

# Viajes EasyTravels

El proyecto "Viajes EasyTravels" es una aplicación web desarrollada durante el curso escolar 2023-2024 bajo la tutoría de Antonio Gabriel González Casado y realizada por Pablo Merinas Soto

## Descripción del Proyecto

"Viajes EasyTravels" es una aplicación web diseñada para permitir a los usuarios documentar y gestionar un registro personal de los países, ciudades y atracciones que han visitado. La aplicación se propone como un diario de viajes digital, donde los usuarios pueden añadir nuevos destinos visitados, visualizar una lista completa de sus viajes y acceder a información detallada sobre cada lugar. Con esta aplicación, se facilita la recopilación de recuerdos y experiencias de viaje en un único lugar, resolviendo el problema de mantener un seguimiento ordenado y accesible de los lugares visitados y cubriendo la necesidad de documentar detalladamente los viajes realizados.

## Objetivo del Proyecto

El objetivo principal de "Viajes EasyTravels" es proporcionar a los usuarios una plataforma intuitiva y eficiente para documentar y gestionar sus viajes de manera organizada. La aplicación actúa como un diario de viajes digital, permitiendo a los usuarios añadir, visualizar y gestionar sus experiencias de viaje de manera fácil y rápida.

## Alcance del Proyecto

- Creación y gestión de una lista personalizada de países, ciudades y atracciones visitadas.
- Provisión de información relevante y experiencias personales sobre cada destino.
- Funcionalidades específicas para dos tipos de usuarios: CUSTOMER y ADMIN.
- Usuarios CUSTOMER: Pueden añadir nuevos lugares visitados, visualizar su historial de viajes y acceder a información detallada de los lugares.
- Usuarios ADMIN: Tienen todas las funcionalidades de los usuarios CUSTOMER, además de privilegios adicionales para gestionar el sistema, administrar usuarios, países, ciudades, atracciones, realizar operaciones de mantenimiento en la base de datos y gestionar el contenido global de la aplicación.

# Breve análisis/comparativa con las alternativas del mercado

A diferencia de otras aplicaciones del mercado que pueden abrumar a los usuarios con una amplia gama de funciones y opciones, easyTravels se centra en ofrecer una experiencia de usuario fluida y sin complicaciones.

## Ventajas de easyTravels:

- **Simplicidad:** La interfaz de easyTravels es intuitiva, lo que permite a los usuarios navegar fácilmente por las opciones disponibles sin necesidad de una curva de aprendizaje pronunciada.
- **Enfoque Directo:** La aplicación se centra en dos funcionalidades principales: marcar sitios como visitados y proporcionar información detallada sobre los destinos. Este enfoque directo evita la sobrecarga de información y funciones, haciendo que la aplicación sea más accesible para los usuarios.
- **Facilidad de Uso:** Con funcionalidades claras y bien definidas los usuarios pueden fácilmente encontrar lo que buscan sin distracciones innecesarias.
- **Personalización:** A través de la selección de ciudades y atracciones específicas, los usuarios pueden personalizar su experiencia de viaje, marcando los lugares que han visitado.

## Comparativa con Alternativas del Mercado:

- **TripAdvisor o Google Trips:** A diferencia de estas plataformas, que ofrecen una amplia gama de servicios incluyendo reservas de hoteles, vuelos, y opiniones de usuarios, easyTravels se enfoca en la simplicidad, ofreciendo una experiencia más enfocada y menos saturada de información.
- **Roadtrippers:** Mientras que Roadtrippers se centra en la planificación detallada de rutas de viaje, easyTravels ofrece una experiencia más simplificada, ideal para usuarios que prefieren una interfaz menos compleja y más directa.
- **PackPoint:** PackPoint es una aplicación orientada a la organización de equipaje basada en el destino y la duración del viaje. easyTravels, por otro lado, se centra en la exploración de destinos y atracciones, complementando aplicaciones como PackPoint en lugar de competir directamente con ellas.

