MANUAL TÉCNICO

PLATAFORMA DE SALUD
PROYECTO DEL CURSO IPC2:
APLICACIÓN WEB CON JAVA Y ANGULAR

HERRAMIENTAS UTILIZADAS

- ## LENGUAJE DE PROGRAMACIÓN JAVA - VERSIÓN 17 (BACKEND)

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems.

El lenguaje Java proporciona:

- El paradigma de la programación orientada a objetos.
- Ejecución de un mismo programa en múltiples sistemas operativos y plataformas.
- Es posible utilizarlo para múltiples propósitos, desde aplicaciones de escritorio hasta en servidores web.
- Tiene una curva de aprendizaje media pero también toma lo mejor de otros lenguajes orientados a objetos, como C++.

También se utilizaron librerías/bibliotecas externas como:

- **JDBC -** Para la conexión a la base de datos
- **Jakarta EE -** Servicios web
- **Apache Tomcat -** Servidor web

- ## RDBMS (SISTEMA DE BASES DE DATOS RELACIONAL): MYSQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web.

MySQL es muy utilizado en aplicaciones web, como Joomla, Wordpress, Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.

● FRAMEWORK DE JAVASCRIPT - ANGULAR v15 (FRONTEND)

Angular (No confundir con AngularJS el cual está descontinuado) es un framework para aplicaciones web desarrollado en TypeScript, de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript.

Angular se basa en clases tipo "Componentes", cuyas propiedades son las usadas para hacer el binding de los datos. En dichas clases tenemos propiedades (variables) y métodos (funciones a llamar).

También se utilizaron librerías/bibliotecas externas como:

- **Angular Material -** Implementación de componentes gráficos de interfaz de Material Design 2 desarrollado por Google para Angular

● IDE: INTELLIJ IDEA ULTIMATE

IntelliJ IDEA es un entorno de desarrollo integrado (IDE) escrito en Java para desarrollar software informático escrito en Java, Kotlin, Groovy y otros lenguajes basados en la JVM. Está desarrollado por JetBrains (anteriormente conocido como IntelliJ) y está disponible como una edición comunitaria con licencia de Apache 2, y en una edición comercial patentada. Ambos se pueden utilizar para el desarrollo comercial.

● GITHUB

Es un sistema de control de versiones de código y gestión de proyectos, a su vez también funciona como una plataforma de estilo red social diseñada para desarrolladores para poder compartir código entre más personas y colaborar en el mismo.

- ## DIAGRAMS.NET - DRAW.IO

Draw.io es una herramienta de creación y edición de diagramas libre que permite la integración con diversas plataformas, principalmente Google Drive, el cual permite la compartición y colaboración de varias personas dentro del mismo proyecto. El software consiste en una aplicación web realizada mayoritariamente en JavaScript y licenciada con Apache v2, lo que la hace funcionar en una amplia gama de navegadores y permite la creación de diagramas, contando con modelos para diversos tipos como pueden ser diagramas UML, esquemas de red, flujogramas, etc. También permite crear colecciones de diagramas e imágenes personalizados para utilizar en los diagramas.

