

Códigos de proyecto de reserva de hotel

Página: Inicio

Archivo: inicio.page.html

código:

```
<ion-header>

<ion-toolbar>

<ion-title>Inicio</ion-title>

</ion-toolbar>

</ion-header>


<ion-content>

<ion-button routerLink="/registro-cliente" expand="full" color="primary">
Registrar Cliente
</ion-button>

<ion-button routerLink="/enc-reserva" expand="full" color="secondary">
Gestionar Reserva
</ion-button>

</ion-content>
```

Archivo: inicio.page.scss

Código:

```
/* Estilos específicos para la página de inicio */
page-inicio {
  ion-content {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
```

```
background: linear-gradient(to right, $primary-color, $secondary-color);  
}
```

```
ion-button {  
width: 80%;  
margin: 10px 0;  
}  
}
```

Pagina: Registro cliente

Archivo: registro-cliente.page.html

codigo:

```
<ion-header>  
<ion-toolbar>  
<ion-title>Registrar Cliente</ion-title>  
</ion-toolbar>  
</ion-header>  
  
<ion-content>  
<form (ngSubmit)="registerCliente()">  
<ion-item>  
<ion-label position="floating">Número de Documento</ion-label>  
<ion-input type="text" [(ngModel)]="cliente.numeroDocumento"  
name="numeroDocumento"></ion-input>  
</ion-item>  
<ion-item>  
<ion-label>Tipo de Documento</ion-label>  
<ion-select [(ngModel)]="cliente.tipoDocumento" name="tipoDocumento">
```

```
<ion-select-option value="CC">CC</ion-select-option>
<ion-select-option value="CE">CE</ion-select-option>
<ion-select-option value="TI">TI</ion-select-option>
<ion-select-option value="Pasaporte">Pasaporte</ion-select-option>
</ion-select>
</ion-item>
<ion-item>
<ion-label position="floating">Nombres</ion-label>
<ion-input type="text" [(ngModel)]="cliente.nombres" name="nombres"></ion-input>
</ion-item>
<ion-item>
<ion-label position="floating">Apellidos</ion-label>
<ion-input type="text" [(ngModel)]="cliente.apellidos" name="apellidos"></ion-input>
</ion-item>
<ion-item>
<ion-label position="floating">Email</ion-label>
<ion-input type="email" [(ngModel)]="cliente.email" name="email"></ion-input>
</ion-item>
<ion-item>
<ion-label position="floating">Dirección</ion-label>
<ion-input type="text" [(ngModel)]="cliente.direccion" name="direccion"></ion-input>
</ion-item>
<ion-item>
<ion-label position="floating">Celular</ion-label>
<ion-input type="text" [(ngModel)]="cliente.celular" name="celular"></ion-input>
</ion-item>
<ion-button expand="full" type="submit">Registrar Cliente</ion-button>
</form>
```

</ion-content>

Archivo: registro-cliente.page.scss

codigo:

```
/* src/app/clientes/clientes.page.scss */
```

```
/* Estilos específicos para la página de registro de clientes */
```

```
page-clientes {
```

```
  ion-content {
```

```
    display: flex;
```

```
    flex-direction: column;
```

```
    justify-content: center;
```

```
    align-items: center;
```

```
    background-color: $background-color;
```

```
  }
```

```
  form {
```

```
    width: 100%;
```

```
    max-width: 500px;
```

```
    padding: 20px;
```

```
    background-color: white;
```

```
    border-radius: 10px;
```

```
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
```

```
  }
```

```
  ion-item {
```

```
    margin-bottom: 15px;
```

```
  }
```

```
  ion-button {
```

```
    width: 100%;
```

```
background-color: $primary-color;

color: white;

border-radius: $button-radius;

}

}
```

Archivo: registro-cliente.page.ts
codigo:

```
import { Component, OnInit } from '@angular/core';
import { DatabaseService } from '../services/database.service';
import { Router } from '@angular/router';

@Component({
  selector: 'app-clientes',
  templateUrl: './clientes.page.html',
  styleUrls: ['./clientes.page.scss'],
})
export class ClientesPage implements OnInit {

  cliente = {

    numeroDocumento: "",

    tipoDocumento: "",

    nombres: "",

    apellidos: "",

    email: "",

    direccion: "",

    celular: ""

  };

  constructor(private dbService: DatabaseService, private router: Router) { }
```

```

ngOnInit() { }

// Método para registrar cliente en la base de datos
async registerCliente() {
  await this.dbService.addCliente(this.cliente);
  // Redirigir a la página de inicio después de registrar
  this.router.navigate(['/inicio']);
}
}

