



UNIVERSIDAD DON BOSCO

**DISEÑO Y PROGRAMACION DE SOFTWARE
MULTIPLATAFORMA**

DPS941 G01T

ING. ALEXANDER SIGÜENZA

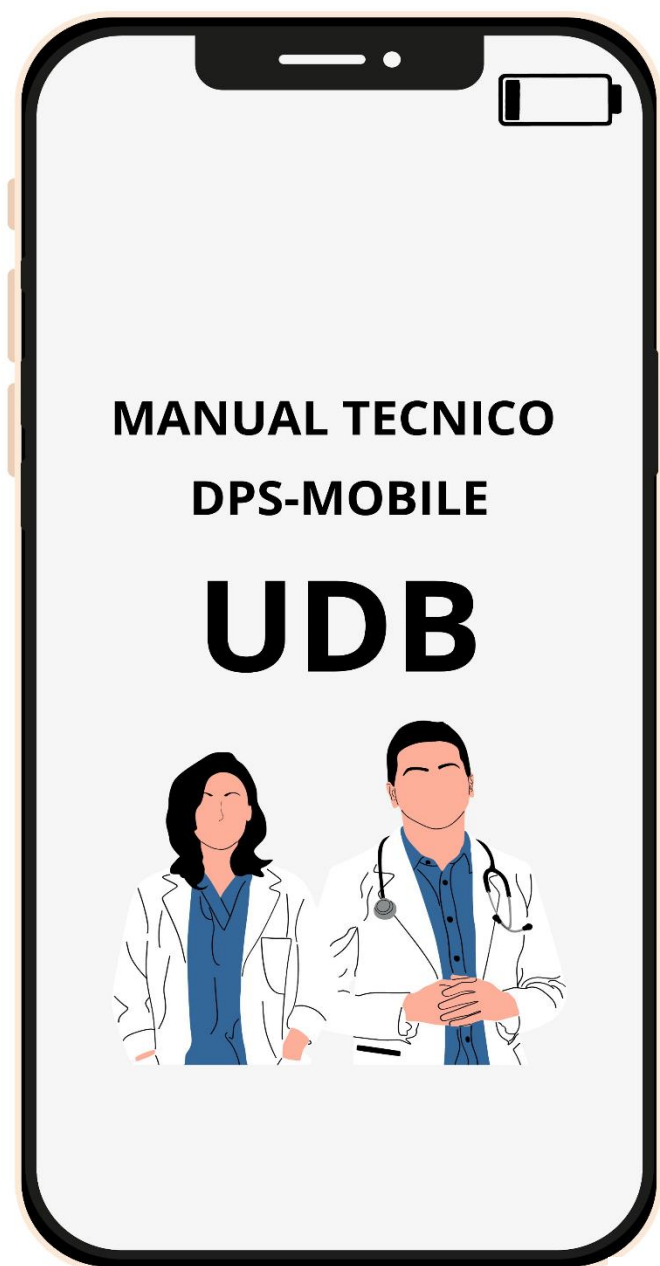
MANUAL TECNICO

DPS-Mobile

No	INTEGRANTES	CARNÉ
1	Caleb Verenice López Gutiérrez	LG211551
2	René Saúl Jovel Calderón	JC211517
3	Bryan Antonio Mena Cortez	MC211787

Contenido

1.INTRODUCCION.	4
2.OBJETIVOS.	4
3.REQUERIMIENTOS TECNICOS DEL SISTEMA.	4
4.EXPLICACION GENERAL DE LAS FASES DEL SISTEMA.....	5
5. DIAGRAMA DEL SISTEMA.....	6
6.DICCIONARIO DE DATOS.....	7
Tabla doctor	7
Tabla cita.....	7
Paciente.....	7
7.INICIACION O IMPLEMENTACION DEL SISTEMA.....	8



1.INTRODUCCION.

En este proyecto presentamos la solución informática por medio de una App la cual hemos titulado DPS-Mobile, esta mantiene características nativas, en donde se ha desarrollado con tecnologías orientadas a ese entorno como lo es React Native, Node Js y el lenguaje de programación Javascript.

2.OBJETIVOS.

Objetivo General:

Desarrollar una aplicación móvil la cual sea eficiente para la solución informática del problema de registro y organización de pacientes en el hospital.

Objetivos Específicos:

- Realizar las interfaces adecuadas para la creación de la app Mobile.
- Hacer las pruebas funcionales, para una mejor entrega, por medio del emulador Expo.

3.REQUERIMIENTOS TECNICOS DEL SISTEMA.