- ## MACOS v13 - SISTEMA OPERATIVO

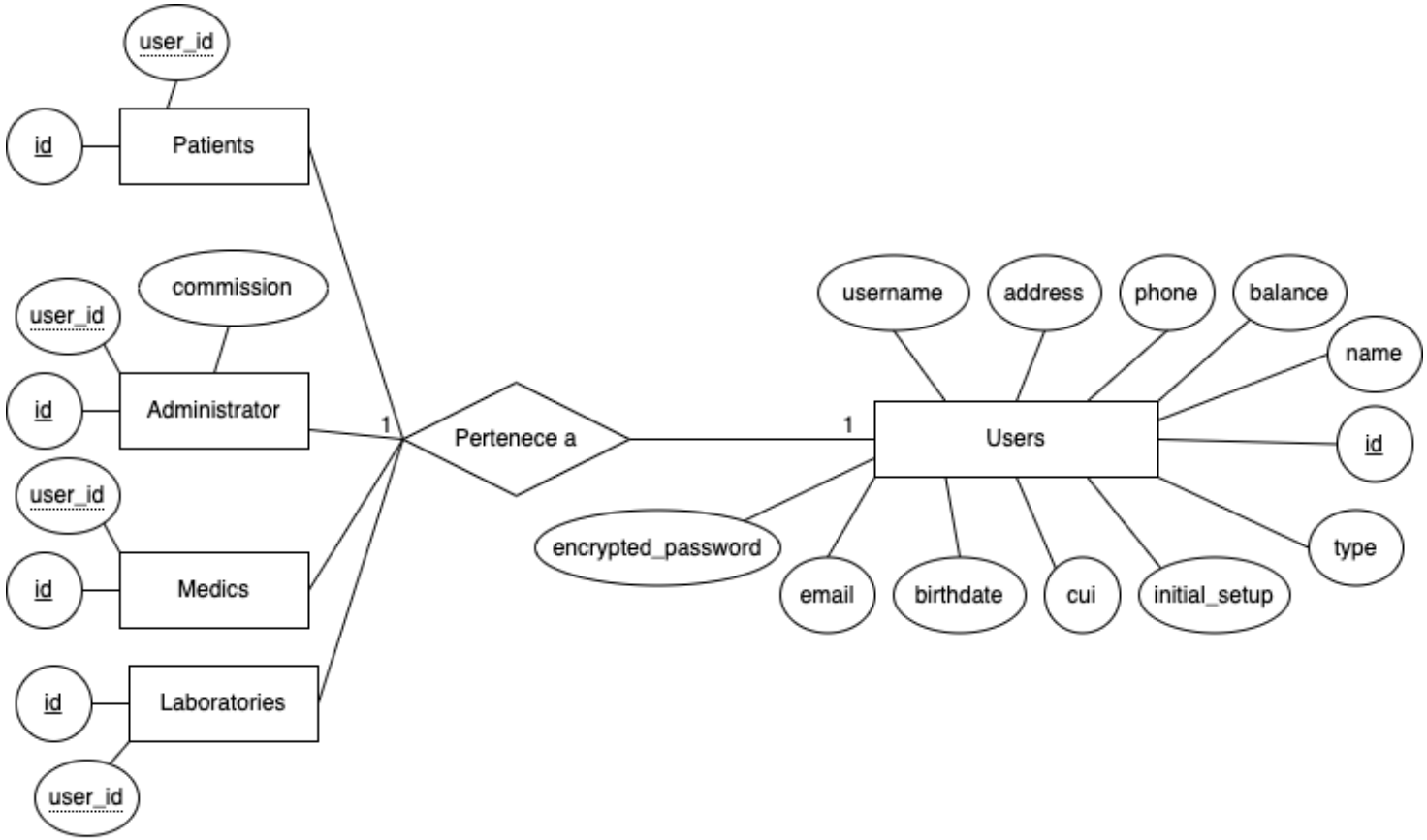
macOS (previamente Mac OS X, luego OS X) es una serie de sistemas operativos gráficos desarrollados y comercializados por Apple desde 2001. Es el sistema operativo principal para la familia de computadoras Mac de Apple. Dentro del mercado de computadoras de escritorio, portátiles, hogareñas y mediante el uso de la web.

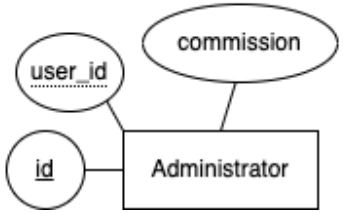
macOS se basa en tecnologías desarrolladas entre 1985 y 1997 en NeXT. Se logró la certificación UNIX 03 para la versión Intel de Mac OS X 10.5 Leopard y todos los lanzamientos de Mac OS X 10.6 Snow Leopard hasta la versión actual también tienen la certificación UNIX 03. macOS comparte su núcleo basado en Unix, llamado Darwin, y muchos de sus frameworks con iOS 16, tvOS y watchOS.

