Reporte

Laravel - PostgreSQL

WHATSAPP sekarang

Materia: Base de Datos para aplicaciones.

WHATSAPP sekarang

Docente: MDIS. Eliezer Samuel Castillo Ruiz

WHATSAPP sekarang

Equipo: TSU.Eric Aguilar, TSU.Miguel López, TSU.Said Ricardes y TSU.Diana Vasquez

UT CHETUMAL

INDICE

INDICE DE FIGURAS

Introducción

El presente documento tiene como objetivo describir el proceso de creación y configuración de un proyecto en Laravel (versión 11) utilizando PostgreSQL como sistema gestor de bases de datos. Laravel es un framework de PHP ampliamente utilizado por su facilidad de uso y su robustez para desarrollar aplicaciones web modernas. Por otro lado, PostgreSQL, conocido por su potencia y características avanzadas, se ha integrado en este proyecto como el sistema preferido para la administración de datos.

La integración de Laravel con PostgreSQL permite aprovechar características como esquemas avanzados, manejo de transacciones y alto rendimiento en consultas complejas, lo que hace de esta combinación una elección ideal para proyectos exigentes. A continuación, se detallan los pasos necesarios para

configurar y poner en marcha este proyecto, junto con ejemplos prácticos y resultados obtenidos.

Creación del Proyecto Laravel

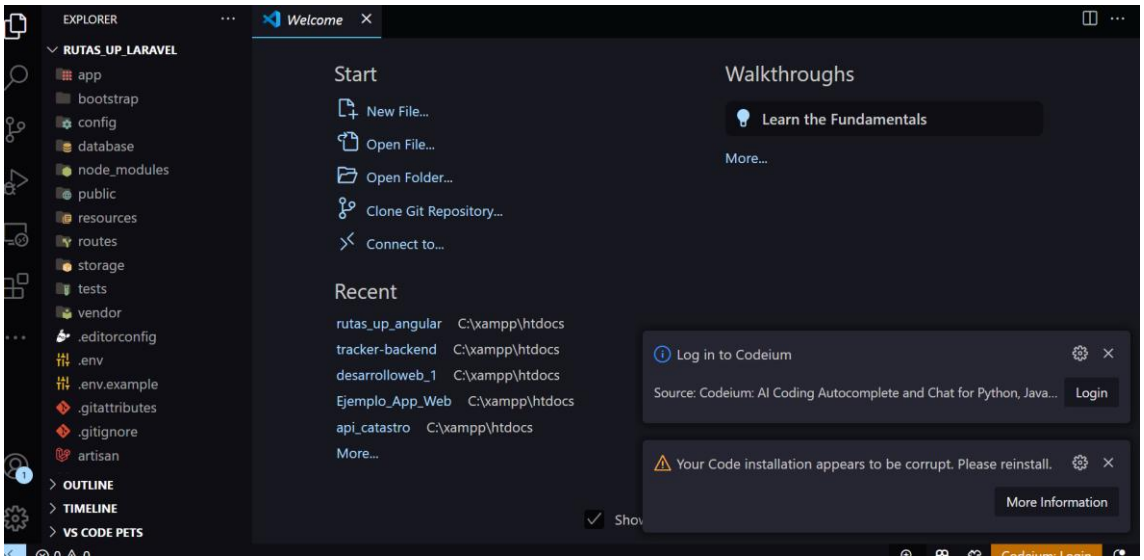
Primero se tiene que crear un proyecto de nuevo de Laravel en su versión 11, llamada rutas_up_laravel.

Figura 1. Creación de Proyecto RutasUp

Fuente: Miguel López (2025).

Luego de haber instalado todas las dependencias tendremos nuestro proyecto listo para empezar a desarrollar.