- Podría contar con los siguientes recursos en la computadora u ordenador para correrlo eficientemente:

Intel(R) Core (TM) i5-6200U CPU @ 2.30GHz 2.40 GHz

RAM 8.00 GB (7.88 GB usable)

64-bit operating system, x64-based processor

Disco duro de 264 GB

- Programas o tecnologías para edición del Código

Visual studio code o un IDE

Node JS

- Contar con las siguientes características para correrlo en el teléfono:

Internet

Aplicación Expo Go

4.EXPLICACION GENERAL DE LAS FASES DEL SISTEMA.

Fase 1: Registro en la aplicación

```
JS RegistroUsuario.js X
views > JS RegistroUsuario.js > ...
1 import * as React from 'react';
2 import { View, Alert,TextInput, Text, StyleSheet, Pressable } from 'react-native';
3
4 import LogoExample from '../components/Logo.js';
5
6 function RegistroUsuario() {
7   return (
8     <View style={styles.container}>
9
10      <View style={styles.view}>
11        <LogoExample/>
12      </View>
13
14      <Text style={styles.title}>
15        Registrarse
16      </Text>
17
18      <View style={styles.card}>
19
20        <TextInput style={styles.input}
21          placeholder=" Nombre"/>
22
23        <TextInput style={styles.input}
24          placeholder=" Apellidos"/>
25
26        <TextInput style={styles.input}
27          placeholder=" Correo"/>
28
29        <Pressable style={styles.button} onPress={() => navigation.navigate('Citas')}>
30          Iniciar Sesión
31        </Pressable>
32      </View>
33    </View>
34  );
35}
```

Fase 2: Inicio de sesión

```
File Edit Selection View Go Run ... prueba-app
EXPLORER
PRUEBA-APP
  > .expo
  > assets
  > components
  > node_modules
  > views
    JS CitasVistas.js
    JS CrearCitas.js
    JS InicioVistas.js
    JS LoginUsuario.js
    JS PacienteRegistros.js
    JS PacientesVistas.js
    JS RegistroUsuario.js
    .gitignore
  JS App.js
  app.json
  babel.config.js
  package-lock.json
  package.json
  OUTLINE
  TIMELINE
  PROBLEMS
  OUTPUT
  DEBUG CONSOLE
  TERMINAL
  PORTS
  node + v ...
  ...

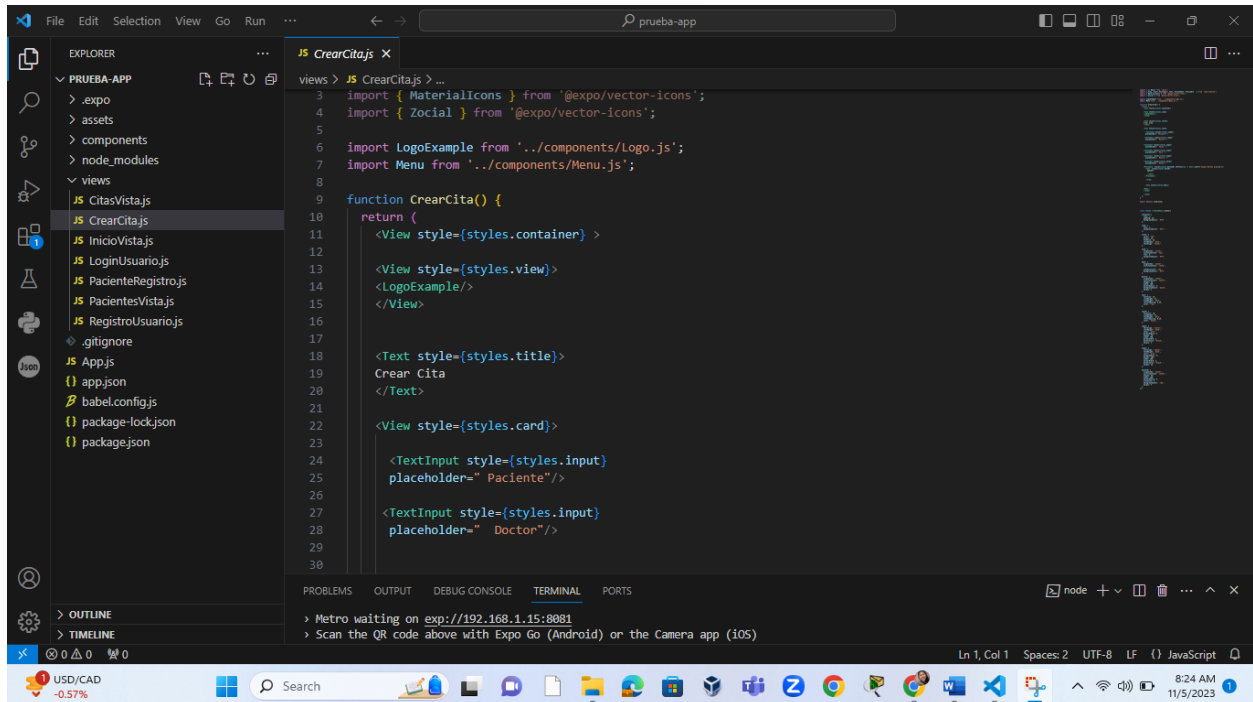
JS LoginUsuario.js X
views > JS LoginUsuario.js > ...
1 import * as React from 'react';
2 import { View, Alert,TextInput, Text, StyleSheet, Pressable } from 'react-native';
3
4 import LogoExample from '../components/Logo.js';
5
6 function LoginUsuario({ navigation }) {
7   return (
8     <View style={styles.container}>
9
10      <View style={styles.view}>
11        <LogoExample/>
12      </View>
13
14      <Text style={styles.title}>
15        Inicio de Sesión
16      </Text>
17
18      <View style={styles.card}>
19
20        <TextInput style={styles.input}
21          placeholder=" Correo"/>
22
23        <TextInput secureTextEntry={true} style={styles.input}
24          placeholder=" Contraseña"/>
25
26        <Pressable style={styles.button} onPress={() => navigation.navigate('Citas')}>
27          Iniciar Sesión
28        </Pressable>
29      </View>
30    </View>
31  );
32}
```