DIAGRAMA DE CLASES

Referirse al documento PDF incluido

DIAGRAMA ENTIDAD RELACIÓN





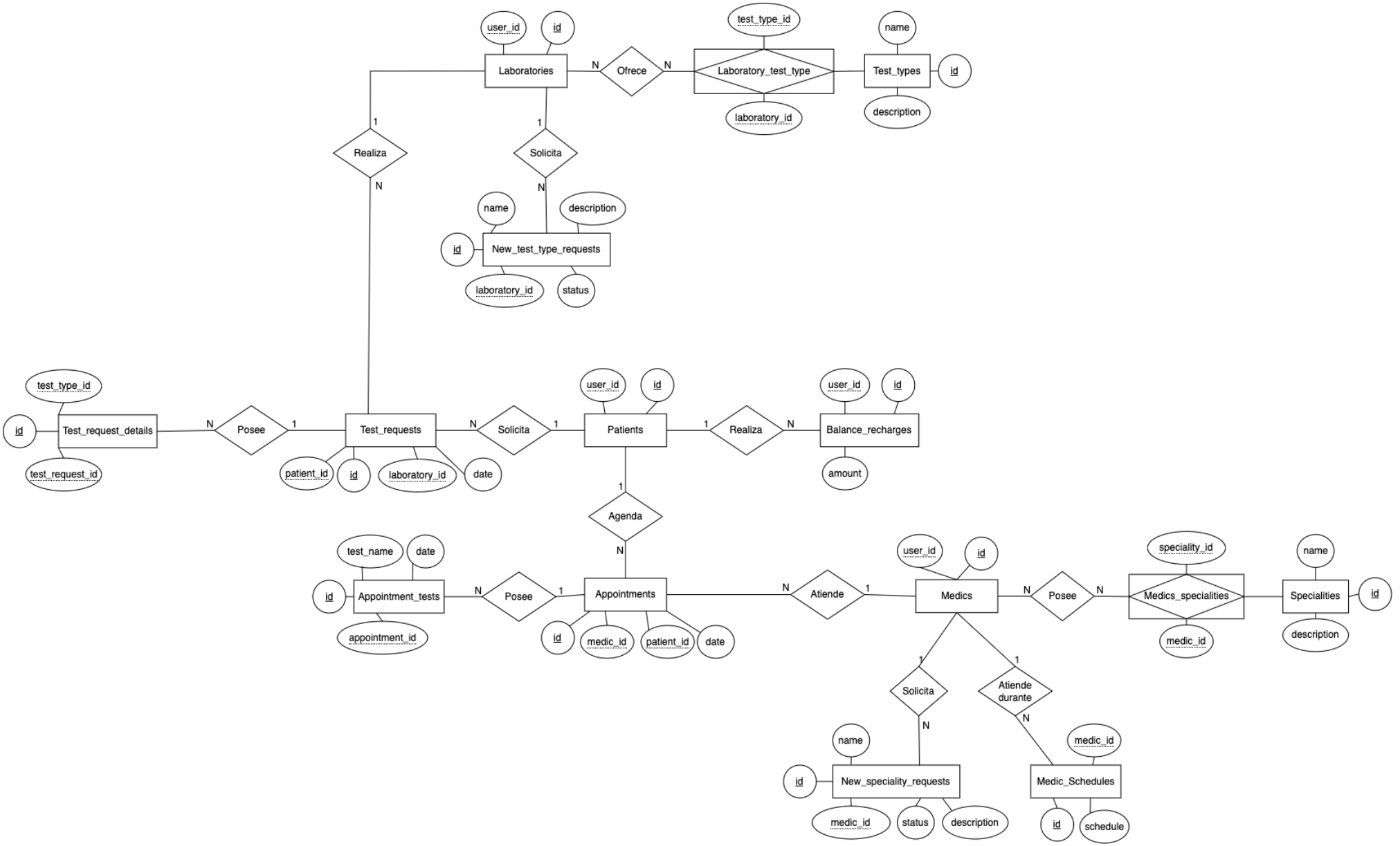

user_id
id
Laboratories
N
Ofrece
N
test_type_id
Laboratory_test_type
laboratory_id
name
Test_types
id
description
1
Realiza
N
1
Solicita
N
name
description
id
New_test_type_requests
laboratory_id
status
test_type_id
id
Test_request_details
N
Posee
1
Test_requests
N
Solicita
1
user_id
id
Patients
1
Realiza
N
user_id
id
Balance_recharges
amount
test_request_id
patient_id
id
laboratory_id
date
1
Agenda
N
test_name
date
id
Appointment_tests
appointment_id
N
Posee
1
Appointments
N
Atiende
1
user_id
id
Medics
N
Posee
N
speciality_id
Medics_specialities
medic_id
name
Specialities
id
description
1
Solicita
N
1
Atiende durante
N
name
id
New_speciality_requests
medic_id
status
description
medic_id
Medic_Schedules
id
schedule

