

Documentación Técnica

“Rally Fotográfico”

ÍNDICE

ÍNDICE	1
Descripción General	1
Características Principales	1
Estructura del Proyecto	2
Estructura de Archivos	2
Dependencias Principales	2
Instrucciones de Instalación	2
Requisitos Previos	2
Configuración del Proyecto	3
Clona el Repositorio	3
Configura Supabase	3
Actualiza las Credenciales de Supabase	5
Instala las Dependencias	5
Coloca la Imagen de Fondo	5
Instrucciones de Uso	6
Para Participantes	6
Registro e Inicio de Sesión	6
Subir una Foto	6
Votar por Fotos	6
Para Administradores	6
Revisar Fotos Pendientes	6
Gestionar Usuarios	6
Eliminar Fotos Aprobadas	6
Resolución de Problemas Comunes	7
Error: Votos Residuales al Eliminar un usuario	7
Síntoma	7
Causa	7
Solución	7
Error: Permiso Denegado al Subir Fotos	7
Síntoma	7
Causa	7
Solución	7

Descripción General

"Rally Fotográfico" es una aplicación móvil desarrollada en Flutter con Supabase como backend, diseñada para permitir a los usuarios participar en un rally fotográfico donde

pueden subir fotos, votar por las fotos de otros, y gestionar usuarios y contenido. La aplicación tiene dos roles principales:

- **Participantes:** Pueden subir fotos, votar por fotos aprobadas, y ver las fotos más votadas.
- **Administradores:** Pueden aprobar o rechazar fotos pendientes, gestionar usuarios (cambiar roles o eliminarlos), y eliminar fotos aprobadas.

Características Principales

- **Autenticación:** Registro e inicio de sesión mediante correo y contraseña, con validación de email y restablecimiento de contraseña.
- **Gestión de Fotos:** Los participantes suben fotos que los administradores deben aprobar o rechazar antes de que sean visibles para todos.
- **Sistema de Votación:** Los usuarios pueden votar por fotos aprobadas (un voto por usuario por foto).
- **Gestión de Usuarios:** Los administradores pueden asignar roles (admin/participante) y eliminar usuarios.
- **Diseño Visual:** Tema con colores lila y morado, fuente Pacifico para títulos, y fondo personalizado.

Estructura del Proyecto

Estructura de Archivos

```
rally_fotografico/  
├── lib/  
│   ├── background_widget.dart    # Widget para el fondo de todas las vistas  
│   ├── main.dart                 # Punto de entrada de la aplicación  
│   └── screens/  
│       ├── home_screen.dart      # Pantalla principal  
│       ├── login_screen.dart     # Pantalla de inicio de sesión/registro  
│       ├── pending_photos_screen.dart # Pantalla para revisar fotos pendientes  
│       ├── reset_password_screen.dart # Pantalla para restablecer contraseña  
│       └── user_management_screen.dart # Pantalla para gestionar usuarios  
├── assets/  
│   └── image.png                 # Imagen de fondo (actualmente no visible)  
└── pubspec.yaml                 # Configuración de dependencias y assets
```

Dependencias Principales

- **Flutter:** Framework para el desarrollo de la UI-
- **Supabase Flutter:** Para autenticación, base de datos, y almacenamiento.
- **Google Fonts:** Para la fuente Pacifico.
- **Image Picker:** Para seleccionar imágenes desde la galería.
- **UUID:** Para generar nombres únicos para las imágenes subidas.
- **Cupertino Icons:** Librería con iconos.

```
dependencies:
  flutter:
    sdk: flutter

  cupertino_icons: ^1.0.8
  supabase_flutter: ^2.9.0
  image_picker: ^1.1.2
  uuid: ^4.5.1
  google_fonts: ^6.2.1
```

Instrucciones de Instalación

Requisitos Previos

- Flutter SDK (versión 3.0 o superior).
- Dart (incluido con Flutter).
- Un editor de código (recomendado: visual Studio Code con las extensiones de flutter y Dart).
- Cuenta en Supabase con un proyecto configurado.

Configuración del Proyecto

Clona el Repositorio

- Si tienes un repositorio, clónalo. Si no, crea un nuevo proyecto Flutter y reemplaza los archivos con los proporcionados.

Configura Supabase

- Crea un proyecto en Supabase.
- Obtén la URL y la clave anónima (anonKey) desde el panel de [Supabase](#) ([Settings](#) > [API](#)).
- Configura las tablas y políticas de RLS en Supabase:
 - Tablas: [users](#), [photos](#), [votes](#).
 - Políticas de RLS (ejecuta las siguientes queries en el editor SQL de Supabase):

–Tabla users–

```
CREATE TABLE users (
  id uuid PRIMARY KEY,
  email text NOT NULL UNIQUE,
  role text NOT NULL DEFAULT 'participant'
);
```

– Tabla photos–

```
CREATE TABLE photos (
  id uuid PRIMARY KEY DEFAULT uuid_generate_v4(),
  user_id uuid REFERENCES users(id),
  url text NOT NULL,
```

```
status text NOT NULL DEFAULT 'pending'
);
```

