

JUSTIFICACIONES

JUSTIFICACIÓN TFG	2
JUSTIFICACIÓN DE REACT NATIVE Y EXPO COMO FRAMEWORK DE FRONT-END	5
JUSTIFICACIÓN POR LA QUE NOS QUEDAMOS CON UNA SOLA PARTE DE LA API	7
JUSTIFICACIÓN DEL USO DE EXPRESS COMO FRAMEWORK PARA EL BACKEND	8
JUSTIFICACIÓN PARA EL USO DE FIREBASE AUTHENTICATION EN NUESTRA APLICACIÓN REACT NATIVE	9

JUSTIFICACIÓN TFG

En la actualidad, el acceso rápido y eficiente a la información es fundamental para fomentar una sociedad informada y avanzada. El Instituto Nacional de Estadística (INE) desempeña un papel crucial al proporcionar datos precisos y relevantes que sirven como base para la

toma de decisiones en diversos sectores. En este contexto, resulta imperativo que el INE adopte una estrategia tecnológica que asegure la accesibilidad y usabilidad de su información para todos los ciudadanos.

A continuación, se presentan varias razones que respaldan la implementación de una aplicación multiplataforma en lugar de la web convencional del INE:

-Mejora en la experiencia de usuario: una aplicación multiplataforma del INE permitirá al usuario tener una mejor experiencia en cualquier dispositivo, y cualquier sistema operativo.

-Acceso directo: a diferencia de las webs, las aplicaciones multiplataforma no dependen del uso del navegador de nuestro dispositivo. Estas aplicaciones pueden ser descargadas e instaladas directamente en nuestros dispositivos, lo que crea un acceso directo en la pantalla principal. Esto proporciona al usuario la capacidad de acceder rápidamente a los servicios de la aplicación que necesitan sin tener la necesidad de tener que abrir el navegador y buscar un sitio web específico.

-Conectividad: contrario a las páginas web que requieren una conexión a internet para funcionar, las aplicaciones multiplataforma pueden ofrecer funcionalidad básicas sin conexión a internet. Esto garantiza que los usuarios puedan acceder a los servicios de la aplicación incluso en condiciones de conectividad limitada e incluso sin conectividad a internet.

-Adaptabilidad a futuras tecnologías: es una de las características más destacadas de las aplicaciones multiplataforma. Estas aplicaciones se construyen sobre una única base de código, lo que simplifica enormemente el proceso de incorporar nuevas funcionalidades o realizar modificaciones, esto además supondrá un ahorro económico para la empresa. Esta eficiencia se debe a que los cambios pueden implementarse de forma sencilla y ágil en todas las plataformas para las cuales se ha desarrollado la aplicación.

-Notificaciones: las aplicaciones multiplataforma nos permitirán el envío de notificaciones al usuario que tenga descargada la aplicación y esto hará que el usuario esté más involucrado con la aplicación. Ya que el INE podrá informar a los usuarios sobre nuevos informes, datos actualizados sobre distintos estudios importantes relacionados con estadísticas a través de la notificación en sus dispositivos.

-Mayor interacción con el usuario: la aplicación multiplataforma tiene el potencial de ofrecer una mayor interacción con los usuarios a través de un servicio innovador. Este servicio permite a los usuarios completar encuestas relacionadas con datos de interés para el INE, recibiendo a cambio un pequeño incentivo económico. Este enfoque no solo facilita el acceso a la información del INE para los usuarios, sino que también los involucra activamente al permitirles contribuir con datos relevantes.

En resumen, las aplicaciones multiplataforma proporcionan una visibilidad mayor, una mejor interacción con los usuarios, una mayor retención y accesibilidad sin conexión, lo que resulta en un alcance más extenso en comparación con las webs tradicionales. Además, la integración de futuras funcionalidades es más sencilla y económica en una aplicación multiplataforma. Estas características hacen que las aplicaciones multiplataforma sean una opción atractiva para las empresas que buscan llegar a una audiencia más amplia.

JUSTIFICACIÓN DE REACT NATIVE Y EXPO COMO FRAMEWORK DE FRONT-END

Para el desarrollo de la aplicación multiplataforma del Instituto Nacional de Estadística, hemos considerado usar el framework React Native con Expo. React Native es un framework de código abierto multiplataforma diseñado para el desarrollo de aplicaciones móviles nativas, basado en JavaScript y React. Expo es una plataforma que trabaja en conjunto con React Native, proporcionando herramientas y servicios que facilitan y aceleran el desarrollo y la implementación de aplicaciones.

