PROYECTO GOALFUT

APLICATIVO MÓVIL PARA LA GESTIÓN DE TORNEOS DE FÚTBOL DE SALÓN DE BARRIO

Estudiante: Marlon Jeshit Lopes Armero

Asignatura: Sistemas Móviles

Ciclo: Dos

Carrera: Ing Sistemas

Semestre: Sexto

Institución: CEIE

TABLA DE CONTENIDO

11. ESTRUCTURA DEL CÓDIGO - GOALFUT

1. ÁRBOL DE PROBLEMAS

1.1 Problema Central

Dificultad en la gestión y difusión de información estadística en los torneos de fútbol de salón de barrio, debido al uso de métodos manuales y herramientas no especializadas que generan retrasos y errores en la comunicación con los usuarios.

1.2 Causas del Problema

Causas Directas:

- **Dependencia de registros físicos:** Uso de planillas de papel y cuadernos para el control de goles, tarjetas y tiempos.

- **Procesamiento manual de datos:** Necesidad de transcribir información a hojas de cálculo (Excel) para generar tablas de posiciones.
- **Canales de información fragmentados:** Difusión de resultados mediante fotos o mensajes de texto que se pierden en el historial.

Causas Indirectas:

- Falta de una plataforma móvil centralizada que unifique las funciones de organizador y espectador.
- Dificultad para registrar eventos del partido en tiempo real desde el lugar del encuentro.
- Limitaciones tecnológicas de los organizadores para implementar sistemas de gestión digital complejos.

1.3 Efectos del Problema

Efectos Directos:

- **Inexactitud estadística:** Frecuentes errores en el conteo de puntos, diferencia de gol y tabla de goleadores.
- **Saturación informativa:** El organizador debe responder manualmente múltiples consultas sobre horarios y resultados.
- **Desinformación del usuario:** Jugadores y aficionados no conocen su posición o próximos encuentros de forma inmediata.

Efectos a Largo Plazo:

- **Baja calidad organizativa:** Percepción de informalidad y falta de transparencia en la competencia.
- **Pérdida de historial deportivo:** Inexistencia de un registro digital acumulativo de los logros de equipos y jugadores.

2. PROBLEMA CENTRAL

El Problema Central que justifica el desarrollo de GoalFut es la **ineficiencia operativa en la administración de competencias deportivas de barrio**.

Esta ineficiencia surge por la desconexión entre la captura de datos en el campo y su publicación final. Al no contar con un flujo digital automatizado, el organizador invierte tiempo excesivo en tareas repetitivas, la información llega con retraso a los interesados y el margen de error en las estadísticas es alto, lo que impide que los torneos de fútbol de salón alcancen un nivel de organización profesional y accesible para la comunidad.

3. JUSTIFICACIÓN DEL PROYECTO

3.1 Problema a Resolver

La ineficiencia operativa en la administración de competencias deportivas de barrio surge por la desconexión entre la captura de datos en campo y su publicación final.

3.2 Solución Propuesta

GoalFut proporciona una plataforma móvil que permite:

- Registro en tiempo real de eventos durante partidos
- Cálculo automático de estadísticas (posiciones, goleadores, vallas)
- Difusión instantánea de resultados a usuarios interesados
- Funcionamiento offline para zonas sin conectividad

3.3 Beneficios

- **Administradores:** Reducción de tiempo en gestión, eliminación de errores manuales
- **Jugadores:** Acceso inmediato a estadísticas personales y del equipo
- **Aficionados:** Seguimiento en tiempo real de torneos favoritos
- **Comunidad:** Historial digital permanente

4. DIAGRAMAS UML

4.1 Diagrama de Casos de Uso

Descripción:
Este diagrama representa las interacciones entre los diferentes tipos de usuarios (actores) y las funcionalidades del sistema. Define QUÉ puede hacer cada tipo de usuario en la aplicación GoalFut.

Actores del Sistema:

- **Invitado:** Usuario sin registro. Solo puede visualizar información pública de torneos, equipos y partidos.
- **Usuario Registrado:** Hereda las funciones del Invitado. Puede seguir torneos y recibir notificaciones.
- **Administrador:** Hereda las funciones del Usuario Registrado. Puede crear y gestionar torneos, equipos, jugadores y partidos.