Metro waiting on exp://192.168.1.15:8081
Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

Ln 1, Col 1 Spaces: 2 UTF-8 LF {} JavaScript

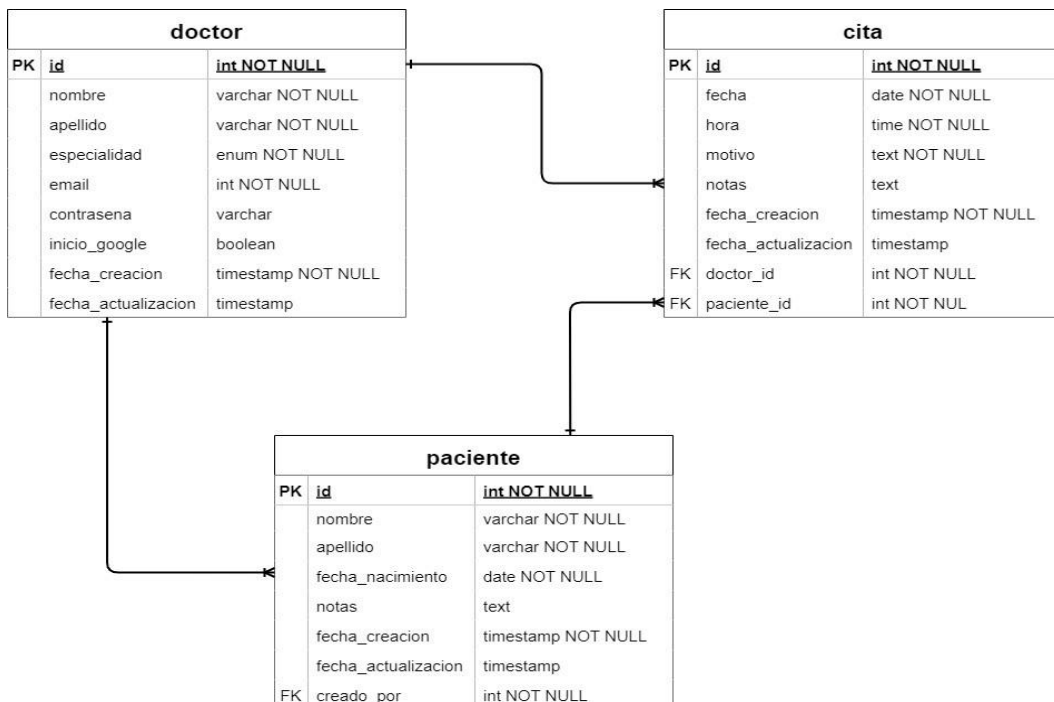
USD/GBP -1.40% 8:24 AM 11/5/2023

Fase 3: Creación de las citas



```
1 import { MaterialIcons } from '@expo/vector-icons';
2 import { Zocial } from '@expo/vector-icons';
3
4 import LogoExample from '../components/Logo.js';
5 import Menu from '../components/Menu.js';
6
7 function CrearCita() {
8   return (
9     <View style={styles.container}>
10       <View style={styles.view}>
11         <LogoExample/>
12       </View>
13
14       <Text style={styles.title}>
15         Crear Cita
16       </Text>
17
18       <View style={styles.card}>
19         <TextInput style={styles.input}
20           placeholder=" Paciente"/>
21
22         <TextInput style={styles.input}
23           placeholder=" Doctor"/>
24       </View>
25     </View>
26   );
27 }
28
29 export default CrearCita;
```

5. DIAGRAMA DEL SISTEMA



6.DICCIONARIO DE DATOS.

