

Historia Listar Eventos

Funcionalidad General

Permite a los usuarios visualizar una tabla con información relevante sobre los eventos disponibles en la plataforma de juegos florales. La tabla incluirá columnas para el ID del evento, nombre, equipos participantes, fixture ID y actividades asociadas. Los usuarios pueden consultar estos detalles tanto en la versión web como en la aplicación móvil.

Página Web

- React: biblioteca de JavaScript desarrollada por Facebook que se utiliza para construir interfaces de usuario, especialmente para aplicaciones web de una sola página
 - Pantalla de Listado de Eventos: La página web mostrará una tabla con las siguientes columnas:
 - ID: Identificador único del evento.
 - Nombre: Nombre del evento.
 - Equipos: Lista de equipos participantes en el evento.
 - FixtureID: Identificador del fixture asociado al evento.
 - Actividades: Actividades o tareas asociadas con el evento.
 - Interacción con la Tabla: Los usuarios podrán ver todos los eventos disponibles en una vista clara y organizada. Se pueden implementar filtros y opciones de búsqueda para mejorar la usabilidad.

App Mobile

- Flutter: Framework de código abierto desarrollado por Google que se utiliza para crear aplicaciones móviles, web y de escritorio a partir de una única base de código.
 - Pantalla de Listado de Eventos: En la aplicación móvil, la pantalla de eventos mostrará una tabla similar a la de la web:
 - ID: Identificador único del evento.
 - Nombre: Nombre del evento.
 - Equipos: Lista de equipos participantes en el evento.
 - FixtureID: Identificador del fixture asociado al evento.
 - Actividades: Actividades o tareas asociadas con el evento.
 - Interacción con la Tabla: La tabla en la aplicación móvil estará optimizada para pantallas táctiles y ofrecerá una experiencia de usuario fluida, con la posibilidad de desplazar y buscar eventos.

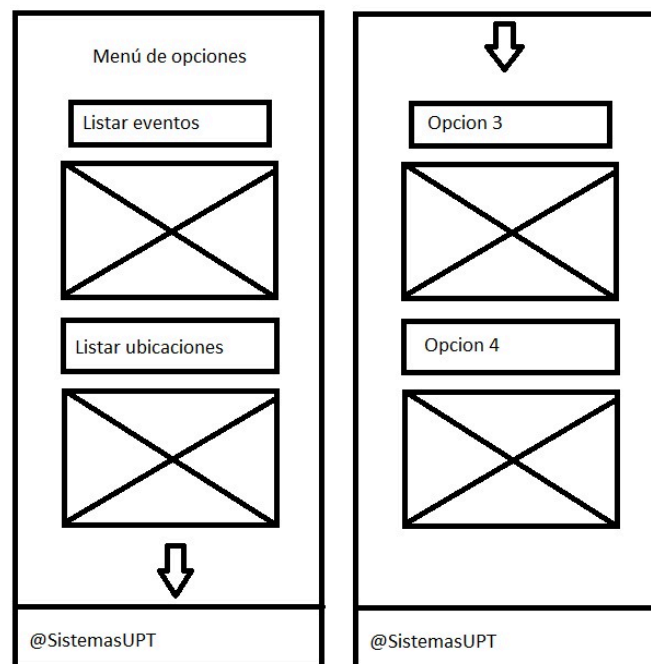
Backend

- C#: Lenguaje de programación desarrollado por Microsoft, conocido por su rendimiento y soporte para aplicaciones empresariales.
- MongoDB.Driver: Biblioteca para interactuar con MongoDB, una base de datos NoSQL, que facilita el manejo de datos flexibles y escalables.
 - Gestión de Datos de Eventos: El backend en C# manejará la lógica para recuperar, procesar y enviar los datos de eventos a las interfaces de usuario. MongoDB.Driver se utilizará para interactuar con la base de datos MongoDB, donde se almacenarán los datos de los eventos.
 - API para Listar Eventos: Se implementará una API en C# que proporcionará los datos de los eventos en formato JSON. La API incluirá endpoints para:
 - Obtener la lista completa de eventos.
 - Obtener detalles específicos de cada evento.

Ventajas

- Accesibilidad: Los usuarios pueden acceder a la información de los eventos desde cualquier dispositivo, ya sea una computadora o un dispositivo móvil.
- Usabilidad: La tabla proporciona una vista clara y estructurada de los eventos, facilitando la búsqueda y la consulta de información relevante.
- Consistencia: La integración entre la versión web y la aplicación móvil asegura que los usuarios tengan una experiencia uniforme y coherente en ambas plataformas.
- Rendimiento y Escalabilidad: El uso de C# y MongoDB.Driver proporciona un backend robusto y escalable, capaz de manejar grandes volúmenes de datos y consultas eficientes.

Ejemplo de Diseño - Listar Eventos



Listar Eventos	
Como	Un usuario de la plataforma de juegos florales
Quiero	Iniciar sesión utilizando mi cuenta de Microsoft
Para que	Pueda acceder a la plataforma de manera segura y rápida sin necesidad de crear una nueva cuenta.
Criterios de Aceptación	
Pantalla de Listado de Eventos:	<ul style="list-style-type: none">• En la página web y en la aplicación móvil, debo ver una tabla con los eventos disponibles.• La tabla debe incluir las siguientes columnas: ID, Nombre, Equipos, FixtureID y Actividades.
Datos en la Tabla:	<ul style="list-style-type: none">• La tabla debe mostrar la información actualizada de los eventos.• Cada fila de la tabla debe representar un evento y

	<p>contener los datos correspondientes en las columnas.</p>
Actualización de Datos:	<ul style="list-style-type: none"> • La tabla debe actualizarse en tiempo real cuando se realicen cambios en los eventos en el backend. • Los datos mostrados deben ser consistentes con la información almacenada en la base de datos.
Seguridad y Rendimiento:	<ul style="list-style-type: none"> • La transmisión de datos entre la interfaz de usuario y el backend debe ser segura y eficiente. • La API del backend debe manejar las solicitudes de manera rápida y efectiva, utilizando C# y MongoDB.Driver para gestionar los datos de eventos.
Infraestructura:	<ul style="list-style-type: none"> • La infraestructura para el servicio de autenticación debe estar desplegada y gestionada mediante Terraform en Azure. • Debe asegurarse que los recursos de Azure necesarios para la autenticación están configurados correctamente.