Casos de Uso Principales:

Paquete	Casos de Uso
Públicas	Ver torneos, Ver equipos, Ver partidos, Ver tabla de posiciones, Ver goleadores
Usuario	Registrarse, Iniciar sesión, Seguir torneo, Configurar notificaciones

Administración	Crear torneo, Gestionar equipos, Gestionar jugadores, Programar partidos, Registrar eventos en vivo

Relaciones:

- **Herencia:** Admin → Usuario Registrado → Invitado
- **Include:** "Gestionar partido en vivo" incluye "Registrar gol" y "Registrar tarjeta"
- **Extend:** "Seguir torneo" extiende "Ver torneo"

Ver Anexo: Imagen 4.1 - Diagrama de Casos de Uso

Diagrama de Casos de Uso - App GoalFut.png

4.2 Diagrama de Clases

Descripción:
Este diagrama muestra la estructura de datos del sistema y las relaciones entre las entidades. Define CÓMO se organiza la información en la base de datos y los servicios de la aplicación.

Entidades Principales:

Clase	Atributos Principales	Responsabilidad
Usuario	id, nombre, email, tipo_usuario, push_token	Gestionar autenticación y permisos
Torneo	id, nombre, estado, fecha_inicio, admin_id	Representar competencia deportiva
Equipo	id, nombre, torneo_id, puntos, goles_favor	Representar participante del torneo
Jugador	id, nombre, equipo_id, numero, posicion	Representar miembro del equipo
Partido	id, equipo_local_id, equipo_visitante_id, estado	Representar encuentro deportivo
EventoPartido	id, partido_id, tipo, minuto, jugador_id	Registrar goles, tarjetas, cambios
TorneoSeguido	id, usuario_id, torneo_id	Relacionar usuario con torneo

Servicios:

Servicio	Función
NotificationService	Enviar notificaciones push de goles
SyncService	Sincronizar datos offline con servidor
OfflineMatchService	Gestionar partidos sin conexión

Relaciones:

- Un Torneo tiene muchos Equipos (1:N)
- Un Equipo tiene muchos Jugadores (1:N)
- Un Partido tiene muchos EventosPartido (1:N)
- Un Usuario sigue muchos Torneos (N:M mediante TorneoSeguido)

Ver Anexo: Imagen 4.2 - Diagrama de Clases

Diagrama de Clases - App GoalFut.png

4.3 Diagramas de Actividades

4.3.1 Diagrama de Actividad: Crear Torneo

Descripción:
Este diagrama muestra el flujo de trabajo completo para crear y configurar un torneo desde cero hasta su activación.

Flujo Principal:

1. Administrador ingresa datos básicos del torneo (nombre, fechas, lugar)
2. Sistema valida información ingresada
3. Si es válido → Administrador agrega equipos participantes
4. Para cada equipo → Administrador agrega jugadores
5. Sistema genera calendario de partidos automáticamente
6. Administrador revisa y activa el torneo
7. Torneo queda disponible para usuarios

Decisiones:

¿Datos válidos? → Sí: continuar / No: mostrar errores
¿Más equipos? → Sí: agregar otro / No: continuar con jugadores
¿Calendario correcto? → Sí: activar / No: editar partidos

Ver Anexo: Imagen 4.3.1 - Diagrama de Actividad Crear Torneo Y Configurar Torneo

Diagrama de Actividad - Crear y Configurar Torneo - App GoalFut.png

4.3.2 Diagrama de Actividad: Partido en Vivo

Descripción:
Este diagrama muestra el flujo para gestionar un partido en tiempo real con cronómetro y registro de eventos.

Flujo Principal:

1. Administrador selecciona partido e inicia encuentro
2. Sistema activa cronómetro automático
3. Durante el partido:
    - Registrar gol → Seleccionar equipo y jugador → Actualizar marcador
    - Registrar tarjeta → Seleccionar tipo y jugador → Guardar evento
4. Al cumplir tiempo → Finalizar primer tiempo
5. Iniciar segundo tiempo → Repetir registro de eventos
6. Al cumplir tiempo total → Finalizar partido
7. Sistema actualiza estadísticas y tabla de posiciones automáticamente

Eventos Paralelos:

Cronómetro corre de forma continua
Notificaciones se envían a seguidores en cada gol

Ver Anexo: Imagen 4.3.2 - Diagrama de Actividad Gestionar Partido en Vivo

Diagrama de Actividad - Gestionar Partido en Vivo - App GoalFut.png

4.3.3 Diagrama de Actividad: Seguir Torneo

Descripción:
Este diagrama muestra el flujo de un usuario para seguir un torneo y recibir notificaciones.

