOFFICE**).
- Una **aplicación móvil**, implementada con **React Native**, que se ejecuta en los dispositivos de los usuarios.
- Una **interfaz administrativa**, también desarrollada con **React** y **Node.js**, utilizada por administradores y vendedores (**BACKOFFICE**).

Todas estas interfaces permiten la interacción con el sistema de forma intuitiva, responsiva y adaptada a los diferentes perfiles de usuario.

## Capa de Negocio

Esta capa se encarga de la lógica del sistema y de la exposición de servicios mediante APIs REST, que son consumidas por las aplicaciones de la capa de presentación.

Está compuesta por dos servicios desarrollados con **Spring Boot**:

- El **servicio principal (BACKEND)**, que implementa toda la **lógica de negocio** de *TecnobusUy*.
- El **servicio de notificaciones (NOTIFICACIONES)**, que gestiona el envío de **notificaciones push** a clientes web y móviles.

Ambos servicios están diseñados de forma modular y pueden escalarse de forma independiente.

## Capa de Datos

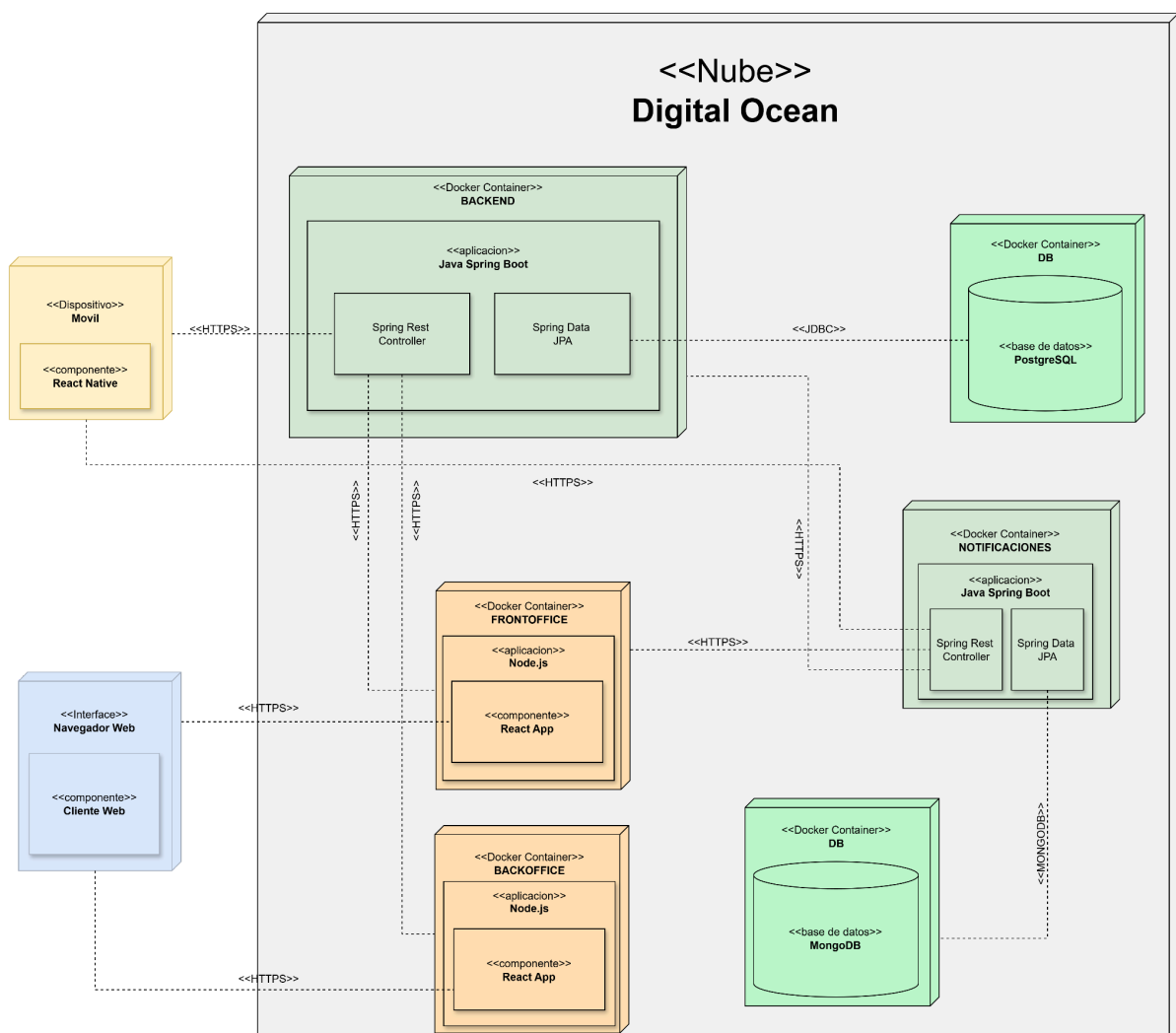
Esta capa gestiona el almacenamiento y acceso a la información utilizada por la capa de negocio.