```

Página: Encabezado de reserva

Archivo:enc-reserva.page.html

codigo:

```

<ion-header>

<ion-toolbar>

<ion-title>Encabezado de Reserva</ion-title>

</ion-toolbar>

</ion-header>

<ion-content>

<ion-item>

<ion-label position="floating">Número de Documento del Cliente</ion-label>

<ion-input type="text" [(ngModel)]="numeroDocumento" name="numeroDocumento"
(ionBlur)="buscarCliente()"></ion-input>

</ion-item>

<!-- Mostrar información del cliente si está disponible -->

```

```
<ion-list *ngIf="cliente">
  <ion-item>
    <ion-label>Nombres:</ion-label>
    <ion-text>{{ cliente.nombres }}</ion-text>
  </ion-item>
  <ion-item>
    <ion-label>Apellidos:</ion-label>
    <ion-text>{{ cliente.apellidos }}</ion-text>
  </ion-item>
  <ion-item>
    <ion-label>Email:</ion-label>
    <ion-text>{{ cliente.email }}</ion-text>
  </ion-item>
  <ion-item>
    <ion-label>Dirección:</ion-label>
    <ion-text>{{ cliente.direccion }}</ion-text>
  </ion-item>
  <ion-item>
    <ion-label>Celular:</ion-label>
    <ion-text>{{ cliente.celular }}</ion-text>
  </ion-item>
</ion-list>

<ion-item>
  <ion-label position="floating">Nombre del Hotel</ion-label>
  <ion-input type="text" [(ngModel)]="nombreHotel" name="nombreHotel"
    (ionBlur)="buscarHotel()"></ion-input>
</ion-item>
```

<!-- Mostrar información del hotel si está disponible -->

<ion-list *ngIf="hotel">

<ion-item>

<ion-label>ID:</ion-label>

<ion-text>{{ hotel.id }}</ion-text>

</ion-item>

<ion-item>

<ion-label>Email:</ion-label>

<ion-text>{{ hotel.email }}</ion-text>

</ion-item>

<ion-item>

<ion-label>Teléfono:</ion-label>

<ion-text>{{ hotel.telefono }}</ion-text>

</ion-item>

<ion-item>

<ion-label>Dirección:</ion-label>

<ion-text>{{ hotel.direccion }}</ion-text>

</ion-item>

<ion-item>

<ion-label>Ciudad:</ion-label>

<ion-text>{{ hotel.ciudad }}</ion-text>

</ion-item>

<ion-item>

<ion-label>País:</ion-label>

<ion-text>{{ hotel.pais }}</ion-text>

</ion-item>

<ion-item>


```
<ion-label>Estrellas:</ion-label>
<ion-text>{{ hotel.estrellas }}</ion-text>
</ion-item>
</ion-list>
```

```
<!-- Botón para continuar al detalle de la reserva -->
```

```
<ion-button expand="full" routerLink="/det-reserva">Continuar al Detalle de
Reserva</ion-button>
</ion-content>
```

Archivo:enc-reserva.page.scss

codigo:

```
/* src/app/enc-reserva/enc-reserva.page.scss */
```

```
/* Estilos específicos para la página de encabezado de reserva */
```

```
page-enc-reserva {
  ion-content {
    display: flex;
    flex-direction: column;
    justify-content: center;
    align-items: center;
    background-color: $background-color;
  }
}
```

```
ion-item {
  width: 100%;
  max-width: 500px;
```

```
margin-bottom: 15px;
}
```

```
ion-list {
  width: 100%;
  max-width: 500px;
  padding: 20px;
  background-color: white;
  border-radius: 10px;
  box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
  margin-top: 20px;
}
```

```
ion-button {
  width: 100%;
  max-width: 500px;
  margin-top: 20px;
  background-color: $secondary-color;
  color: white;
  border-radius: $button-radius;
}
}
```

Archivo:enc-reserva.page.ts

codigo:

```
import { Component, OnInit } from '@angular/core';
import { DatabaseService } from '../services/database.service';
```

```

@Component({
  selector: 'app-enc-reserva',
  templateUrl: './enc-reserva.page.html',
  styleUrls: ['./enc-reserva.page.scss'],
})
export class EncReservaPage implements OnInit {
  numeroDocumento: string;
  cliente: any;
  nombreHotel: string;
  hotel: any;

  constructor(private dbService: DatabaseService) {}

  ngOnInit() {}

  // Método para buscar cliente por documento
  async buscarCliente() {
    this.cliente = await this.dbService.getClienteporDocumento(this.numeroDocumento);
  }

  // Método para buscar hotel por nombre
  async buscarHotel() {
    this.hotel = await this.dbService.getHotelPorNombre(this.nombreHotel);
  }
}