Flujo Principal:

1. Usuario navega a detalle de torneo
2. Sistema verifica: ¿Usuario autenticado?
    - **No:** Mostrar mensaje "Debe iniciar sesión para seguir torneos"
    - **Sí:** Continuar
3. Usuario toca botón de seguir
4. Sistema registra seguimiento y push token
5. Usuario recibirá notificaciones de goles de ese torneo

Flujo Alternativo (Dejar de seguir):

1. Usuario toca botón de dejar de seguir

2. Sistema elimina registro de seguimiento
3. Usuario no recibirá más notificaciones

Ver Anexo: Imagen 4.3.3 - Diagrama de Actividad Seguir Torneo

Diagrama de Actividad - Seguir Torneo - App GoalFut.png

4.3.4 Diagrama de Actividad: Sincronización Offline

Descripción:
Este diagrama muestra el flujo de datos cuando el administrador trabaja sin conexión a internet.

Flujo Principal:

1. Sistema detecta: ¿Hay conexión a internet?
   - **Sí:** Operación normal con Supabase
   - **No:** Activar modo offline
2. En modo offline:
   - Eventos se guardan en cola local (AsyncStorage)
   - UI muestra indicador "Pendiente de sincronizar"
3. Sistema detecta: ¿Conexión restaurada?
   - **No:** Continuar guardando en cola
   - **Sí:** Iniciar sincronización
4. Proceso de sincronización:
   - Obtener operaciones pendientes
   - Para cada operación → Enviar a Supabase
   - Si éxito → Marcar como sincronizado
   - Si error → Reintentar (máximo 3 veces)
5. Cola vacía → Fin del proceso

Ver Anexo: Imagen 4.3.4 - Diagrama de Actividad Sincronización Offline

Diagrama de Actividad - Sincronización Offline - App GoalFut.png

5. TECNOLOGÍAS UTILIZADAS

5.1 Stack de Desarrollo

Tecnología	Versión	Propósito
React Native	0.76+	Framework principal para desarrollo móvil multiplataforma
Expo	SDK 52	Plataforma de desarrollo, compilación y distribución
JavaScript/ES6+	-	Lenguaje de programación principal

Supabase	-	Backend as a Service (Base de datos, Autenticación, Storage)
PostgreSQL	-	Base de datos relacional (provista por Supabase)

5.2 Herramientas de Desarrollo

Herramienta	Propósito
Visual Studio Code	Editor de código principal
EAS Build	Servicio de compilación en la nube para generar APK
Git	Control de versiones
npm	Gestor de paquetes de Node.js

5.3 Servicios en la Nube

Servicio	Uso en GoalFut
Supabase Database	Almacenamiento de torneos, equipos, jugadores, partidos
Supabase Auth	Autenticación de usuarios con email/contraseña
Supabase Realtime	Actualizaciones en tiempo real de partidos
Expo EAS	Compilación y distribución de la aplicación

6. DEPENDENCIAS DEL PROYECTO

6.1 Dependencias Principales

Dependencia	Versión	Descripción y Uso
react	18.3.1	Librería base para construir interfaces de usuario con componentes
react-native	0.76.6	Framework para crear aplicaciones móviles nativas usando React
expo	~52.0.27	SDK que simplifica el desarrollo con React Native, proporciona APIs nativas
@supabase/supabase js	^2.47.10	Cliente oficial de Supabase para conectar con la base de datos y autenticación

6.2 Navegación

Dependencia	Descripción y Uso

@react-navigation/native	Librería principal de navegación para React Native
@react-navigation/stack	Navegación tipo pila (push/pop) entre pantallas
@react-navigation/bottom-tabs	Barra de navegación inferior con pestañas
react-native-screens	Optimiza el rendimiento de navegación usando vistas nativas
react-native-safe-area-context	Maneja áreas seguras en dispositivos con notch

6.3 Almacenamiento y Offline

Dependencia	Descripción y Uso
@react-native-async-storage/asyncstorage	Almacenamiento local persistente para datos offline y caché

