

Informe Desarrollo Web II

(Gestión de Eventos: EventFlow)

Estudiantes

Jessica Nohely Ávila Cruz - T32411145

Estephany Alejandra Portillo Luna – T32411300

Docente:

Erick Roberto Rápalo García

Materia:

Desarrollo Aplicaciones Web II

Sección:

T65

Fecha de entrega:

21/03/2025

Versiones del documento

Fecha	Versión	Descripción	Autor
13-03-25	Número 1	Se realizo la base de datos y la conexión con frontend para el inicio de la sesión para el administrador y usuario	Jessica Nohely Ávila Cruz
16-03-25	Número 2	Se creo las funcionalidades de las pantallas para el usuario	Estephany Alejandra Portillo Luna
19-03-25	Numero 3	Se unió la parte de manejo del administrador y el usuario para la finalización de los procesos del proyecto	Jessica Nohely Ávila Cruz Estephany Alejandra Portillo Luna



Contenido

Informe Desarrollo Móvil I	4
1. DESCRIPCIÓN DE LA SOLUCIÓN	4
1.1 Resumen gerencial	4
1.2 Glosario	4
1.3 Restricciones y metas de arquitectura.....	5
1.4 Diagrama de Proceso.....	6
1.5 Requerimientos del sistema.....	7
1.5.1 Vista de Casos de uso de la aplicación	7
2. Tecnología 8	
2.1 Desarrollo	8
2.2 Modelo Entidad Relación	9

Informe Desarrollo Web II

1. DESCRIPCIÓN DE LA SOLUCIÓN

1.1 Resumen gerencial

"EventFlow" permite administrar de manera eficiente todo el ciclo de vida de un evento. Está diseñada para que el rol de administrador pueda gestionar eventos, presupuestos y tareas asociadas, así como visualizar a los asistentes confirmados y revisar solicitudes de cotización. Gracias a una interfaz simple e intuitiva, el administrador puede mantener el control operativo de cada evento.

Por otro lado, el rol de usuario tiene acceso a funcionalidades clave como visualizar eventos activos, confirmar su asistencia y enviar solicitudes de cotización personalizadas. Esto permite una comunicación directa y clara entre los organizadores del evento y los participantes.

1.2 Glosario

Término	Definición
React Native Web	Una biblioteca que permite ejecutar aplicaciones de React Native en la web.
Estado (State):	Datos que gestionan el comportamiento de un componente, lo que permite que se actualice y vuelva a renderizarse.
Context API:	Método para manejar el estado global en una aplicación sin necesidad de pasar props manualmente a través de todos los niveles del árbol de componentes.
Componentes:	Unidades reutilizables de la interfaz que encapsulan su propia lógica, estilos y estructura.
Context	Es una forma de compartir valores entre componentes sin tener que pasar props manualmente por cada nivel del árbol componente. Es útil para datos que se consideran "globales", como el tema de la UI o la información del usuario autenticado.
Models	Se refiere a una representación estructurada de datos que describe cómo se almacenan y se gestionan. En aplicaciones web, suele estar relacionado con la forma en que se manejan los datos de la base de datos.
Provider	Componente en React que utiliza la Context API para proporcionar valores a todos los componentes en su árbol. Permite que los componentes hijos accedan a los datos sin necesidad de recibirlos como props de forma explícita.

Layout	Sirve para hacer referencia a la estructura y diseño visual de la interfaz de usuario de una aplicación. En React, un archivo de layout a menudo se utiliza para definir la apariencia general de las páginas, incluyendo encabezados, pies de página y navegación, manteniendo una consistencia visual en toda la aplicación
Page	Es un componente que representa una ruta específica en la aplicación. Cada archivo de página generalmente corresponde a una ruta en la URL y es responsable de renderizar el contenido específico para esa ruta.
import	Descripción: Se utiliza para importar módulos, componentes y bibliotecas en un archivo.
export	Descripción: Se utiliza para exportar componentes o funciones desde un módulo, de modo que puedan ser importados en otros archivos.
Functional Component	Descripción: Un componente definido como una función. Puede aceptar props y retornan JSX.
CSS Modules	Descripción: Una forma de escribir CSS que es específica para un componente.
useRouter (Next.js)	Descripción: Hook que permite acceder al enrutador y manejar la navegación en Next.js.

