

INDICE

[1. Introducción 2](#_Toc163693762)

[1.1 Objetivos 2](#_Toc163693763)

[1.2 Tecnologías empleadas 2](#_Toc163693764)

[1.3 Tecnologías empleadas 2](#_Toc163693765)

[2. Metodología de desarrollo 3](#_Toc163693766)

# Introducción

Actualmente, el desarrollo de aplicaciones móviles ha alcanzado una relevancia importante, cambiando la manera en que accedemos y gestionamos información. Este proyecto de fin de grado se enfoca en la creación de una aplicación multiplataforma destinada a la exploración y gestión de datos sobre museos. Esta aplicación pretende abordar la creciente demanda de acceso rápido y sencillo a información cultural, convirtiéndose en una herramienta esencial para aficionados al arte y la historia.

A lo largo de este documento explicaré los pasos del desarrollo de mi aplicación, desde los requisitos necesarios hasta el diseño de la arquitectura. El documento también abordará las cuestiones tecnológicas que determinan la elección de plataformas y herramientas, así como los servicios de back-end diseñados para garantizar el rendimiento adecuado de una aplicación.

## 1.1 Objetivos

El objetivo principal de esta iniciativa es facilitar el acceso a información detallada sobre museos, uniendo dicha información en una única plataforma y accesible. Además, se busca atraer a la generación actual, muy conectada a dispositivos móviles, mediante la creación de un espacio cívico digital que permita la consulta de información y la reserva de actividades culturales de manera fácil y eficiente. Este método innovador busca fomentar el interés y la participación activa en el ámbito cultural para promover el enriquecimiento personal y social a través del acceso a experiencias culturales significativas.

## Tecnologías empleadas

La aplicación se ha desarrollado en JavaScript utilizando React Native como framework para el desarrollo del front-end, y Node.js con Express.js para el backend. Para el almacenamiento de datos de la aplicación, se ha decidido utilizar MySQL como sistema de gestión de bases de datos relacional. MySQL proporciona un almacenamiento seguro y escalable para los datos de la aplicación, lo que garantiza su disponibilidad y confiabilidad.

**React Native**

React Native es un framework de desarrollo de aplicaciones móviles creado por Facebook en 2015. A diferencia de otros frameworks como Java para Android o Swift para iOS, React Native permite desarrollar aplicaciones móviles multiplataforma utilizando JavaScript y React, un marco de trabajo para la construcción de interfaces de usuario.

Este enfoque ofrece varias ventajas significativas. En primer lugar, React Native permite a los desarrolladores crear una única base de código que puede ejecutarse en múltiples plataformas, como Android e iOS. Esto significa que los desarrolladores pueden escribir una vez y ejecutar en cualquier lugar, lo que resulta en un desarrollo más rápido y eficiente.

Además, React Native utiliza una arquitectura basada en componentes, lo que facilita la reutilización de código y la construcción de interfaces de usuario consistentes en toda la aplicación. Los componentes de React Native están diseñados para ser altamente modulares y personalizables, lo que permite a los desarrolladores crear experiencias de usuario únicas y atractivas.

Otra ventaja importante de React Native es su capacidad para acceder a las API nativas del dispositivo a través de JavaScript. Esto significa que los desarrolladores pueden integrar fácilmente características nativas, como la cámara, el GPS y los sensores del dispositivo, en sus aplicaciones sin necesidad de escribir código nativo separado para cada plataforma.

**MySQL**

Las bases de datos que se van a utilizar son relacionales y estarán basadas en MySQL. Este sistema de gestión de bases de datos relacionales sigue un formato estructurado basado en tablas. A diferencia de las bases de datos NoSQL, que están orientadas a documentos, MySQL almacena datos en tablas con un esquema predefinido.

La ventaja que nos ofrece MySQL es su robustez y amplio soporte en la industria. Aunque menos flexible que las bases de datos NoSQL, MySQL sigue siendo una opción confiable para aplicaciones que requieren integridad de datos y consultas complejas.

Una de las ventajas de MySQL es que permite definir un esquema de base de datos específico que debe seguirse para almacenar datos. Esto proporciona una estructura definida para los datos y garantiza la coherencia en la forma en que se almacenan y se acceden a los datos.

**Express.js y Node.js:**

Para el desarrollo del back-end de la aplicación, se utilizará Node.js junto con Express.js. Node.js proporciona un entorno de ejecución de JavaScript del lado del servidor, mientras que Express.js es un marco de aplicación web para Node.js que simplifica el desarrollo de API.

