**Diseño de Software**

**App Prográmate en Medellín Versión 1.0 2025**

**Cristian David López Cardona**

**Dayana García Rodríguez**

**Brian Alexis Cadavid Castaño**

**Institución Universitaria Pascual Bravo**

**Facultad de Ingeniería**

**Profesora Claudia María Giraldo Flórez**

**Medellín, Colombia 25 de Marzo de 2025**

**Selección del Diseño Arquitectónico**

**Patrón Arquitectónico**

Para el desarrollo de la aplicación, se utilizará el patrón **Modelo-Vista-Controlador (MVC)**, que permitirá separar la lógica de negocio de la interfaz de usuario, facilitando la escalabilidad y el mantenimiento. Este patrón permite un desarrollo modular y flexible, asegurando una buena organización del código.

**Tecnologías Utilizadas**

* **Frontend:** React Native (para compatibilidad con Android e iOS)
* **Backend:** Node.js con Express
* **Base de Datos:** SQL Server con Entity Framework
* **Autenticación:** Firebase Authentication
* **Hosting del Backend:** AWS o Heroku
* **Geolocalización:** Google Maps API
* **Almacenamiento de Imágenes:** Firebase Storage

**Diagrama General de la Arquitectura**

El sistema seguirá un enfoque cliente-servidor, donde la aplicación móvil se comunicará con el backend a través de una API RESTful. El backend gestionará la autenticación, el acceso a la base de datos y la lógica de negocio.

Diagrama

El contenido generado por IA puede ser incorrecto.

**Diseño Detallado**

**Interfaz de Usuario**

La aplicación tendrá una interfaz intuitiva y atractiva, basada en el diseño mostrado en la imagen de referencia. Se priorizará la usabilidad y accesibilidad.

**Secciones Principales:**

1. **Pantalla de inicio** - Imagen de Medellín y botones de navegación.
2. **Categorías** - Opciones como Restaurantes, Cultura, Diversión y Entretenimiento.
3. **Detalles de los lugares** - Información específica sobre cada sitio, con fotos y reseñas.
4. **Perfil del usuario** - Configuraciones y opciones personales.
5. **Autenticación** - Inicio de sesión y registro con Firebase.
6. **Geolocalización** - Búsqueda de lugares cercanos en un mapa interactivo.

**Flujo de Navegación**

1. Inicio -> Selección de Categoría
2. Selección de Categoría -> Lista de Lugares
3. Lista de Lugares -> Detalle del Lugar
4. Perfil del Usuario -> Configuraciones
5. Buscar en el Mapa -> Detalles de un Lugar

**Diagramas**

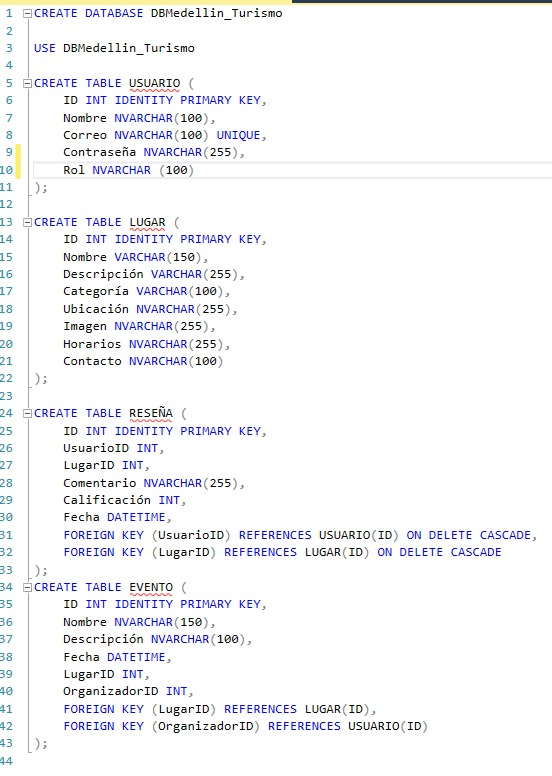
* **Wireframes de Interfaz:**
* **Diagrama de Flujo de Navegación:**

**Diagrama

El contenido generado por IA puede ser incorrecto.**

**Modelo Relacional (SQL Server)**

Se utilizará SQL Server como motor de base de datos relacional con las siguientes tablas:



**Relaciones entre Entidades**

* Un **usuario** puede tener múltiples lugares favoritos.
* Un **lugar** puede tener varias reseñas de distintos usuarios.
* Un **usuario** puede escribir varias reseñas.
* Un **evento** estará vinculado a un lugar específico.

**Diagrama de la Base de Datos (SQL Server)  
  
Diagrama

El contenido generado por IA puede ser incorrecto.**

**Diagrama de Clases**

El diagrama de clases representará las entidades y sus relaciones en la aplicación.

**Clases Principales**

1. **Usuario**
   * Atributos: ID, Nombre, Correo, Contraseña
   * Métodos: registrar(), iniciarSesion(), actualizarPerfil()
2. **Lugar**
   * Atributos: ID, Nombre, Descripción, Categoría, Ubicación, Imagen, Horarios, Contacto
   * Métodos: obtenerDetalles(), actualizarDatos(), obtenerReseñas()
3. **Reseña**
   * Atributos: ID, UsuarioID, LugarID, Comentario, Calificación, Fecha
   * Métodos: agregarReseña(), eliminarReseña()
4. **Evento**
   * Atributos: ID, Nombre, Descripción, Fecha, LugarID, Organizador
   * Métodos: crearEvento(), modificarEvento(), eliminarEvento()

Diagrama

El contenido generado por IA puede ser incorrecto.