

Programación en dispositivos móviles (o6MASW)

ACTIVIDAD 2: Aplicación React Native



Miembros del equipo:

Natalia Santos Martínez
Adrián Hernández Monterroso
Ana Regalado Arregui

Git:

<https://github.com/NataliaSantosMart/RhythmMad>

ÍNDICE

1. Introducción.....	2
2. Mockups (MÁXIMO 2 PÁGINAS).....	2
2.1. Añadir un nuevo evento.....	2
2.2. Lista de eventos.....	3
2.3. Detalles de un evento.....	3
3. Aplicación en funcionamiento (MÁXIMO 2 PÁGINAS).....	4
3.1. Splash inicial.....	4
3.2. Añadir un nuevo evento.....	4
3.3. Lista de eventos.....	5
3.4. Detalles de un evento.....	5
4. Conclusiones.....	5

1. Introducción

El presente documento es la memoria de un desarrollo de una aplicación móvil utilizando React Native, enfocada en proporcionar información actualizada sobre eventos en la ciudad de Madrid. La aplicación tiene como objetivo principal facilitar a los usuarios la búsqueda y visualización de eventos, tales como conciertos, exposiciones, conferencias y actividades culturales, entre otros.

El desarrollo de esta aplicación no solo implica la implementación de funcionalidades básicas de navegación y visualización de datos, sino también la integración de características avanzadas para mejorar la experiencia del usuario. Entre estas características se incluyen la capacidad de agregar nuevos eventos a la base de datos Firebase, así como la posibilidad de acceder a información detallada sobre cada evento, incluyendo su descripción, ubicación, fecha y hora, entre otros detalles relevantes.

2. Mockups

2.1. Añadir un nuevo evento

A continuación se puede observar el mockup correspondiente al formulario de añadir un nuevo evento. Este formulario contiene un conjunto de campos a completar: el nombre del evento, precio, fecha, etc. Además, aparece un botón de guardar que permitirá guardar el nuevo evento y que posteriormente aparecerá en el listado de eventos de la aplicación.

Crear evento

RhythmMod

NUEVO EVENTO

Nombre:

Precio: €

Fecha:

Hora:

Ubicación:

Descripción:

Url imagen:

GUARDAR

Lista eventos

Añadir evento

Figura: Mockup Añadir nuevo evento

2.2. Lista de eventos

A continuación, se puede observar el mockup correspondiente al listado de todos los eventos programados en Madrid. Este listado incluye información como el día y hora del evento, además de la ubicación y nombre de éste. Al presionar cualquiera de los eventos del listado se accederá a la pantalla de los detalles del evento.

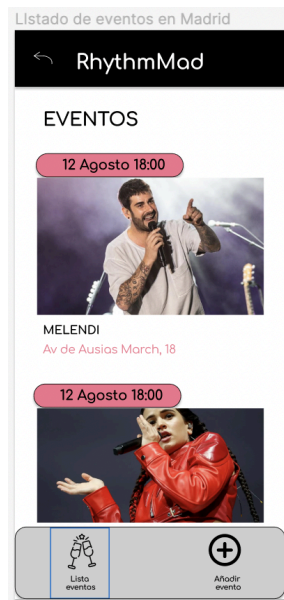


Figura: Mockup Lista de eventos

2.3. Detalles de un evento

A continuación, se puede observar el mockup correspondiente a los detalles de un evento. Este listado incluye toda la información relativa al evento.