```

Página: Detalle de reserva

Archivo:det-reserva.page.html

codigo:

```
<ion-header>

<ion-toolbar>

<ion-title>Detalle de Reserva</ion-title>

</ion-toolbar>

</ion-header>


<ion-content>

<ion-item>

<ion-label>Habitación</ion-label>

<ion-select [(ngModel)]="habitacionSeleccionada" name="habitacionSeleccionada"
(ionChange)="mostrarDetallesHabitacion()">

<ion-select-option *ngFor="let habitacion of habitaciones" [value]="habitacion">{{
habitacion.numero }}</ion-select-option>

</ion-select>

</ion-item>


<!-- Mostrar información de la habitación seleccionada -->

<ion-list *ngIf="habitacionSeleccionada">

<ion-item>

<ion-label>Número de Habitación:</ion-label>

<ion-text>{{ habitacionSeleccionada.numero }}</ion-text>

</ion-item>

<ion-item>

<ion-label>Capacidad:</ion-label>

<ion-text>{{ habitacionSeleccionada.capacidad }}</ion-text>

</ion-item>

<ion-item>
```

```

<ion-label>Disponibilidad:</ion-label>
<ion-text>{{ habitacionSeleccionada.disponibilidad }}</ion-text>
</ion-item>
<ion-item>
<ion-label>Tipo de Habitación:</ion-label>
<ion-text>{{ habitacionSeleccionada.tipo }}</ion-text>
</ion-item>
<ion-item>
<ion-label>Nombre de la Suite:</ion-label>
<ion-text>{{ habitacionSeleccionada.suite }}</ion-text>
</ion-item>
<ion-item>
<ion-label>Número de Camas:</ion-label>
<ion-text>{{ habitacionSeleccionada.camass }}</ion-text>
</ion-item>
<ion-item>
<ion-label>Precio:</ion-label>
<ion-text>{{ habitacionSeleccionada.precio }}</ion-text>
</ion-item>
</ion-list>

<ion-button expand="full" (click)="confirmarReserva()">Confirmar Reserva</ion-
button>
</ion-content>

```

Archivo:det-reserva.page.scss

codigo:

```
/* src/app/detalle-reserva/detalle-reserva.page.scss */
```

/* Estilos específicos para la página de detalle de reserva */

page-detalle-reserva {

 ion-content {

 display: flex;

 flex-direction: column;

 justify-content: center;

 align-items: center;

 background-color: \$background-color;

 }

 ion-item {

 width: 100%;

 max-width: 500px;

 margin-bottom: 15px;

 }

 ion-list {

 width: 100%;

 max-width: 500px;

 padding: 20px;

 background-color: white;

 border-radius: 10px;

 box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);

 margin-top: 20px;

 }

 ion-button {

```
width: 100%;  
max-width: 500px;  
margin-top: 20px;  
background-color: $primary-color;  
color: white;  
border-radius: $button-radius;  
}  
}
```

Archivo:det-reserva.page.ts

codigo:

```
import { Component, OnInit } from '@angular/core';  
import { DatabaseService } from '../services/database.service';  
import { Router } from '@angular/router';  
  
@Component({  
  selector: 'app-detalle-reserva',  
  templateUrl: './detalle-reserva.page.html',  
  styleUrls: ['./detalle-reserva.page.scss'],  
})  
export class DetalleReservaPage implements OnInit {  
  habitaciones: any[] = [];  
  habitacionSeleccionada: any;  
  
  constructor(private dbService: DatabaseService, private router: Router) {}  
  
  ngOnInit() {  
    this.cargarHabitaciones();  
  }  
}
```

```
}
```

```
// Método para cargar todas las habitaciones
```

```
async cargarHabitaciones() {
```

```
this.habitaciones = await this.dbService.getHabitaciones();
```

```
}
```

```
// Método para mostrar los detalles de la habitación seleccionada
```

```
mostrarDetallesHabitacion() {
```

```
// Aquí se puede agregar lógica adicional si es necesario
```

```
}
```

```
// Método para confirmar y guardar la reserva
```

```
async confirmarReserva() {
```

```
const reserva = {
```

```
  clientId: this.cliente.id,
```

```
  hotelId: this.hotel.id,
```

```
  habitacionId: this.habitacionSeleccionada.id,
```

```
  fechaEntrada: new Date().toISOString(),
```

```
  fechaSalida: new Date().toISOString() // Actualizar según la lógica de la reserva
```

```
};
```

```
await this.dbService.addReserva(reserva);
```

```
// Redirigir a la página de inicio después de confirmar la reserva
```

```
this.router.navigate(['/inicio']);
```

```
}
```

```
}
```


Adicionalmente se creo un directorio para verifica que tu servicio de base de datos (src/app/services/database.service.ts) esté configurado para manejar las operaciones necesarias.