DIAGRAMA DE TABLAS

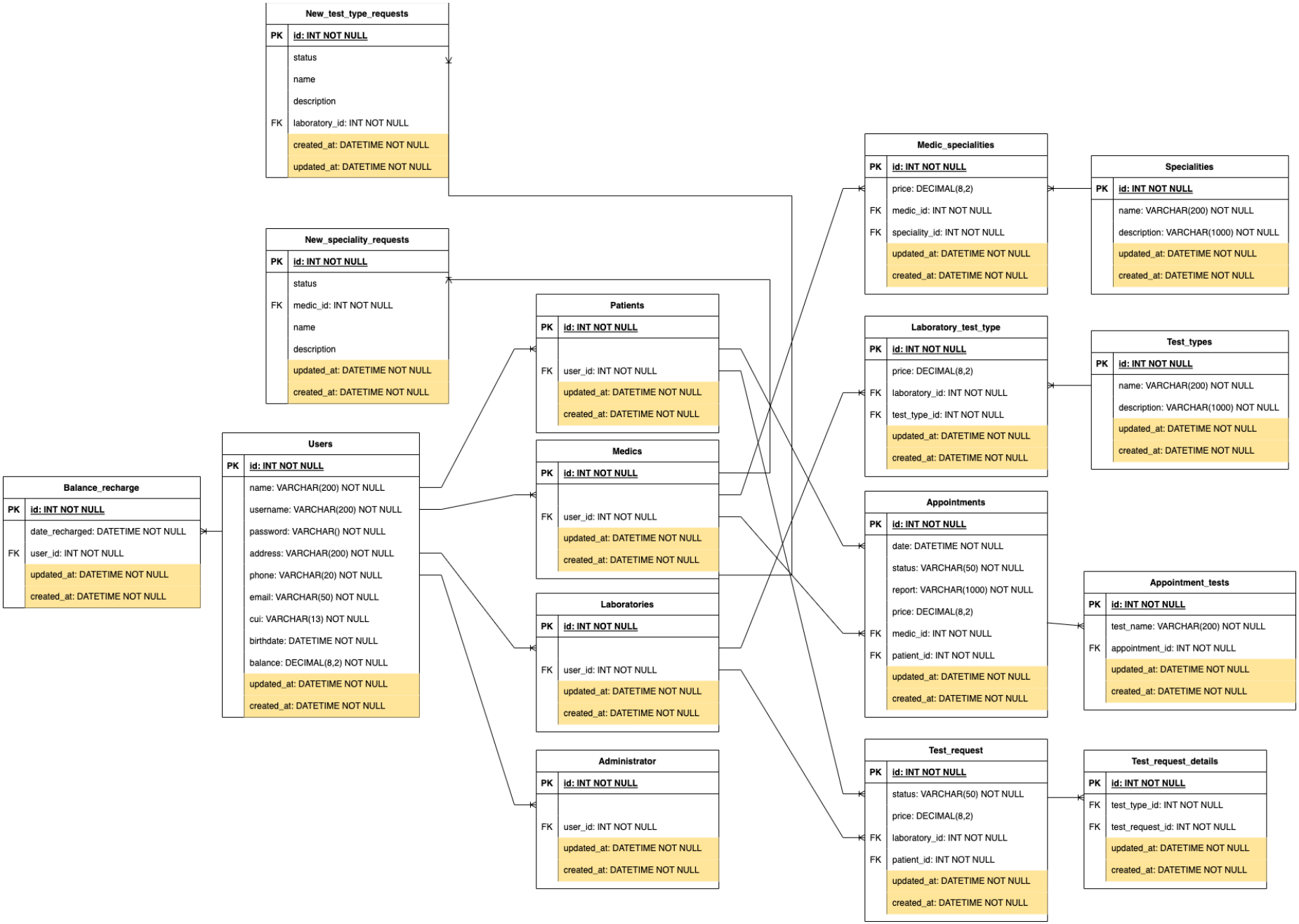
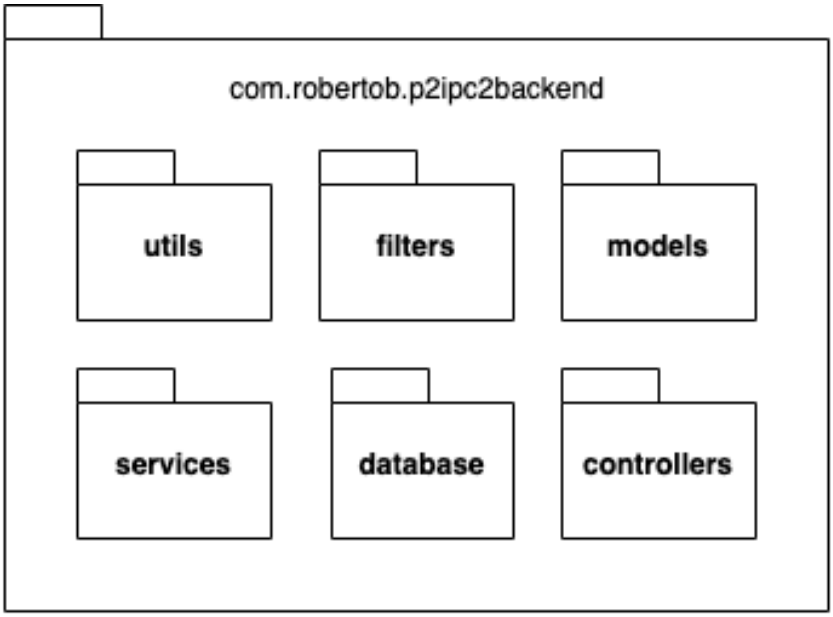