Una de las razones por las que hemos usado React Native y Expo es porque tienen las siguientes ventajas:

- **Código abierto:** React Native y Expo nos ofrecen la oportunidad de utilizar sus plataformas de forma gratuita.
- **Multiplataforma:** React Native con Expo nos permite el desarrollo de aplicaciones para Android, iOS y la web mediante soluciones adicionales como React Native Web.
- **Desarrollo rápido y eficiente:** React Native y Expo nos posibilitan el desarrollo rápido y eficiente al usar tecnologías web estándares como JavaScript y React. Además, Expo cuenta con una gran biblioteca de componentes preconstruidos y APIs que facilitan el desarrollo.
- **Desarrollo Nativo:** React Native permite la creación de aplicaciones móviles con rendimiento cercano al nativo, ya que se compilan a código nativo.
- **Gran comunidad y soporte:** React Native y Expo tienen una comunidad activa y un soporte técnico robusto. Además, podremos encontrar una gran cantidad de recursos en línea, tutoriales, documentación y componentes preconstruidos creados por la comunidad para facilitar el desarrollo de nuestra aplicación.

Antes de decidirnos por React Native con Expo como el framework para nuestra aplicación multiplataforma, evaluamos varias opciones como Laravel, Xamarin, IONIC y Ruby. Al final, optamos por React Native con Expo por su enfoque en las tecnologías web estándar y su capacidad para crear interfaces de usuario atractivas. Además, la posibilidad de desarrollar aplicaciones con rendimiento nativo fue un factor determinante en nuestra elección.

Por último, se muestra una tabla de la comparación de los frameworks nombrados anteriormente:

	React Native con Expo	IONIC	Laravel	Xamarin	Ruby
Lenguaje de Programación	JavaScript	TypeScript	PHP	C#	Ruby
Plataforma Soportadas	Web,Android, IOS	Android, IOS y Android	Web	Android, IOS y Windows	Android, IOS y Windows
Desarrollo Nativo	No	No	No	Sí	No
Desarrollo Web desde la misma base de código	No	Si	Si	No	Si
Biblioteca de Componentes	Si	Si	No	Si	Si
Actualizaciones	Actualizaciones frecuentes	Actualizaciones Frecuentes	Actualizaciones Regulares	Actualizaciones Regulares	Actualizaciones frecuentes
Coste	Gratis	Gratis	Gratis	Requiere licencia	Gratis

JUSTIFICACIÓN POR LA QUE NOS QUEDAMOS CON UNA SOLA PARTE DE LA API

Inicialmente, la aplicación estaba destinada a cubrir todas las áreas del Instituto Nacional de Estadística (INE). Sin embargo, finalmente se decidió enfocar el desarrollo en un área específica: la demografía y la población. Esta decisión se tomó debido a la ineficiencia, la mala estructura y la falta de datos de la API del INE.

Durante la fase de investigación sobre la API, descubrimos que las operaciones necesarias para extraer series, tablas y sus datos correspondientes no estaban disponibles. Por lo tanto, optamos por centrarnos en la parte de la API que proporcionaba la mayor cantidad de datos útiles. Esta área, enfocada en demografía y población, permitirá un desarrollo más efectivo y un uso más significativo de la información disponible.

Por otro lado, hemos decidido crear una base de datos que contenga todos los datos disponibles del INE a través de la API (actualizados al 10/07/2024), excluyendo los datos de series y tablas debido a la gran cantidad de opciones y volumen de información, lo que hace inviable su almacenamiento completo en la base de datos.

Esta decisión se tomó porque incluso consultas simples para obtener un valor o un conjunto de valores podían tardar entre 5 y 10 minutos en retornar una respuesta a través de la API, lo cual es ineficiente para nuestros propósitos. Al transferir estos datos a una base de datos, podemos realizar consultas más selectivas y detalladas de manera mucho más eficiente.

Para la creación de la base de datos, desarrollamos varios scripts en Python que realizaron todas las llamadas a la API e introdujeron los datos en la base de datos de forma automatizada. Esto nos permitirá manejar la información de manera más ágil y precisa.

