

1. Introducción

1.1 Propósito

El propósito de este documento es especificar los requisitos funcionales y no funcionales para la creación de una aplicación móvil con Flutter que permita a los estudiantes universitarios y al personal administrativo gestionar eventos dentro de la unidad académica. La aplicación facilitará la búsqueda, el registro, la promoción y la gestión de eventos de manera eficiente.

1.2 Alcance del proyecto

El proyecto comprende el desarrollo de una aplicación móvil para Android y un backend para la administración y persistencia de los datos. Se centrará en los siguientes módulos principales:

- Autenticación de usuarios.
- Exploración y búsqueda de eventos.
- Registro y gestión de asistencia a eventos.
- Creación y administración de eventos por parte de organizadores.
- Notificaciones para eventos.

1.3 Público objetivo

- **Estudiantes:** Buscan y se registran en eventos académicos, culturales y deportivos.
- **Organizadores:** Crean, editan y gestionan sus propios eventos.
- **Personal administrativo:** Supervisa y valida eventos a nivel institucional.

2. Requerimientos funcionales

2.1 Módulo de autenticación de usuarios

- **RF-001:** El sistema debe permitir a los usuarios registrarse con correo electrónico institucional.
- **RF-002:** El sistema debe permitir el inicio de sesión con credenciales.
- **RF-003:** El sistema debe ofrecer la opción de recuperar la contraseña mediante un enlace enviado al correo electrónico.
- **RF-004:** El sistema debe soportar tres tipos de roles: estudiante, organizador y administrador.
- **RF-005:** El sistema debe permitir la edición del perfil del usuario (nombre, foto, intereses).

2.2 Módulo de eventos

- **RF-006:** El sistema debe mostrar una lista de todos los eventos disponibles.
- **RF-007:** El sistema debe permitir filtrar eventos por categorías, fecha y ubicación.
- **RF-008:** El sistema debe mostrar los detalles de cada evento, incluyendo descripción, fecha, hora, lugar y organizador.
- **RF-009:** El sistema debe permitir a los usuarios registrarse para asistir a un evento.
- **RF-010:** El sistema debe enviar una notificación de confirmación al usuario registrado.

2.3 Módulo de gestión para organizadores

- **RF-011:** El sistema debe permitir a los usuarios con rol organizador crear nuevos eventos.
- **RF-012:** El sistema debe permitir a los organizadores editar o cancelar los eventos que han creado.
- **RF-013:** El sistema debe permitir a los organizadores ver una lista de los asistentes registrados a sus eventos.
- **RF-014:** El sistema debe permitir a los organizadores notificar a los asistentes sobre cambios en el evento.

2.4 Módulo de notificaciones *

- **RF-015:** El sistema debe enviar notificaciones push a los usuarios para recordarles eventos próximos en los que se han registrado.
- **RF-016:** El sistema debe enviar notificaciones a los organizadores cuando alguien se registra en su evento.
- **RF-017:** Los usuarios deben poder configurar sus preferencias de notificación.

3. Requerimientos no funcionales

3.1 Usabilidad

- **RNF-001:** La aplicación debe tener una interfaz de usuario intuitiva y fácil de usar.
- **RNF-002:** El diseño debe ser adaptable a diferentes tamaños de pantalla (diseño responsivo).
- **RNF-003:** La navegación debe ser clara y consistente en toda la aplicación.

3.2 Rendimiento

- **RNF-004:** La aplicación debe cargar la lista de eventos en menos de 3 segundos, incluso con una gran cantidad de datos.

- **RNF-005:** Las transiciones y animaciones deben ser fluidas y sin demoras.

3.3 Confiabilidad

- **RNF-006:** La aplicación debe manejar correctamente los errores de conexión a internet, mostrando mensajes apropiados al usuario.
- **RNF-007:** Los datos de los eventos y usuarios deben ser persistentes y no perderse en caso de fallo.

3.4 Seguridad

- **RNF-008:** Las contraseñas de los usuarios deben almacenarse de forma segura, utilizando técnicas de hashing.
- **RNF-009:** La comunicación entre la aplicación y el backend debe ser cifrada (HTTPS).
- **RNF-010:** Solo los usuarios autenticados deben tener acceso a las funcionalidades restringidas.

4. Arquitectura y tecnología

4.1 Arquitectura

- **Frontend (Flutter):** Se utilizará Flutter para el desarrollo de la aplicación móvil.
- **Gestión de estado:** Se implementará un patrón de gestión de estado como Provider, BLoC o Riverpod para manejar el flujo de datos de la aplicación.
- **Backend:** Se recomienda utilizar Firebase/Supabase por su facilidad de uso y por ofrecer una base de datos en tiempo real, autenticación y almacenamiento. También puede utilizar base de datos relacionales, con su respectiva API.
- **Notificaciones*:** Se utilizará Firebase Cloud Messaging (FCM) para enviar notificaciones push a los usuarios.

La presentación de proyectos se llevará a cabo del 6 al 8 de enero de 2026.