DIAGRAMA DE PAQUETES



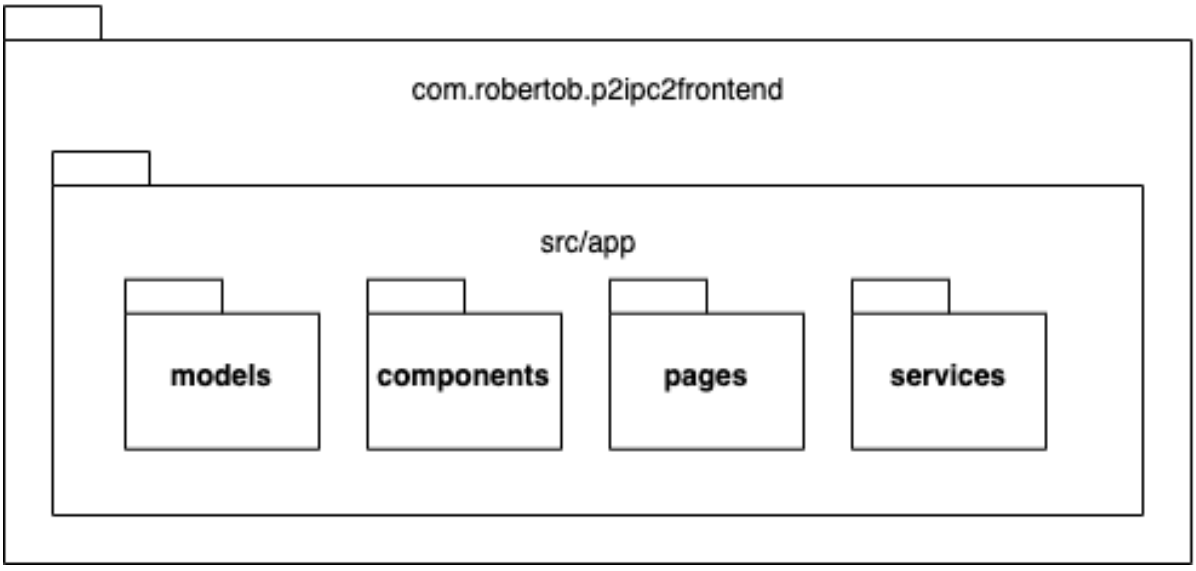


DIAGRAMA DE COMPONENTES