–Tabla votes–

```
CREATE TABLE votes (
  id uuid PRIMARY KEY DEFAULT uuid_generate_v4(),
  user_id uuid REFERENCES users(id),
  photo_id uuid REFERENCES photos(id),
  UNIQUE(user_id, photo_id)
);
```

–Crear función get_user_role–

```
CREATE OR REPLACE FUNCTION get_user_role(user_id uuid)
RETURNS text AS $$
DECLARE
  user_role text;
BEGIN
  SELECT role INTO user_role
  FROM users
  WHERE id = user_id;
  RETURN COALESCE(user_role, 'participant');
END;
$$ LANGUAGE plpgsql;
```

–Políticas de RLS para users–

```
ALTER TABLE users ENABLE ROW LEVEL SECURITY;
CREATE POLICY "Allow authenticated users to insert their own user record" ON users
  FOR INSERT
  WITH CHECK (auth.uid() = id AND auth.role() = 'authenticated');
CREATE POLICY "Allow admins to delete users" ON users
  FOR DELETE
  USING (get_user_role(auth.uid()) = 'admin');
CREATE POLICY "Allow admins to update users" ON users
  FOR UPDATE
  USING (get_user_role(auth.uid()) = 'admin');
CREATE POLICY "Allow users to select their own role" ON users
  FOR SELECT
  USING (auth.uid() = id OR get_user_role(auth.uid()) = 'admin');
CREATE POLICY "Allow select for email check" ON users
  FOR SELECT
  USING (true);
```

–Políticas de RLS para photos–

```
ALTER TABLE photos ENABLE ROW LEVEL SECURITY;
CREATE POLICY "Allow authenticated users to insert photos" ON photos
  FOR INSERT
  WITH CHECK (auth.uid() = user_id);
CREATE POLICY "Allow admins to update photos" ON photos
```

```

FOR UPDATE
USING (get_user_role(auth.uid()) = 'admin');
CREATE POLICY "Allow admins to delete photos" ON photos
FOR DELETE
USING (get_user_role(auth.uid()) = 'admin');
CREATE POLICY "Allow select for approved photos" ON photos
FOR SELECT
USING (status = 'approved' OR get_user_role(auth.uid()) = 'admin');

```

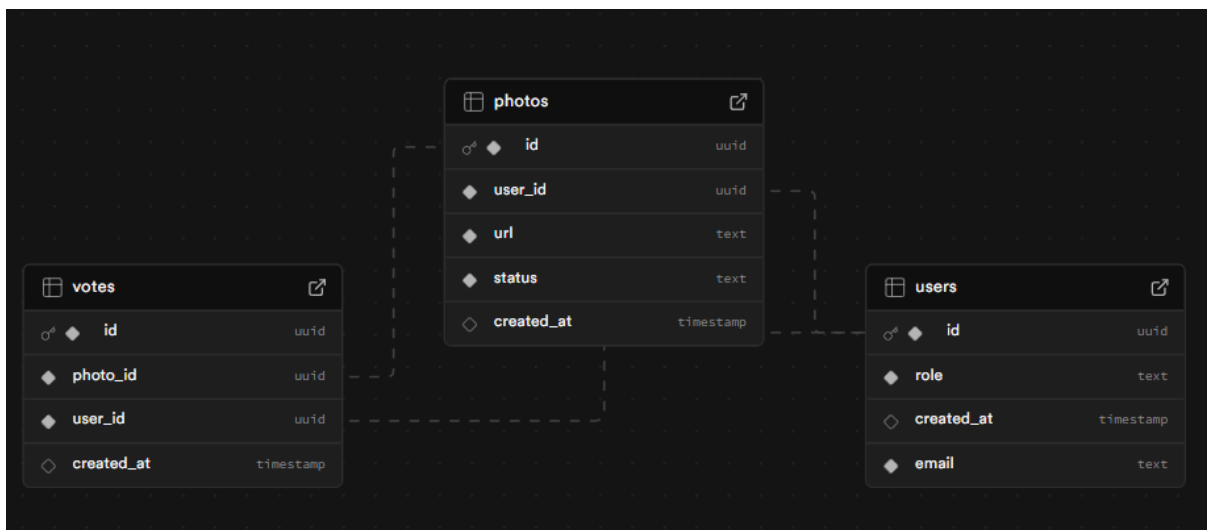
–Políticas de RLS para votes–

```

ALTER TABLE votes ENABLE ROW LEVEL SECURITY;
CREATE POLICY "Allow authenticated users to insert votes" ON votes
FOR INSERT
WITH CHECK (auth.uid() = user_id);
CREATE POLICY "Allow authenticated users to delete their own votes" ON votes
FOR DELETE
USING (auth.uid() = user_id);
CREATE POLICY "Allow admins to delete votes" ON votes
FOR DELETE
USING (get_user_role(auth.uid()) = 'admin');
CREATE POLICY "Allow select for votes" ON votes
FOR SELECT
USING (true);

```

- Configura un bucket de almacenamiento (**rallyphotos**) para las imágenes subidas:
 - Ve a **Storage > Buckets** y crea un bucket llamado **rallyphotos**.
 - Asegúrate de que el bucket sea público o configure políticas de acceso adecuadas.