En conclusión, easyTravels se destaca en el mercado por su simplicidad y enfoque directo, ofreciendo una herramienta útil para aquellos viajeros que buscan una manera sencilla de marcar y recordar los sitios visitados, así como obtener información relevante sobre sus próximos destinos.

# Stack Tecnológico

El stack tecnológico de este proyecto se compone de varias tecnologías y herramientas utilizadas tanto en el backend como en el frontend. A continuación, se detalla el stack tecnológico basado en los archivos de configuración y código fuente proporcionados:

## Backend (API)

- Spring Boot: Un framework de Java para la creación de aplicaciones web y microservicios de manera rápida y sencilla. Se utiliza como la base para construir la API del proyecto.
- Versión especificada en el pom.xml: 3.2.2.
- Spring Data JPA: Un subproyecto de Spring que facilita la implementación de repositorios basados en JPA para el acceso a datos.
- Spring Security: Proporciona autenticación y autorización a las aplicaciones de Spring.
- MySQL: El sistema de gestión de bases de datos relacional utilizado para almacenar los datos de la aplicación.
- Driver especificado: mysql-connector-j versión 8.3.0.
- Lombok: Una biblioteca de Java que se utiliza para minimizar el código repetitivo como getters, setters, constructores, entre otros, mediante anotaciones.
- Logback y SLF4J: Utilizados para el manejo de logs en la aplicación.
- JUnit y Mockito: Para la realización de pruebas unitarias y de integración.
- JWT (JSON Web Tokens): Para la implementación de autenticación y autorización basada en tokens.
- Maven: Un sistema de gestión y comprensión de proyectos de software. Se utiliza para gestionar las dependencias del proyecto, construcción, y despliegue del backend.

## Frontend

- React: Una biblioteca de JavaScript para construir interfaces de usuario.
- React Router: Utilizado para manejar el enrutamiento en aplicaciones web construidas con React.
- HTML5 y CSS3: Para la estructura y el diseño de las páginas web.
- JavaScript: El lenguaje de programación utilizado para la lógica del lado del cliente.
- npm (Node Package Manager): Gestor de paquetes para JavaScript, utilizado para manejar las dependencias del proyecto frontend.
- Git: Sistema de control de versiones para el manejo del código fuente del proyecto.
- JSON: Formato utilizado para el intercambio de datos entre el frontend y el backend.

## Identificación de las necesidades del proyecto.

### Backend (API de Viajes)

- Gestión de Dependencias: Utiliza Maven como herramienta de gestión de dependencias, lo cual es esencial para manejar las bibliotecas y frameworks necesarios para el proyecto, como Spring Boot, JPA, seguridad.
- Seguridad: Implementa Spring Security para manejar la autenticación y autorización dentro de la aplicación. Esto incluye la configuración de roles y la generación de tokens JWT para los usuarios. ([UserSecurityService](#))
- Persistencia de Datos: Utiliza Spring Data JPA para interactuar con la base de datos, lo cual facilita la implementación de operaciones CRUD sobre las entidades del modelo de datos. ([UserEntity](#))
- Inicialización de Datos: Cuenta con una clase para la inicialización de datos predeterminados en la base de datos, lo cual es útil para pruebas y para establecer un estado inicial de la aplicación. ([DataInitializer](#))
- Controladores REST: Implementa controladores REST para exponer operaciones relacionadas con usuarios, roles, y otras entidades a través de una API HTTP. ([UserRestController](#))

## Frontend (Aplicación de Viajes)

- Interfaz de Usuario: Utiliza React para construir una interfaz de usuario dinámica y responsiva. Esto incluye componentes para la visualización y gestión de información personal, atracciones turísticas, entre otros. ([PersonalInfo](#))
- Manejo de Estados: La aplicación debe manejar estados de forma eficiente para reflejar los cambios en la interfaz de usuario basados en interacciones del usuario o respuestas del servidor.
- Autenticación y Autorización: Necesita integrar la autenticación y autorización con el backend, posiblemente utilizando tokens JWT para mantener la sesión del usuario y controlar el acceso a diferentes partes de la aplicación.
- Rutas y Navegación: Implementar un sistema de rutas para navegar entre diferentes vistas y componentes de la aplicación de manera eficiente.
- Responsive Design: Asegurar que la aplicación sea accesible y usable en diferentes dispositivos y tamaños de pantalla.
- Internacionalización: Considerar la posibilidad de soportar múltiples idiomas para llegar a un público más amplio.
- Optimización y Despliegue: Optimizar la aplicación para producción y configurar el despliegue continuo para facilitar las actualizaciones y el mantenimiento.