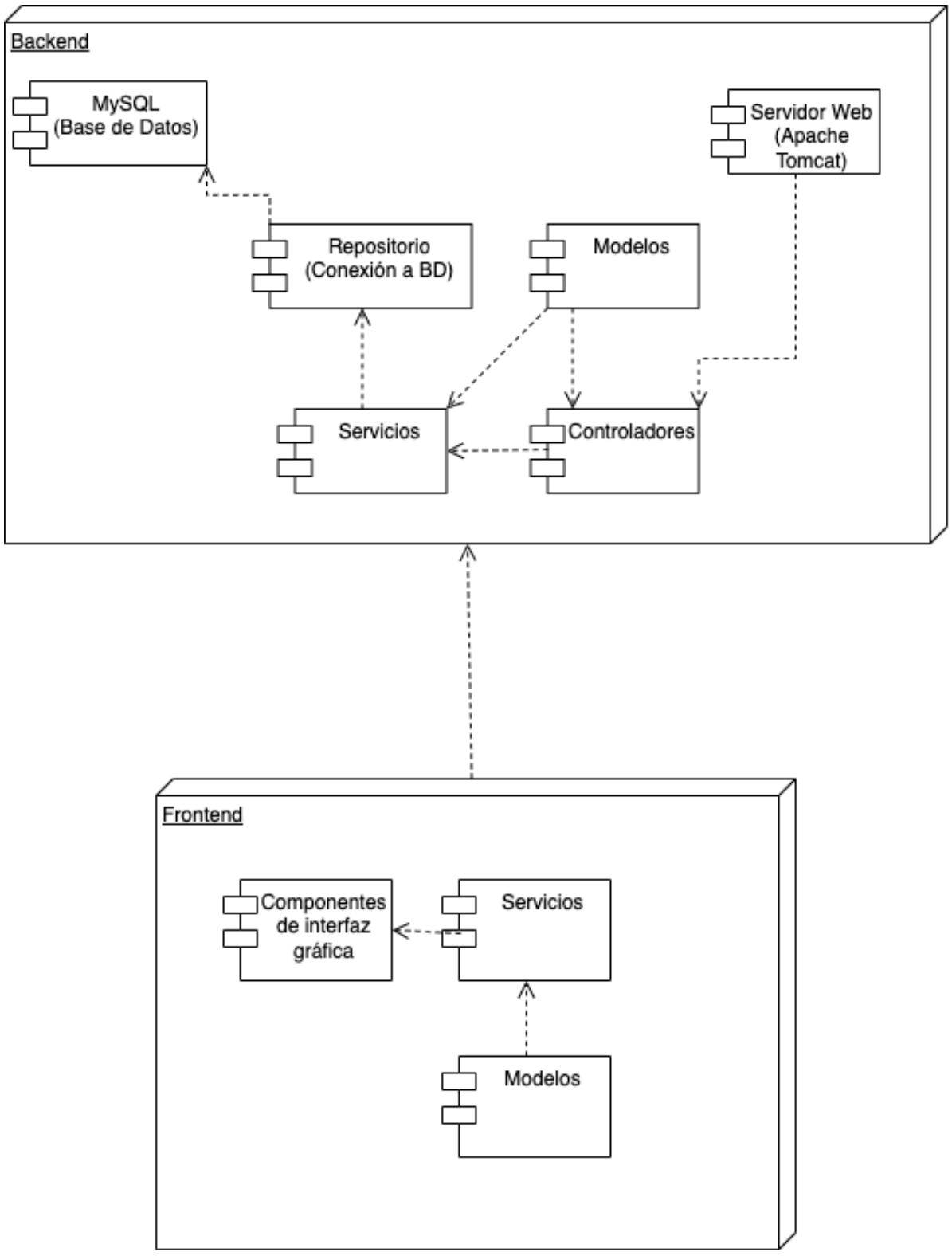
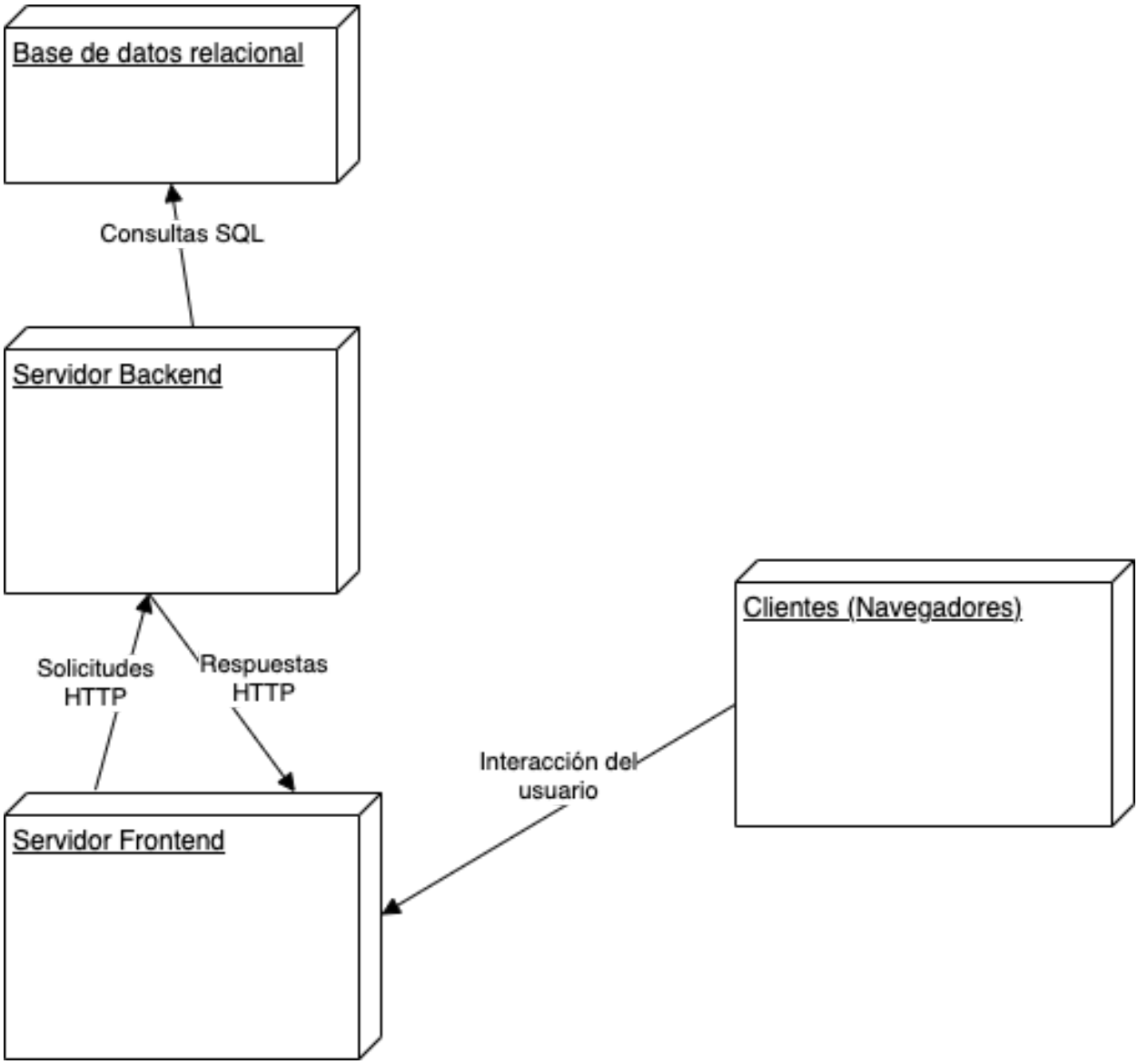