1.3 Restricciones y metas de arquitectura

Restricciones

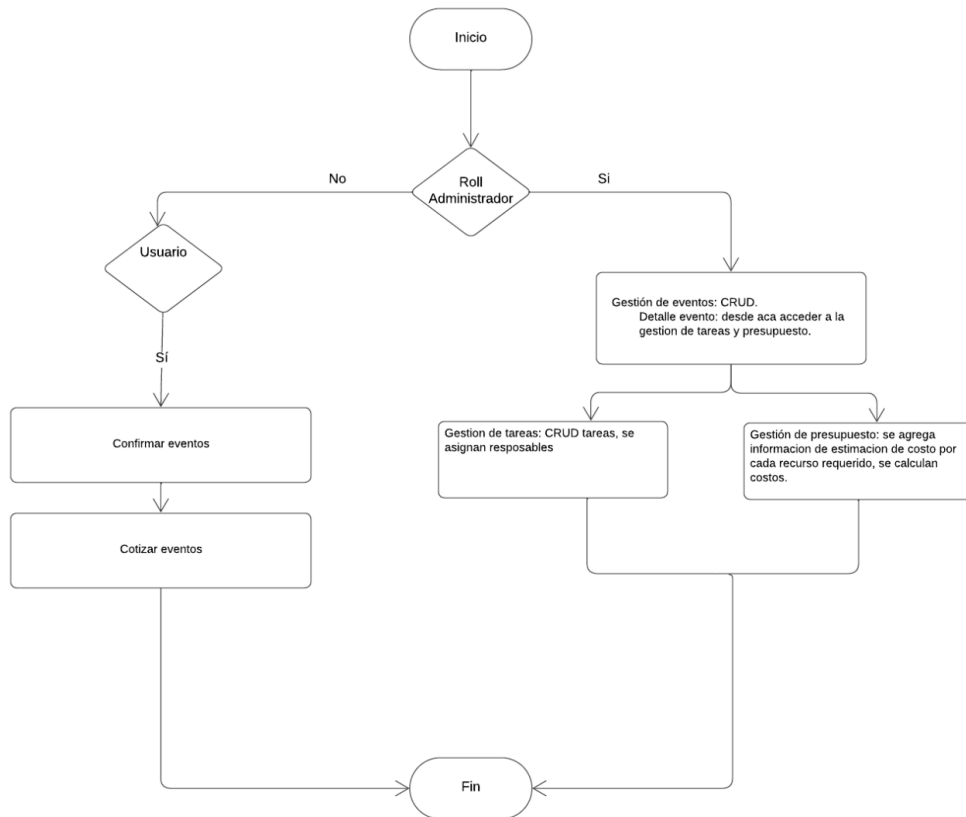
- Una de las principales restricciones que la aplicación web cuenta es que se necesita ingresar al sistema con el correo electrónico y una contraseña, que ya se han registrado como administrador o como usuario regular.
- Tecnologías Definidas: El desarrollo debe realizarse exclusivamente con Next.js para el frontend, Express.js para el backend y MySQL como sistema gestor de base de datos relacional.

Metas de Arquitectura

- Separación de Responsabilidades: El frontend, backend y base de datos deben estar claramente desacoplados, permitiendo el mantenimiento y escalabilidad independiente de cada capa.
- Reutilización de Componentes: Se deben crear componentes y funciones reutilizables para mantener un código limpio, organizado y fácil de mantener.

1.4 Diagrama de Proceso

Recursos de la aplicación para Gestión de Eventos y tareas



1.5 Requerimientos del sistema

1.5.1 Vista de Casos de uso de la aplicación

Nombre del caso de uso	Descripción de la funcionalidad	Ruta del Caso de Uso
Inicio de sesión	Se debe permitir el inicio de sesión con usuario y contraseña. El usuario puede ser administrador o cliente.	• /login
Gestión de eventos	El administrador puede crear, editar, inactivar y cargar la lista de eventos.	• /eventos
Confirmación de asistencia a eventos	Los usuarios pueden visualizar los eventos activos y confirmar su asistencia. El usuario puede ver la lista de eventos a los cuales confirmó asistencia.	• /confirmaciones
Gestión de tareas	El administrador puede asignar tareas a eventos, editar y eliminar tareas.	• /tareas
Gestión de presupuesto	El administrador puede registrar, actualizar y eliminar partidas del presupuesto de un evento.	• /presupuesto
Gestión de asistencia y aprobación de cotizaciones.	El administrador puede visualizar el listado de usuarios que asistirán a un evento específico. El administrador, visualiza las cotizaciones que recibe de los clientes y aprueba o rechaza la cotización,	• /cotizaciones
Solicitar cotización	Los usuarios pueden enviar cotizaciones con detalles del evento que desean organizar. Una vez que el administrador las aprueba o rechaza, el cliente puede ver en su lista, el estado de la misma (aprobado/rechazado)	• /cotizaciones

2. Tecnología

2.1 Desarrollo

Tecnología	Descripción
Next.js	Framework de React que permite la generación de aplicaciones del lado del servidor, mejorando el rendimiento y la optimización. Facilita el enrutamiento y la integración con APIs.
React (Hooks y Context API)	Librería de JavaScript utilizada para la construcción de interfaces de usuario interactivas y dinámicas. Se emplea Context y Provider, para la gestión del estado global sin necesidad de props.
TypeScript	Superset de JavaScript que añade tipado estático, mejorando la mantenibilidad y detección de errores en el código. Se utiliza en todo el proyecto para asegurar mayor robustez.
Express.js	Framework para Node.js que facilita la creación de servidores y API REST, manejando peticiones HTTP de manera eficiente.
MySQL	Sistema de gestión de bases de datos relacional utilizado para almacenar y gestionar la información de eventos, usuarios, cotizaciones y asistencias.
Sequelize ORM	Herramienta de mapeo objeto-relacional (ORM) para Node.js que permite interactuar con la base de datos MySQL de manera sencilla y estructurada.
Tailwind CSS	Framework de CSS de utilidad que permite un diseño responsivo y modular, agilizando la implementación de estilos.
Postman	Plataforma utilizada para probar y documentar las APIs, asegurando la correcta comunicación entre frontend y backend.
Git y GitHub	Sistema de control de versiones y repositorio remoto utilizado para la gestión del código fuente, permitiendo colaboración y seguimiento del desarrollo del proyecto.

2.2 Modelo Entidad Relación