## Esquema de la BD

### TABLAS

t\_attraction:

- id (PK): Identificador único de la atracción (entero grande, autoincremental).
- price: Precio de la atracción (doble).
- city\_id (FK): ID de la ciudad a la que pertenece la atracción (entero grande).
- attraction: Nombre de la atracción (cadena de caracteres, máximo 255 caracteres).
- img\_url: URL de la imagen de la atracción (cadena de caracteres, máximo 255 caracteres).
- info: Información adicional sobre la atracción (cadena de caracteres, máximo 255 caracteres).

- category: Categoría de la atracción (enumeración: LANDMARK, MUSEUM, CHURCH, NATIONAL\_PARK).

t\_city:

- id (PK): Identificador único de la ciudad (entero grande, autoincremental).
- population: Población de la ciudad (entero).
- country\_id (FK): ID del país al que pertenece la ciudad (entero grande).
- airport\_code: Código del aeropuerto principal de la ciudad (cadena de caracteres, máximo 255 caracteres).
- city: Nombre de la ciudad (cadena de caracteres, máximo 255 caracteres).
- img\_url: URL de la imagen de la ciudad (cadena de caracteres, máximo 255 caracteres).
- info: Información adicional sobre la ciudad (cadena de caracteres, máximo 255 caracteres).
- state: Estado o provincia de la ciudad (cadena de caracteres, máximo 255 caracteres).

t\_country:

- id (PK): Identificador único del país (entero grande, autoincremental).
- currency\_symbol: Símbolo de la moneda del país (carácter).
- population: Población del país (entero).
- capital: Capital del país (cadena de caracteres, máximo 255 caracteres).
- country: Nombre del país (cadena de caracteres, máximo 255 caracteres).
- country\_code: Código del país (cadena de caracteres, máximo 255 caracteres).
- currency\_code: Código de la moneda del país (cadena de caracteres, máximo 255 caracteres).
- img\_url: URL de la imagen representativa del país (cadena de caracteres, máximo 255 caracteres).
- info: Información adicional sobre el país (cadena de caracteres, máximo 255 caracteres).
- language\_code: Código del idioma principal del país (cadena de caracteres, máximo 255 caracteres).

t\_notification:

- id (PK): Identificador único de la notificación (entero grande, autoincremental).
- time\_ago: Tiempo transcurrido desde la notificación (cadena de caracteres, máximo 255 caracteres).
- title: Título de la notificación (cadena de caracteres, máximo 255 caracteres).

t\_notification\_user\_notified:

- notification\_id (FK): ID de la notificación (entero grande).
- user\_notified\_username (FK): Nombre de usuario del usuario notificado (cadena de caracteres, máximo 20 caracteres).

t\_user:

- username (PK): Nombre de usuario único (cadena de caracteres, máximo 20 caracteres).
- active: Indica si el usuario está activo o no (booleano).
- email: Correo electrónico del usuario (cadena de caracteres, máximo 50 caracteres).
- phone: Número de teléfono del usuario (cadena de caracteres, máximo 50 caracteres).
- password: Contraseña del usuario (cadena de caracteres, máximo 100 caracteres).
- bio: Biografía del usuario (cadena de caracteres, máximo 256 caracteres).