6.4 Interfaz de Usuario

Dependencia	Descripción y Uso
@expo/vector-icons	Conjunto de íconos vectoriales (Ionicons, MaterialIcons, etc.)
expo-font	Carga de fuentes personalizadas en la aplicación
react-native-gesture-handler	Manejo de gestos táctiles (swipe, pinch, etc.)
@react-native-community/datetimepicker	Selector nativo de fecha y hora para programar partidos

6.5 Funcionalidades del Dispositivo

Dependencia	Descripción y Uso
expo-notifications	Envío y recepción de notificaciones push
expo-image-picker	Acceso a cámara y galería para subir fotos de jugadores/equipos
@react-native-community/netinfo	Detecta estado de conexión a internet para modo offline

6.6 Utilidades

Dependencia	Descripción y Uso
uuid	Genera identificadores únicos universales para registros
expo-status-bar	Control de la barra de estado del dispositivo

7. HARDWARE Y SENSORES DEL DISPOSITIVO

7.1 Hardware Utilizado

Componente	Uso en GoalFut	Obligatorio
Pantalla Táctil	Interfaz de usuario, cronómetro, registro de eventos	Sí
Almacenamiento	Caché de datos para modo offline (AsyncStorage)	Sí
Conexión de Red	WiFi/Datos móviles para sincronización con Supabase	Sí*
Cámara	Subir fotos de jugadores y logos de equipos	No
Vibración	Feedback en notificaciones push de goles	No

*Requerido para sincronización inicial, pero la app puede funcionar offline para algunas operaciones.

7.2 APIs y Permisos del Sistema

```
{
    “permisos_android”: [
        “INTERNET”,
        “ACCESS_NETWORK_STATE”,
        “CAMERA”,
        “READ_EXTERNAL_STORANGE”,
        “WRITE_EXTERNAL_STORAGE”,
        “RECEIVE_BOOT_COMPLETED”,
        “VIBRATE”
    ]
}
```

7.3 Librerías de Hardware

Librería	Función
expo-image-picker	Acceso a cámara y galería de fotos
expo-notifications	Notificaciones push locales y remotas

@react-native-async-storage/async-storage	Almacenamiento local persistente
@react-native-community/netinfo	Detección de estado de red

8. LIMITACIONES DE LA APP

8.1 Funcionalidades NO Incluidas

Transmisión de Video:

- La app no transmite partidos en vivo
- No hay integración con plataformas de streaming
- No graba videos de los partidos

Comunicación entre Usuarios:

- No hay chat entre usuarios
- No hay foro de discusión
- No hay comentarios en partidos

Funcionalidades de Pago:

- No procesa pagos de inscripciones
- No gestiona cobros a jugadores
- No hay integración con pasarelas de pago

Geolocalización:

- No muestra ubicación GPS de las canchas
- No hay mapas integrados
- Solo muestra dirección en texto

Análisis Avanzado:

- No genera gráficos estadísticos complejos
- No hay mapas de calor de jugadores
- No calcula posesión de balón

Redes Sociales:

- No publica automáticamente en redes sociales
- No permite login con redes sociales

8.2 Límites Operativos

Límite	Valor
Torneos activos por administrador	3 máximo

Tamaño máximo de imagen	5 MB
Reintentos de sincronización offline	3 intentos
Idioma disponible	Solo Español

9. LIMITACIONES TÉCNICAS DEL BUILD

9.1 Problema con la Generación del APK

Durante el desarrollo del proyecto, se presentaron dificultades técnicas para generar un nuevo build (APK) que incluyera todas las configuraciones implementadas para las funcionalidades de **notificaciones push** y **modo offline completo**.

Causa del problema:

El servicio de compilación EAS Build (Expo Application Services) presentó errores recurrentes relacionados con:

1. **Ruta del proyecto:** El proyecto estaba ubicado en una ruta con caracteres especiales y espacios, lo cual causaba fallos en el proceso de compresión y subida de archivos al servicio de compilación.
2. **Dependencia expo-sqlite:** Se intentó implementar SQLite como solución de almacenamiento offline más robusta, pero esta dependencia requiere compilación nativa que no era compatible con el build existente.
3. **Errores de tar/compresión:** El proceso de empaquetado del proyecto fallaba con el error "Cannot open: No such file or directory" en múltiples intentos de build.