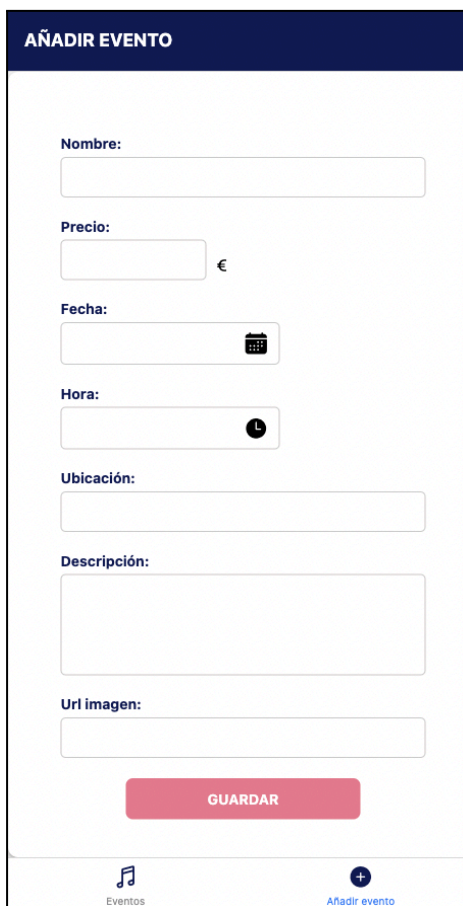


3. Aplicación en funcionamiento

3.1. Splash inicial

Se implementa un SplashScreen, que muestra una imagen de logotipo de la aplicación durante tres segundos al iniciar la aplicación. Esto se logra mediante el uso de un estado local `isLoading`, que se activa inicialmente y luego se desactiva después de tres segundos utilizando `useEffect`. Una vez desaparece el Splash, la aplicación navega a la pantalla principal que contiene una lista de eventos y la opción para agregar un nuevo evento.

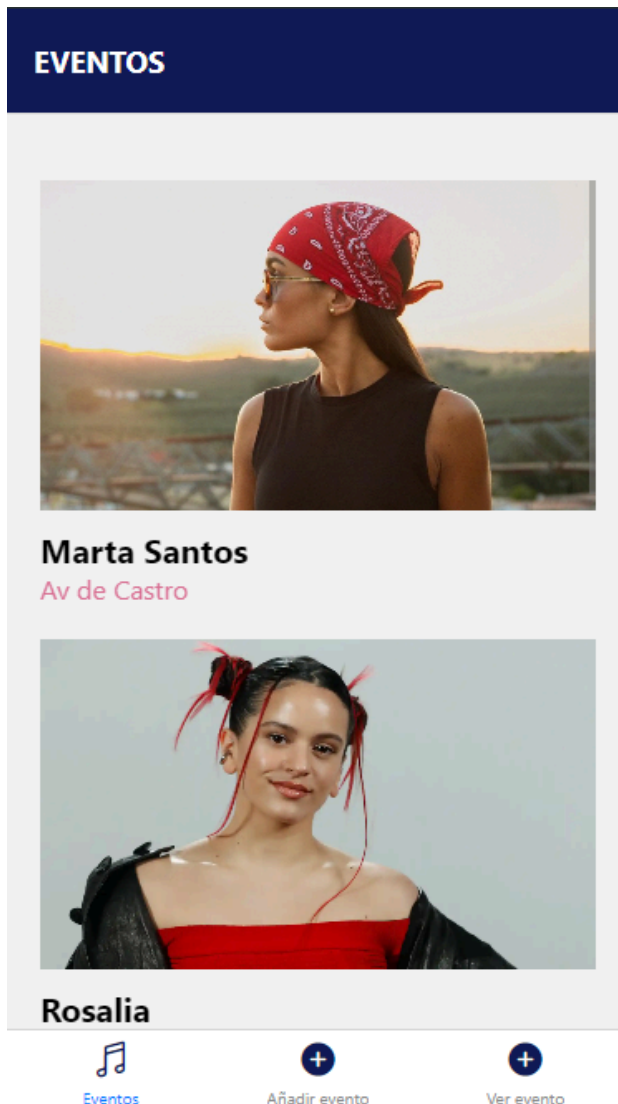
3.2. Añadir un nuevo evento



La pantalla de añadir un nuevo evento recoge campos como nombre, precio, fecha, etc. Se utiliza el estado local y funciones para manejar la entrada de usuario, incluyendo la selección de fecha y hora mediante “`DateTimePickerModal`.” Además, el formulario se ha adentrado en un “`ScrollView`” para que en caso de encontrarnos con dispositivos más pequeños se permita hacer un scroll de la vista.

Por otro lado, cuando se presiona el botón de guardar que se corresponde con un “`TouchableOpacity`” personalizado, se valida que todos los campos estén completados, en caso de no estarlo se muestra un mensaje de error y en caso contrario aparece una alerta de éxito. Posteriormente, los datos serán guardados en Firebase.