MySQL será la base de datos utilizada para almacenar y gestionar los datos de la aplicación. Se aprovechará su estructura relacional basada en tablas para modelar y almacenar la información de usuarios, museos y reservas.

Express.js será fundamental para implementar las rutas del back-end y manejar las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) en la base de datos MySQL. Cada ruta del back-end corresponderá a una entidad específica, como Usuarios, Administradores, Museos y Reservas, permitiendo así interactuar con los datos a través de solicitudes HTTP.

Los controladores en Express.js serán responsables de manejar las solicitudes entrantes y realizar operaciones en la base de datos MySQL. Se integrará la conexión a la base de datos MySQL para almacenar y recuperar información, y las rutas del back-end actuarán como una API REST que permitirá que el front-end realice solicitudes HTTP para interactuar con los datos de manera segura y eficiente.

**GIT**

Es un sistema de control de versiones que permite ir guardando los cambios que vas haciendo en tu código en la nube, creando un historial que te permite volver a un punto determinado si es necesario. GIT se puede usar en conjunto con herramientas para alojar el código como GitHub para facilitar que tu código esté disponible en todo momento. En mi caso lo he utilizado como backup para evitar perder el proyecto si se me rompía el ordenador o por si tenía la necesidad de trabajar desde otro ordenador poder tener disponible el código. Por otra parte, GIT me ha ayudado para ir viendo los cambios en el código, y me permite volver atrás en el caso de que haya cometido algún fallo y hubiera borrado cosas que no quería borrar.

## Tecnologías empleadas

**Visual Studio Code**

Visual Studio Code es un editor de código fuente desarrollado por Microsoft que se ha vuelto muy popular entre los desarrolladores de software. Está disponible de forma gratuita para Microsoft Windows, macOS y GNU/Linux bajo Licencia MIT.

Es un entorno de desarrollo integrado (IDE) ligero pero potente que ofrece una amplia gama de funcionalidades para el desarrollo de software. Aunque no está específicamente diseñado para el desarrollo en Android como Android Studio, Visual Studio Code es altamente personalizable y admite extensiones que pueden integrarse con herramientas específicas para Android, como emuladores y herramientas de depuración.

La razón principal para elegir Visual Studio Code es su flexibilidad y su amplio ecosistema de extensiones que permiten adaptarlo a las necesidades específicas del desarrollo de aplicaciones multiplataforma. Además, al estar familiarizado con Visual Studio Code por haberlo usado durante el grado de Desarrollo de Aplicaciones Multiplataforma, esto ayuda a acelerar el proceso de desarrollo y hacerlo más eficiente.

**MySQL Workbench**

MySQL Workbench es una herramienta de diseño y gestión de base de datos MySQL. Esta aplicación es esencial para diseñar la estructura de tu base de datos relacional. Permite crear, modificar y eliminar tablas, así como ejecutar consultas SQL y realizar tareas de administración como la optimización de consultas. Con MySQL Workbench, se puede visualizar y modelar nuestra base de datos de una manera intuitiva, lo que facilita el diseño y la administración de bases de datos MySQL.

**Postman**

Postman es una herramienta popular para probar y documentar API REST. Permite enviar solicitudes HTTP a la API y ver las respuestas en tiempo real. Postman es esencial para probar la funcionalidad de la API y asegurarse de que esté funcionando correctamente antes de implementarla en la aplicación. Además, Postman ofrece características de documentación que te permiten describir y explicar cada endpoint de tu API, lo que facilita su comprensión y uso por parte de otros desarrolladores.

**Expo CLI**

Expo CLI es una herramienta de línea de comandos para iniciar, desarrollar y publicar proyectos de Expo en React Native. Expo es una plataforma que simplifica el desarrollo de aplicaciones móviles al proporcionar un conjunto de herramientas y servicios integrados, como un entorno de desarrollo, un conjunto de componentes UI predefinidos y acceso a API nativas. Expo CLI te permite iniciar un nuevo proyecto de Expo, desarrollar tu aplicación con facilidad y publicarla en las tiendas de aplicaciones. Es una herramienta imprescindible para el desarrollo y la implementación de aplicaciones React Native utilizando Expo.

# Metodología de desarrollo

# Análisis de Requisitos

# Diseño del Sistema

# Diseño del Programa

# Codificación

# Ejecución de Pruebas

# Verificación

# Mantenimiento