Archivo: database.service.ts

Codigo:

```
import { Injectable } from '@angular/core';
import { Platform } from '@ionic/angular';
import { SQLite, SQLiteObject } from '@awesome-cordova-plugins/sqlite/ngx';
```

```
@Injectable({
  providedIn: 'root'
})
export class DatabaseService {
  private dbInstance: SQLiteObject;

  constructor(private platform: Platform, private sqlite: SQLite) {
    this.platform.ready().then(() => {
      this.createDatabase();
    });
  }

  async createDatabase() {
    try {
      this.dbInstance = await this.sqlite.create({
        name: 'hotel.db',
        location: 'default'
      });
      await this.createTables();
    }
  }
}
```

```
} catch (error) {  
  console.error('Error creating database', error);  
}  
}
```

```
async createTables() {  
  try {  
    await this.dbInstance.executeSql(`  
      CREATE TABLE IF NOT EXISTS clientes (  
        id INTEGER PRIMARY KEY AUTOINCREMENT,  
        numeroDocumento TEXT,  
        tipoDocumento TEXT,  
        nombres TEXT,  
        apellidos TEXT,  
        email TEXT,  
        direccion TEXT,  
        celular TEXT  
      );  
    ` , []);  
    await this.dbInstance.executeSql(`  
      CREATE TABLE IF NOT EXISTS hoteles (  
        id INTEGER PRIMARY KEY AUTOINCREMENT,  
        nombre TEXT,  
        direccion TEXT,  
        telefono TEXT,  
        ciudad TEXT,  
        pais TEXT,  
        estrellas INTEGER,
```

email TEXT

);

` , []);

await this.dbInstance.executeSql(`

CREATE TABLE IF NOT EXISTS habitaciones (

id INTEGER PRIMARY KEY AUTOINCREMENT,

hotelId INTEGER,

numero TEXT,

capacidad INTEGER,

disponibilidad TEXT,

tipo TEXT,

suite TEXT,

camas INTEGER,

precio REAL,

FOREIGN KEY(hotelId) REFERENCES hoteles(id)

);

` , []);

await this.dbInstance.executeSql(`

CREATE TABLE IF NOT EXISTS reservas (

id INTEGER PRIMARY KEY AUTOINCREMENT,

clienteId INTEGER,

hotelId INTEGER,

habitacionId INTEGER,

fechaEntrada TEXT,

fechaSalida TEXT,

FOREIGN KEY(clienteId) REFERENCES clientes(id),

FOREIGN KEY(hotelId) REFERENCES hoteles(id),

FOREIGN KEY(habitacionId) REFERENCES habitaciones(id)

```

);
`, []);
} catch (error) {
  console.error('Error creating tables', error);
}
}

```

```

async addCliente(cliente) {
  const data = [cliente.numeroDocumento, cliente.tipoDocumento, cliente.nombres,
    cliente.apellidos, cliente.email, cliente.direccion, cliente.celular];

  return this.dbInstance.executeSql('INSERT INTO clientes (numeroDocumento,
    tipoDocumento, nombres, apellidos, email, direccion, celular) VALUES (?, ?, ?, ?, ?, ?,
    ?)', data)

    .then(res => {
      return res;
    })

    .catch(e => {
      console.error('Error adding cliente', e);
      throw e;
    });
}

```

```

async getClientePorDocumento(numeroDocumento) {
  return this.dbInstance.executeSql('SELECT * FROM clientes WHERE
    numeroDocumento = ?', [numeroDocumento])

    .then(res => {
      if (res.rows.length > 0) {
        return res.rows.item(0);
      }
    })
}

```

```
return null;

})

.catch(e => {
  console.error('Error getting cliente', e);
  throw e;
});
}
```

```
async getHotelPorNombre(nombre) {
  return this.dbInstance.executeSql('SELECT * FROM hoteles WHERE nombre = ?',
    [nombre])
    .then(res => {
      if (res.rows.length > 0) {
        return res.rows.item(0);
      }
      return null;
    })
    .catch(e => {
      console.error('Error getting hotel', e);
      throw e;
    });
}
```

```
async getHabitaciones() {
  return this.dbInstance.executeSql('SELECT * FROM habitaciones', [])
    .then(res => {
      const habitaciones = [];
      for (let i = 0; i < res.rows.length; i++) {
```

```
habitaciones.push(res.rows.item(i));  
}  
return habitaciones;  
})  
.catch(e => {  
  console.error('Error getting habitaciones', e);  
  throw e;  
});  
}
```

```
async addReserva(reserva) {  
  const data = [reserva.clientId, reserva.hotelId, reserva.habitacionId,  
    reserva.fechaEntrada, reserva.fechaSalida];  
  return this.dbInstance.executeSql('INSERT INTO reservas (clientId, hotelId,  
    habitacionId, fechaEntrada, fechaSalida) VALUES (?, ?, ?, ?, ?)', data)  
    .then(res => {  
      return res;  
    })  
    .catch(e => {  
      console.error('Error adding reserva', e);  
      throw e;  
    });  
}
```