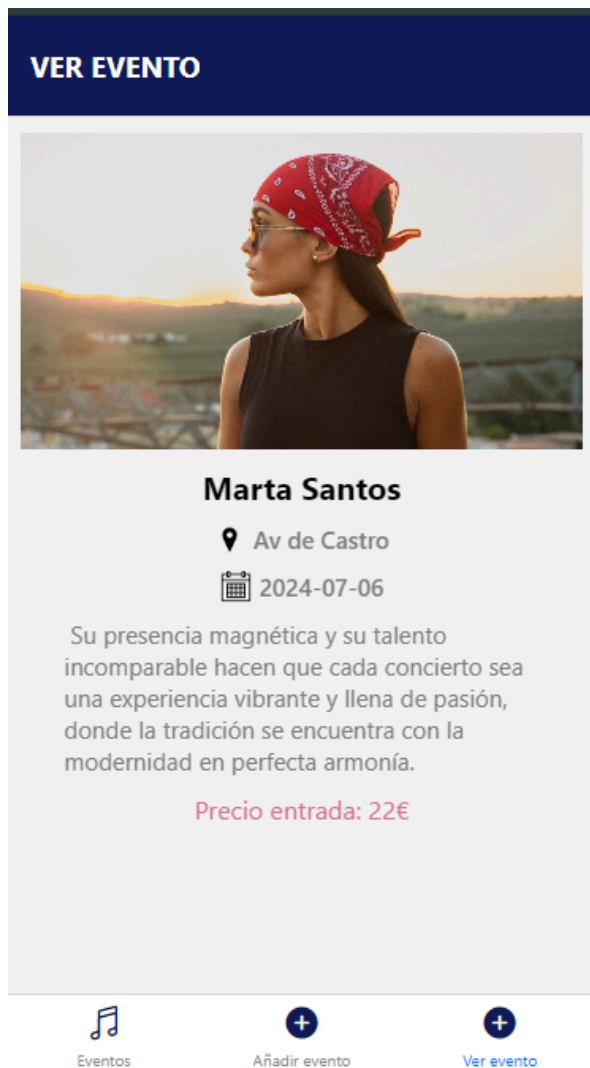
3.3. Lista de eventos



La pantalla de lista de eventos muestra una lista de eventos disponibles, obtenidos de una base de datos en tiempo real. Para lograr esto, se utiliza el hook `useState` para gestionar el estado local de la aplicación, específicamente para almacenar los datos recuperados de Firebase Firestore. Mediante el hook `useEffect`, se realiza una solicitud a la base de datos para obtener la colección de eventos. La función `fetchData` se ejecuta cuando el componente se monta por primera vez y se encarga de recuperar los datos de la base de datos. Una vez que se reciben los datos, se actualiza el estado local mediante `setDataSetServer`, lo que provoca una actualización de la interfaz de usuario para reflejar los eventos recuperados.

La pantalla utiliza un componente `FlatList` para mostrar los eventos, cada uno representado por un elemento de la lista. Cada evento incluye una imagen y detalles como el nombre y la ubicación. La información se obtiene de Firebase Firestore y se actualiza automáticamente en la pantalla cuando hay cambios en la base de datos. La pantalla está diseñada para ser fácilmente accesible desde la barra de navegación principal de la aplicación, utilizando un ícono y un título descriptivo.

3.4. Detalles de un evento



Esta es la pantalla de ver los detalles de un evento. Para ello a partir de la vista de lista de eventos hemos envuelto cada evento en un “TouchableOpacity” y además se pasará el ítem “evento” por props gracias al hook de useNavigation de react-navigation.

Una vez tenemos todos los datos necesarios en la nueva pantalla, se ha creado un “ScrollView” para poder scrollear por si los datos ocupan más del alto de la pantalla y se han pintado todos los datos pasados por props.

4. Conclusiones

El desarrollo de la aplicación móvil de eventos de Madrid con React Native ha ampliado nuestro conocimiento en el campo del desarrollo de aplicaciones móviles. Trabajar en equipo ha sido fundamental, permitiéndonos colaborar eficazmente para alcanzar nuestros objetivos.

A lo largo del proyecto, hemos aprendido a utilizar tecnologías avanzadas como React Native y Firebase Firestore, integrando estas herramientas para crear una aplicación intuitiva y funcional. La creación de una interfaz de usuario atractiva y la implementación de características como la visualización de eventos han mejorado la experiencia del usuario.