DIAGRAMA DE DESPLIEGUE



MAPEO FÍSICO DE LA BASE DE DATOS

```
// database.sql
DROP DATABASE IF EXISTS clinic;
CREATE DATABASE clinic;
USE clinic;

CREATE TABLE users (
    id INT NOT NULL AUTO_INCREMENT ,
    name VARCHAR(200) NOT NULL ,
    username VARCHAR(200) NOT NULL UNIQUE ,
    encrypted_password VARCHAR(1000) NOT NULL ,
    address VARCHAR(200) NOT NULL DEFAULT 'Sin dirección',
    phone VARCHAR(20) NOT NULL DEFAULT '0000000000',
    email VARCHAR(50) NOT NULL ,
    cui VARCHAR(13) NOT NULL DEFAULT '0000000000000',
    birthdate DATETIME NOT NULL ,
    balance DECIMAL(10,2) NOT NULL DEFAULT 100,
    type VARCHAR(50) NOT NULL  ,
    initial_setup BOOLEAN NOT NULL DEFAULT FALSE,
    PRIMARY KEY (id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE patients (
    id INT NOT NULL AUTO_INCREMENT ,
    user_id INT NOT NULL ,
    PRIMARY KEY (id) ,
    FOREIGN KEY (user_id) REFERENCES users(id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE medics (
    id INT NOT NULL AUTO_INCREMENT ,
    user_id INT NOT NULL ,
    PRIMARY KEY (id) ,
    FOREIGN KEY (user_id) REFERENCES users(id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE medic_schedules (
    id INT NOT NULL AUTO_INCREMENT ,
    medic_id INT NOT NULL ,
```

```
    schedule VARCHAR(30) NOT NULL ,
    PRIMARY KEY (id) ,
    FOREIGN KEY (medic_id) REFERENCES medics(id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE laboratories (
    id INT NOT NULL AUTO_INCREMENT ,
    user_id INT NOT NULL ,
    PRIMARY KEY (id) ,
    FOREIGN KEY (user_id) REFERENCES users(id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE administrator (
    id INT NOT NULL AUTO_INCREMENT ,
    user_id INT NOT NULL ,
    commission DECIMAL(3,2) NOT NULL DEFAULT 0.04,
    PRIMARY KEY (id) ,
    FOREIGN KEY (user_id) REFERENCES users(id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);


CREATE TABLE specialities (
      id INT NOT NULL AUTO_INCREMENT ,
      name VARCHAR(200) NOT NULL ,
      description VARCHAR(1000) NOT NULL ,
      PRIMARY KEY (id),
      created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
      updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE appointments (
    id INT NOT NULL AUTO_INCREMENT ,
    medic_id INT NOT NULL ,
    speciality_id INT NOT NULL ,
    patient_id INT NOT NULL ,
    date DATETIME NOT NULL ,
    schedule VARCHAR(30) NOT NULL ,
    status VARCHAR(50) NOT NULL DEFAULT 'AGENDADA' ,
    report VARCHAR(1000) ,
    price DECIMAL(10,2) NOT NULL ,
    PRIMARY KEY (id) ,
```

```
    FOREIGN KEY (medic_id) REFERENCES medics(id) ,
    FOREIGN KEY (patient_id) REFERENCES patients(id),
    FOREIGN KEY (speciality_id) REFERENCES specialities(id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE appointment_tests(
    id INT NOT NULL AUTO_INCREMENT ,
    test_name VARCHAR(200) NOT NULL ,
    appointment_id INT NOT NULL ,
    date DATETIME NOT NULL ,
    PRIMARY KEY (id) ,
    FOREIGN KEY (appointment_id) REFERENCES appointments(id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE test_requests(
    id INT NOT NULL AUTO_INCREMENT ,
    patient_id INT NOT NULL ,
    laboratory_id INT NOT NULL ,
    status VARCHAR(50) NOT NULL ,
    price DECIMAL(10,2) NOT NULL ,
    date DATETIME NOT NULL ,
    PRIMARY KEY (id) ,
    FOREIGN KEY (patient_id) REFERENCES patients(id) ,
    FOREIGN KEY (laboratory_id) REFERENCES laboratories(id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE test_request_details (
    id INT NOT NULL AUTO_INCREMENT ,
    test_request_id INT NOT NULL ,
    test_type_id INT NOT NULL ,
    PRIMARY KEY (id) ,
    FOREIGN KEY (test_request_id) REFERENCES test_requests(id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE medics_specialities (