JUSTIFICACIÓN DEL USO DE EXPRESS COMO FRAMEWORK PARA EL BACKEND

Por último para el desarrollo del backend de la aplicación hemos optado por utilizar Express como el framework del backend. Express es un framework web flexible para Node.js y presenta varias ventajas que se alinean con nuestras necesidades y objetivos. A continuación se detallan las razones que justifican nuestra elección:

- **Simplicidad y Minimalismo:** Express proporciona una estructura simple y clara que facilita la creación y mantenimiento de aplicaciones web. Su diseño minimalista nos permite agregar solo las funcionalidades necesarias, evitando la complejidad innecesaria y manteniendo el código limpio y manejable.
- **Enrutamiento Potente:** Express ofrece un sistema de enrutamiento robusto que facilita la definición y manejo de rutas para diferentes endpoints de la aplicación. Esto es especialmente útil para organizar nuestra API de manera clara y estructurada, permitiendo un acceso más eficiente y coherente a los datos.
- **Flexibilidad:** Express no impone una estructura rígida para las aplicaciones, lo que nos permite organizar nuestro código de la manera que mejor se adapte a nuestras necesidades específicas. Esta flexibilidad es esencial para integrar diferentes componentes y servicios en nuestro backend.
- **Compatibilidad y Ecosistema:** Al ser parte del ecosistema de Node.js, Express se integra fácilmente con muchas otras bibliotecas y frameworks. Esto nos permite ampliar sus funcionalidades según las necesidades del proyecto.
- **Comunidad Activa y Soporte:** Express cuenta con una comunidad activa y un soporte técnico robusto. Podemos encontrar una gran cantidad de recursos en línea, tutoriales y documentación que facilitan la resolución de problemas y la implementación de nuevas funcionalidades.

Para concluir utilizar Express para el backend de nuestra aplicación nos proporciona una solución eficiente, flexible y bien soportada para manejar las necesidades específicas de nuestro proyecto. Su simplicidad, potente enrutamiento y la capacidad de integración con otras herramientas hacen de Express la elección ideal para desarrollar una aplicación backend robusta y escalable.

JUSTIFICACIÓN PARA EL USO DE FIREBASE AUTHENTICATION EN NUESTRA APLICACIÓN REACT NATIVE

Para la implementación del sistema de autenticación en nuestra aplicación React Native, hemos decidido utilizar Firebase Authentication por varias razones que son las que se muestran a continuación.

En primer lugar, Firebase Authentication ofrece una facilidad de integración excepcional con React Native. Google proporciona una biblioteca oficial de Firebase para React Native, lo que simplifica enormemente el proceso de implementación de servicios de autenticación en nuestra aplicación. Esta integración sin complicaciones nos permite enfocarnos en otros aspectos del desarrollo sin tener que preocuparnos por la complejidad de la configuración de la autenticación.

Además, Firebase Authentication soporta múltiples plataformas, lo que permite a los usuarios autenticarse en dispositivos Android, iOS y en la web sin necesidad de desarrollar soluciones separadas para cada plataforma. Esta capacidad multiplataforma es crucial para asegurar una experiencia de usuario coherente y fluida en todos los dispositivos.

Una de las mayores ventajas de Firebase Authentication es su soporte para una amplia variedad de proveedores de autenticación. Los usuarios pueden iniciar sesión utilizando métodos como correo electrónico y contraseña, así como a través de servicios como Google, Facebook y X. Esta flexibilidad nos permite ofrecer múltiples opciones de inicio de sesión, mejorando así la accesibilidad y conveniencia para nuestros usuarios.

La seguridad y escalabilidad son también factores clave en nuestra decisión. Firebase Authentication emplea prácticas de seguridad robustas y escalables, asegurando que las credenciales de los usuarios se manejen de manera segura y eficiente.

Otra ventaja significativa es la simplificación en la gestión de usuarios que proporciona Firebase. Herramientas integradas para la verificación de correo electrónico, restablecimiento de contraseñas y administración de perfiles de usuario facilitan enormemente el desarrollo y mantenimiento del sistema de autenticación.

En conclusión, el uso de Firebase Authentication en nuestra aplicación React Native nos proporciona una solución segura, escalable y fácil de integrar para la gestión de la autenticación de usuarios. Su capacidad para soportar múltiples proveedores de autenticación y la simplicidad en la gestión de usuarios hacen de Firebase Authentication la opción ideal para nuestra aplicación, permitiéndonos ofrecer una experiencia de usuario superior y eficiente.