- img\_url: URL de la imagen de perfil del usuario (cadena de caracteres, máximo 255 caracteres).

t\_user\_role:

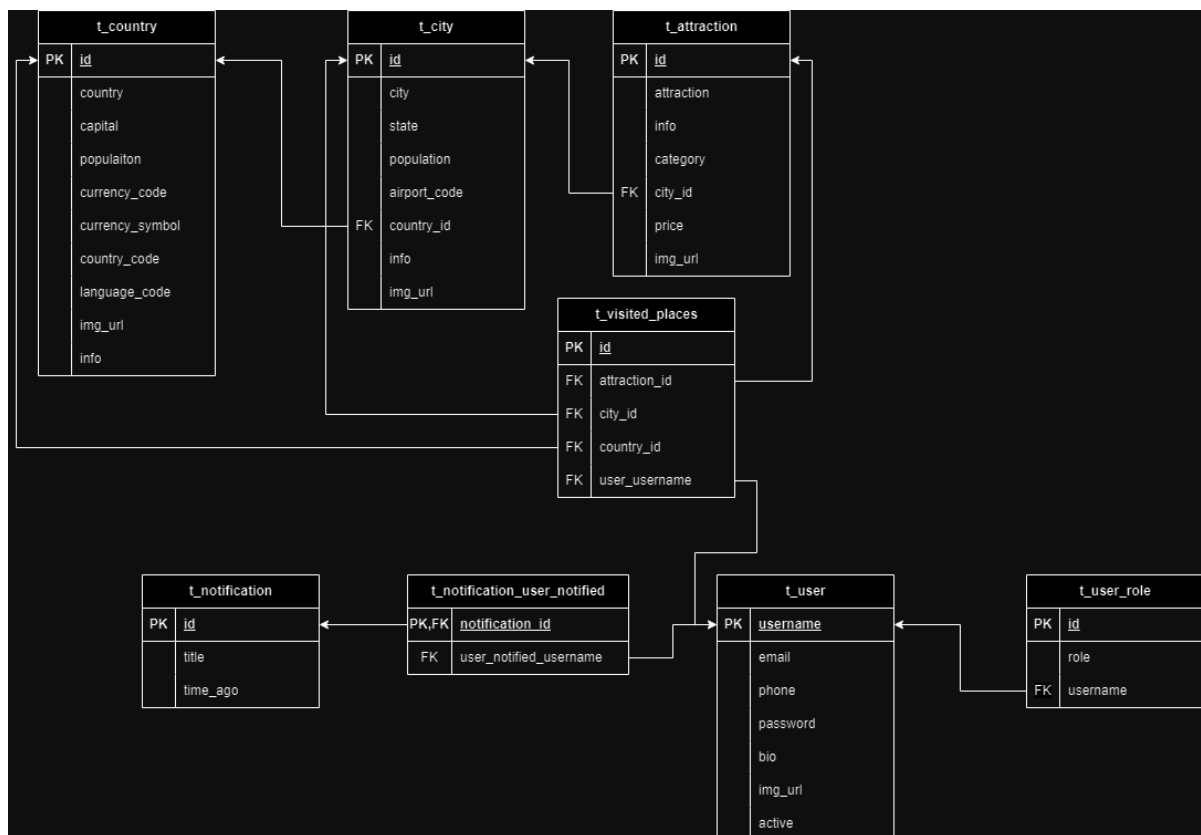
- id (PK): Identificador único de la relación usuario-rol (entero grande, autoincremental).
- role: Rol del usuario (cadena de caracteres, máximo 20 caracteres).
- username (FK): Nombre de usuario al que se le asigna el rol (cadena de caracteres, máximo 20 caracteres).

t\_visited\_places:

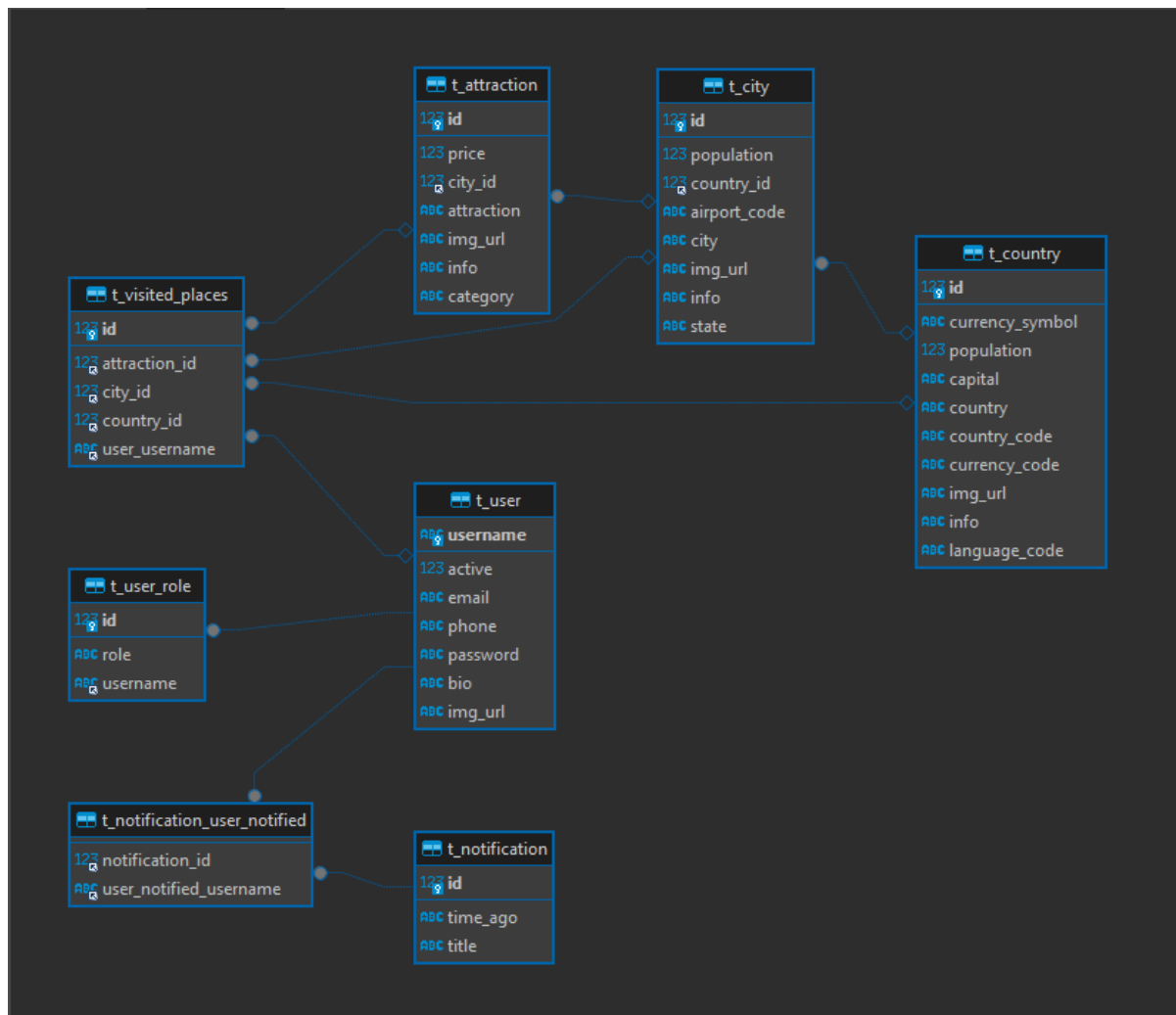
- id (PK): Identificador único del lugar visitado (entero grande, autoincremental).
- attraction\_id (FK): ID de la atracción visitada (entero grande).
- city\_id (FK): ID de la ciudad visitada (entero grande).
- country\_id (FK): ID del país visitado (entero grande).
- user\_username (FK): Nombre de usuario que visitó el lugar (cadena de caracteres, máximo 20 caracteres).