Actualiza las Credenciales de Supabase

- Abre `lib/main.dart` y actualiza la URL

```
await Supabase.initialize(  
  url: 'https://hnunevlnlgeochunbuba.supabase.co',  
  anonKey: 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3M',  
  authOptions: const FlutterAuthClientOptions(  
    authFlowType: AuthFlowType.pkce,  
    autoRefreshToken: true,  
  ),  
);
```

Instala las Dependencias

- Asegúrate de que `pubspec.yaml` incluya las dependencias necesarias (ver sección “Estructura del Proyecto”).

```
flutter pub get
```

Coloca la Imagen de Fondo

- Coloca la imagen `image.png` en la carpeta `assets/`.
- Asegúrate de que `pubspec.yaml` declare el asset:

```
flutter:
  uses-material-design: true
  assets:
    - assets\image.png
```

- Ejecuta `flutter pub get` nuevamente.

```
flutter run
```

Instrucciones de Uso

Para Participantes

Registro e Inicio de Sesión

- Abre la aplicación y haz click en “[Iniciar Sesión / Registrarse](#)”.
- Regístrate con un correo y contraseña (mínimo 8 caracteres, con mayúscula, minúscula, número y carácter especial).
- Confirma tu correo (revisa tu bandeja de entrada o los logs de Supabase).
- Inicia sesión con tus credenciales.

Subir una Foto

- Desde la pantalla principal ([HomeScreen](#)), haz click en “Subir Foto”.
- Selecciona una imagen desde tu galería.
- La foto se subirá y quedará pendiente de aprobación por un administrador.

Votar por Fotos

- En **HomeScreen**, verás las fotos aprobadas ordenadas por número de votos (descendente),
- Haz click en el ícono de corazón para votar por una foto (solo puedes votar una vez por foto).
- Para retirar tu voto, haz click nuevamente en el icono de corazón.

Para Administradores

Revisar Fotos Pendientes

- Desde **HomeScreen**, haz click en “Revisar Fotos Pendientes”.
- En **PendingPhotosScreen**, verás las fotos pendientes de aprobación.
- Haz click en “Aprobar” o “Rechazar” para cada foto.
- Las fotos aprobadas serán visibles para todos los usuarios en **HomeScreen**.

Gestionar Usuarios

- Desde **HomeScreen**, haz click en “Gestionar Usuarios”.
- En **UserManagementScreen**, verás una lista de usuarios.
- Usa el interruptor para cambiar el rol de un usuario entre “admin” y “participante”.
- Haz click en el icono de papelera para eliminar un usuario (esto elimina sus votos, fotos y registro en **users**).

Eliminar Fotos Aprobadas

- En **HomeScreen**, las fotos aprobadas tienen un icono de papelera (visible solo para administradores).
- Haz click en el icono de papelera para eliminar una foto (esto también elimina los votos asociados a la foto).

Resolución de Problemas Comunes

Error: Votos Residuales al Eliminar un usuario

Síntoma

- Al intentar eliminar un usuario desde **UserManagmentScreen**, aparece un error como:

```
No se pudieron eliminar todos los votos asociados al usuario. Votos residuales: 1
```

Causa

- Hay votos en la tabla votes que no se eliminaron correctamente debido a permisos de RLS o errores silenciosos.

Solución

1. Revisa los logs para identificar el ID del voto residual.

2. Elimina el voto manualmente:
 - a. Ve a Database > Tables > **votes** en Supabase.
 - b. Busca el voto con el ID **[voteld]** y elimínalo.
3. Vuelve a intentar eliminar el usuario.
4. Si el problema persiste, verifica las políticas de RLS en **votes**.
SELECT * FROM pg_policies WHERE tablename = 'votes';
 - a. Asegúrate de que no haya políticas conflictivas y que los administradores puedan eliminar todos los votos.

Error: Permiso Denegado al Subir Fotos

Síntoma

- Al intentar subir una foto, aparece un error como:

```
Error al subir la foto: permission denied for table photos
```

Causa

- Las políticas de RLS en la tabla **photos** no permiten la operación.

Solución

- Verifica las políticas de RLS para **photos**:
SELECT * FROM pg_policies WHERE tablename = 'photo';
- Asegúrate de que la política **Allow authenticated users to insert photos** esté activa (ver sección “Configuración del Proyecto”).

Notas Adicionales

Limitaciones Conocidas

La eliminación de usuarios de **auth.users** debe hacerse manualmente desde el panel de Supabase después de eliminarlos de la tabla users.

Mejoras Futuras

- Crear una nueva feature dónde el participante pueda añadir sus datos y añadir una foto de perfil.
- Automatizar la eliminación de usuario de **auth.users** mediante la API de Supabase.

Contacto y Soporte

Para soporte adicional, contacte al equipo de desarrollo o revise los logs de depuración para más detalle. Si necesitas ayuda con un error específico, incluye el mensaje de error completo y los logs relevantes.