    medic_id INT NOT NULL ,
    speciality_id INT NOT NULL ,
    price DECIMAL(10,2) NOT NULL ,
    PRIMARY KEY (medic_id, speciality_id) ,
```

```
    FOREIGN KEY (medic_id) REFERENCES medics(id) ,
    FOREIGN KEY (speciality_id) REFERENCES specialities(id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE test_types(
    id INT NOT NULL AUTO_INCREMENT ,
    name VARCHAR(200) NOT NULL ,
    description VARCHAR(1000) NOT NULL ,
    PRIMARY KEY (id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE laboratory_test_type(
    laboratory_id INT NOT NULL ,
    test_type_id INT NOT NULL ,
    price DECIMAL(10,2) NOT NULL ,
    PRIMARY KEY (laboratory_id, test_type_id) ,
    FOREIGN KEY (laboratory_id) REFERENCES laboratories(id) ,
    FOREIGN KEY (test_type_id) REFERENCES test_types(id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE new_test_type_requests(
    id INT NOT NULL AUTO_INCREMENT ,
    laboratory_id INT NOT NULL ,
    status VARCHAR(50) NOT NULL ,
    name VARCHAR(200) NOT NULL ,
    description VARCHAR(1000) NOT NULL ,
    PRIMARY KEY (id) ,
    FOREIGN KEY (laboratory_id) REFERENCES laboratories(id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE new_speciality_requests(
    id INT NOT NULL AUTO_INCREMENT ,
    medic_id INT NOT NULL ,
    status VARCHAR(50) NOT NULL ,
    name VARCHAR(200) NOT NULL ,
    description VARCHAR(1000) NOT NULL ,
    PRIMARY KEY (id) ,
    FOREIGN KEY (medic_id) REFERENCES medics(id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
```

```
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE balance_recharges(
    id INT NOT NULL AUTO_INCREMENT ,
    user_id INT NOT NULL ,
    amount DECIMAL(10,2) NOT NULL ,
    PRIMARY KEY (id) ,
    FOREIGN KEY (user_id) REFERENCES users(id),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ,
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

INSTRUCCIONES PARA EJECUCIÓN

Para poder ejecutar este proyecto, es necesario tener instalado NodeJS y Java en la computadora que se desea usar.
Node puede ser instalado a través de la herramienta de línea de comando llamada nvm la cual puede ser descargada del repositorio oficial en GitHub

Java por el otro lado, puede ser descargado e instalado con un simple instalador que es proveído por Oracle en su respectiva página web

Para comenzar, para poner en ejecución la aplicación de la parte interfaz gráfica, es necesario tener el código de este.
Luego, se navega a traves de la linea de comandos al directorio donde se encuentra este. Y luego simplemente debe correrse el comando `ng serve` y esto pondrá en marcha la aplicación

Para el lado de los datos, se debe tener instalado algún IDE del lenguaje de programación de Java, en este caso se recomienda IntelliJ de JetBrains
Luego, es necesario abrir el directorio que tiene de nombre “p2ipc2backend” en este programa, el cual se encargará de juntar estos archivos y permitir correrlos. Por último, se debe seleccionar la opción de correr el programa completo en el IDE y se pondrá en ejecución

También es necesario crear la base de datos, para esto es necesario tener el sistema de base de datos relacionales MySQL, el cual tambien puede ser descargado desde su página oficial. Luego de haberlo instalado y configurado, solo es necesario navegar por medio de la linea de comandos al directorio donde se encuentren los archivos database.sql y test_seeds.sql, y luego abrir el programa de terminal mysqld y ejecutar el comando `source ./database.sql y source ./test_seeds.sql`

Luego de haber hecho estas configuraciones, la aplicación debería estar lista para usarse en el navegador web.