Intentos de solución:

Intento	Acción	Resultado
1	Build desde ruta original	Error de compresión
2	Copiar proyecto a nueva ruta	Error por dependencia expo-sqlite
3	Remover expo-sqlite	Error persistente en EAS Build
4	Múltiples builds en EAS	Todos fallaron

Estado actual:

El código fuente incluye toda la lógica para notificaciones push y modo offline
Las funcionalidades están implementadas y funcionan en modo desarrollo

No se pudo generar un APK de producción con estas características
El build de producción disponible corresponde a una versión anterior sin estas funcionalidades completas

9.2 Funcionalidades Afectadas

Funcionalidad	Estado en Código	Estado en APK Producción
Notificaciones push de goles	Implementada	No probada en producción
Modo offline para admin	Implementada	Funciona con caché básico
Sincronización de cola	Implementada	No probada completamente

9.3 Recomendaciones Futuras

Para resolver estos problemas en futuras versiones:

1. Utilizar una ruta de proyecto sin caracteres especiales ni espacios
2. Configurar un ambiente de desarrollo en una máquina con menos restricciones
3. Considerar el uso de GitHub Actions para compilación automatizada
4. Realizar builds incrementales para detectar problemas temprano

10. REPOSITORIO DEL CÓDIGO

10.1 Acceso al Código Fuente

El código fuente completo del proyecto GoalFut está disponible en el siguiente repositorio:

Enlace del repositorio: https://github.com/JESHIT297/GoalFut

10.2 Estructura del Repositorio

```
GoalFut/
├── App.js                # Punto de entrada de la aplicación
├── app.json              # Configuración de Expo
├── package.json          # Dependencias del
proyecto ├── src/
│   ├── components/             # Componentes reutilizables
│   ├── contexts/         # Estados globales (Auth, Offline)
│   ├── screens/          # Pantallas de la aplicación
│   ├── services/         # Lógica de negocio
│   ├── database/         # Almacenamiento local
│   ├── navigation/       # Configuración de navegación
│   ├── config/           # Configuración de Supabase
│   └── utils/            # Constantes y funciones helper
```

```
├── supabase/
│   └── migrations/      # Scripts SQL de la base de datos
└── docs/                # Documentación del proyecto
```

10.3 Instrucciones de Instalación

1. Clonar el repositorio
    - git clone [URL_DEL_REPOSITORIO]
2. Instalar dependencias
    - cd GoalFut
    - npm install
3. Iniciar servidor de desarrollo
    - npx expo start
4. Escanear código QR con Expo Go (Android/iOS)

11. ESTRUCTURA DEL CÓDIGO - GOALFUT

Descripción General

GoalFut es una aplicación móvil desarrollada con React Native y Expo SDK 54, diseñada para la gestión de torneos de fútbol. Utiliza Supabase como backend para autenticación, base de datos y almacenamiento de imágenes.

Organización de Carpetas

Carpeta Principal (src/):

Carpeta	Descripción
screens/	Contiene todas las pantallas de la aplicación, organizadas en subcarpetas: auth/ (login y registro), admin/ (panel de administración) y public/ (vistas para usuarios).
services/	Lógica de negocio y comunicación con Supabase. Incluye servicios para torneos, equipos, jugadores, partidos, notificaciones e imágenes.
components/	Componentes de interfaz reutilizables como botones, tarjetas, campos de entrada y estados de carga.

contexts/	Manejo del estado global usando Context API. Incluye AuthContext para autenticación y OfflineContext para modo sin conexión.
navigation/	Configuración de rutas y navegación entre pantallas usando React Navigation.
config/	Configuración del cliente de Supabase.
database/	Esquema y operaciones de base de datos local (SQLite) para funcionalidad offline.

Tecnologías Utilizadas

Framework: React Native con Expo

Navegación: React Navigation v7

Backend: Supabase (autenticación, base de datos, almacenamiento)

Estado Global: Context API de React

Notificaciones: Expo Notifications

Almacenamiento Local: AsyncStorage y SQLite

Flujo de la Aplicación

La aplicación inicia en App.js, que configura los proveedores de contexto (AuthProvider y OfflineProvider) y carga el navegador principal (AppNavigator). Dependiend del estado de autenticación, el usuario es dirigido a las pantallas de login/registro o al contenido principal de la aplicación.

ANEXOS

Anexo de Imágenes

A continuación se listan las imágenes referenciadas en este documento:

Anexo	Descripción
Imagen 4.1	Diagrama de Casos de Uso - App GoalFut
Imagen 4.2	Diagrama de Clases - App GoalFut