## ESQUEMA

Conceptual (<https://app.diagrams.net/>)



## Final




## Prototipo en Figma o similar

Enlace al prototipo en Figma:

<https://www.figma.com/design/Sd5bJuVArkSw0eleSW9d2T/Proyecto-Final-Grado?node-id=0%3A1&t=Ai0y5OiO6B5zIDzC-1>

Enlace al pdf del prototipo:

 Proyecto Final Grado.pdf

# Definición API REST publicación servicios

Enlace a la documentación de la API REST desde postman:

<https://documenter.getpostman.com/view/31759288/2sA3JNa11V>

Enlace al JSON de la exportacion de postman:

[Proyecto Final PMS.postman\\_collection.json](#)

Enlace a la documentación Javadoc de la API REST:

<https://github.com/PabloMerinas/Proyecto-Final-Pablo-Merinas-Soto/tree/main/docs/Javadoc>

## Manual de despliegue

Enlace al manual de despliegue completo:

 [Manual de despliegue](#)

## Postmortem del Proyecto

### Resumen del Proyecto

Este proyecto fue una mejora significativa del proyecto realizado previamente para la dual. Inicialmente, la idea principal era tomar el proyecto existente y mejorarlo un poco añadiendo seguridad. Sin embargo, al decidir separar la lógica del backend y el frontend y enfocarme en aprender React, terminé empezando todo desde cero. A medida que iba aprendiendo nuevas tecnologías y aplicando mejoras, el proyecto evolucionó de manera considerable, resultando en una versión mucho más avanzada y funcional en términos de tecnologías usadas y funcionalidades ofrecidas.

### Lo que Funcionó Bien

1. Aprendizaje de React: Aprendí bastante sobre React y me sorprendió la facilidad y las posibilidades que ofrece. Esta experiencia me ha inspirado a seguir explorando esta tecnología.



2. Programación de la API: La programación de la API fue bastante sencilla en general, gracias a mi dominio previo en este ámbito.
3. Despliegue en AWS: El despliegue en AWS fue más sencillo de lo esperado y dio muy buenos resultados, mejorando significativamente la accesibilidad y la disponibilidad de la aplicación.

## Lo que No Funcionó Bien

1. Gestión de Imágenes de Perfil: Tuve problemas al guardar y recuperar la imagen del perfil. Al principio, las imágenes se guardaban dentro del propio proyecto, lo cual causaba problemas al empaquetar el jar. La solución fue cambiar la ruta de almacenamiento a una fuera del proyecto.
2. Manejo del Usuario Activo: Inicialmente, gestioné el usuario activo mediante cookies, lo cual no era eficiente. Luego cambié a un global context de React, pero esto causaba problemas al recargar la página. Finalmente, modifiqué el global context para guardar también en las cookies, solucionando el problema pero siendo un proceso complejo y problemático.
3. Pruebas de Hosting: Probé muchos hosting y sitios de alojamiento sin éxito, hasta que me enfoqué en AWS, donde logré desplegar la aplicación completamente.
4. Gestión de Estados en React: Al no tener experiencia previa con React, tuve varios problemas con la gestión de estados que me llevaron semanas resolver. Sin embargo, esto resultó en un aprendizaje profundo y valioso.

## Conclusiones

1. Aprendizaje Continuo: Este proyecto ha sido una experiencia de aprendizaje intensiva, especialmente en React. Me ha motivado a seguir aprendiendo y explorando nuevas tecnologías para realizar proyectos más interesantes y avanzados.
2. Mejora en AWS: He mejorado considerablemente en el uso de AWS y he aprendido mucho sobre el despliegue de aplicaciones en esta plataforma.
3. Comprensión de Frontend: A lo largo del desarrollo del frontend, he mejorado mi comprensión sobre la renderización de componentes y la gestión de estados en React, lo cual se refleja en la calidad creciente del código.

En resumen, el proyecto no solo ha resultado en una aplicación funcional y mejorada, sino que también ha sido un proceso enriquecedor que ha contribuido significativamente a mi crecimiento profesional y técnico.

## Enlace al repositorio GitHub con el código fuente completo

<https://github.com/PabloMerinas/Proyecto-Final-Pablo-Merinas-Soto>