Tabla doctor

Campo	Tipo de dato	Descripción
id	int	Llave primaria
nombre	Varchar o string	Nombre del doctor
apellido	Varchar o string	Apellido del doctor
especialidad	Enum	Especialidad que atiende
email	Varchar o string	Correo electrónico del doctor
contraseña	Varchar o string	Contraseña del correo del doctor
inicio_google	Boolean	Registro de identificación de Google.
fecha_creacion	timestamp	Fecha de la creación del registro
fecha_actualizacion	timestamp	Fecha de la actualización del registro

Tabla cita

Campo	Tipo de dato	Descripción
id	int	Llave primaria
fecha	Date	Fecha de la cita
hora	time	Tiempo u hora especifica de la cita
motivo	Text	Motivo de la cita
notas	Timestamp	Notas de la cita
fecha_creacion	Timestamp	Fecha de la creación del registro.
fecha_actualizacion	Int	Fecha de la actualización del registro.
doctor	Int	Llave foránea
paciente	int	Llave foránea

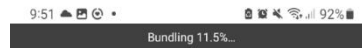
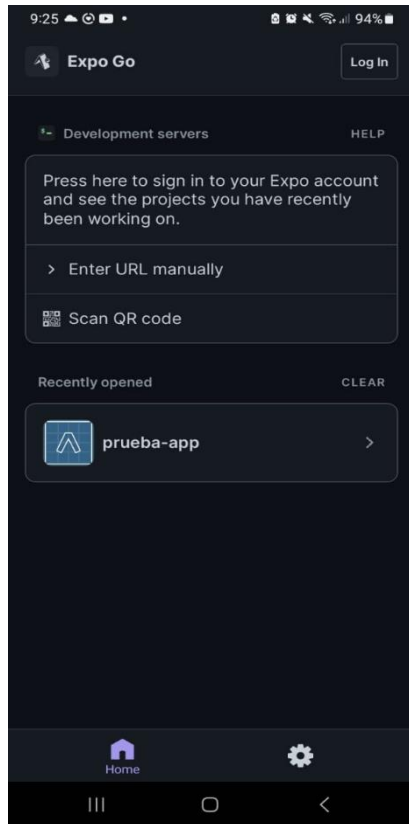
Paciente

Campo	Tipo de dato	Descripción
id	int	Llave primaria
nombre	Varchar	Nombre del paciente
apellido	Date	Apellido del paciente
notas	text	Notas del paciente
fecha_nacimiento	Timestamp	Fecha de nacimiento
fecha_actualizacion	Timestamp	Fecha de la actualización del registro.
creado_por	Int	La creación del registro

7.INICIACION O IMPLEMENTACION DEL SISTEMA.

Paso 1:

Abrir la aplicación de expo y escanear el Código QR, después de eso se inicia la aplicación.



UDB



10:29 AM

38%

Inicio

Paso 2: ingresar al sistema ya sea al registro o al inicio de sesión si ya se tienen los datos para ingresar.

UDB

Iniciar Sesión

Registrarse

Paso 3: Nos registramos e iniciamos sesión

10:30 AM

38%

← Registro

UDB

Registrarse

Nombre

Apellidos

Especialidad

Correo

Contraseña

Registrarse

0

Registrarse con Gmail

10:29 AM

38%

← Login

UDB

Inicio de Sesión

Correo

Contraseña

Iniciar Sesión

0

Iniciar sesión con Gmail

Paso 4: El sistema nos envía al menú de opciones en donde podemos registrar a los pacientes.

5:56

52%

← Crear-Paciente

UDB

Registrar Paciente

Nombres

Apellidos

Fecha de nacimiento

Notas

Registrar

Citas Pacientes Salir

5:56

52%

← Pacientes

UDB

Pacientes

Registrar Nuevo Paciente

Citas Pacientes Salir

5:56

52%

← Pacientes

UDB

Pacientes

Registrar Nuevo Paciente

Citas Pacientes Salir

Paso 7: Puedes consultar las citas o salir de la aplicación.

5:57





Crear-Paciente

UDB

Registrar Paciente

Caleb Verenice

Lopez Gutierrez

9 de julio 2000

Hipertenso

Registrar

 Citas

 Pacientes

 Salir