Está compuesta por dos sistemas de gestión de bases de datos:

- **PostgreSQL**, utilizado como base de datos principal del sistema, responsable de almacenar la información estructurada gestionada por el backend.
  - **MongoDB**, utilizado por el servicio de notificaciones para almacenar los **tokens de dispositivos** y los **mensajes** de cada notificación enviada.
- La capa de datos garantiza la persistencia, seguridad y eficiencia en el manejo de la información de todo el sistema.

## Vista de Modelo de Despliegue

A continuación se presenta un diagrama que describe el sistema en su totalidad y la arquitectura que lo compone, centrándose en la estructura interna, los componentes tanto internos como externos, y las conexiones que posibilitan el intercambio de datos entre ellos.



Todos los nodos de la aplicación se encuentran desplegados en la nube, específicamente en la infraestructura de **Digital Ocean**, con excepción de los clientes.

El **cliente móvil**, implementado con **React Native**, se ejecuta localmente en el dispositivo del usuario. De forma similar, el **cliente web**, accesible mediante un navegador, también se ejecuta en el entorno local del usuario y actúa como interfaz de interacción con la aplicación.

En el entorno cloud, la arquitectura consta de varios **contenedores Docker**, cada uno con un rol específico:

- El contenedor **BACKEND** encapsula la estructura principal de la aplicación, desarrollada con **Spring Boot**. Aquí se gestiona toda la **lógica de negocio** de *TecnobusUy*, mediante controladores REST (**Spring Rest Controller**) para manejar solicitudes HTTP y una capa de persistencia (**Spring Data JPA**) que interactúa con la base de datos principal.
- El contenedor **NOTIFICACIONES** alberga un segundo servicio basado también en **Spring Boot**, responsable de gestionar el sistema de **notificaciones push** que reciben los clientes, tanto en la aplicación web como en la móvil. Este servicio también sigue una arquitectura REST.
- El contenedor **FRONTOFFICE** aloja una aplicación construida con **React**, servida por un entorno **Node.js**, y representa la **interfaz de usuario final**, es decir, la utilizada por los clientes de TecnobusUy.
- El contenedor **BACKOFFICE**, también construido con **React** y servido por **Node.js**, representa la **interfaz utilizada por administradores y vendedores**, proporcionando acceso a herramientas de gestión y administración del sistema.
- La infraestructura cuenta con **dos contenedores de bases de datos**:
  - El contenedor **PostgreSQL**, utilizado como **base de datos principal** para el backend, donde se almacenan los datos persistentes de la aplicación. La comunicación entre el backend y esta base se realiza mediante el protocolo **JDBC**.
  - El contenedor **MongoDB**, utilizado por el servicio de notificaciones para almacenar los **tokens de los dispositivos** y los **mensajes**

asociados a cada notificación enviada. La comunicación entre el servicio de notificaciones y esta base de datos se realiza mediante el protocolo **mongodb**.

Toda la comunicación entre los distintos componentes desplegados en la nube se realiza mediante el protocolo **HTTPS**, garantizando la **confidencialidad** e **integridad** de los datos transmitidos.