Figura 2. Vista del proyecto RutasUp



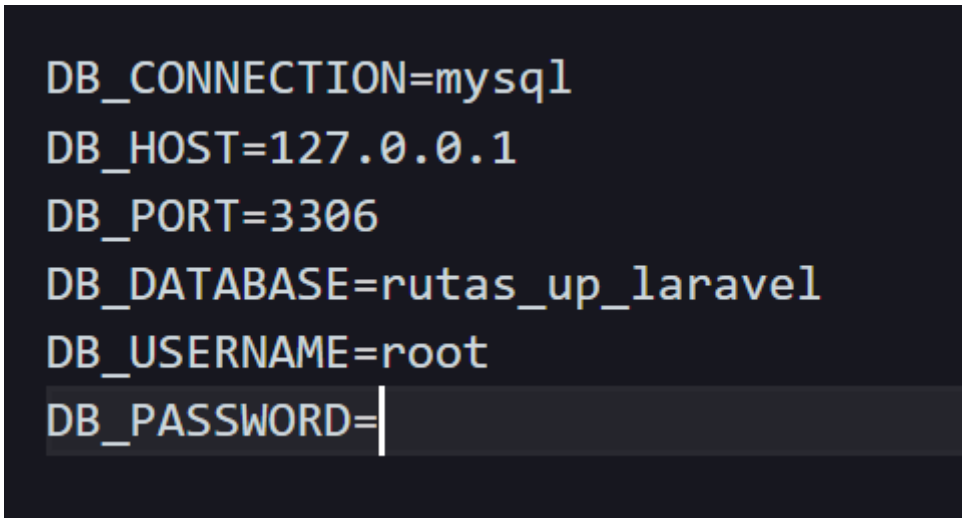
Fuente: Miguel López (2025).

Configuración de PostgreSQL como Gestor de Base de Datos

Para cambiar el sistema de gestor de base de datos de MySQL se sigue estos pasos:

Modificar el archivo. env, por default vamos a encontrar esta configuración por default:

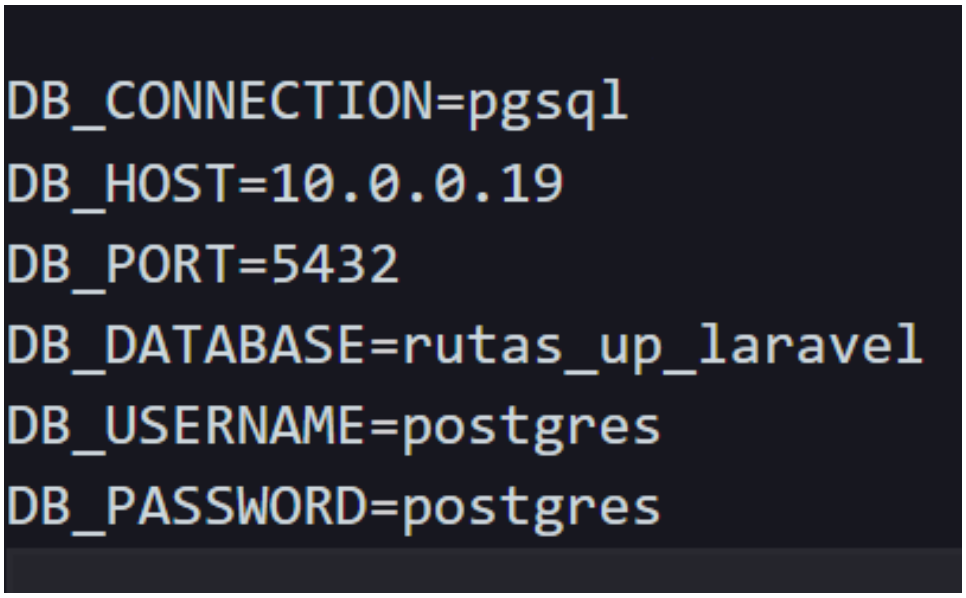
Figura 3. Modificación del .env



Fuente: Miguel López (2025).

Y se le tiene que configurar para PostgreSQL:

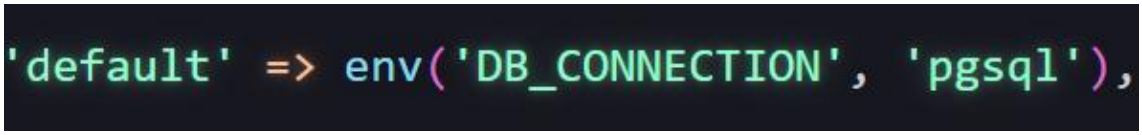
Figura 4. ENV configurado



Fuente: Miguel López (2025).

Luego nos vamos al archivo de database.php dentro de la carpeta config y ponemos como conexión por defecto pgsql y agregamos la configuración para PostgreSQL.

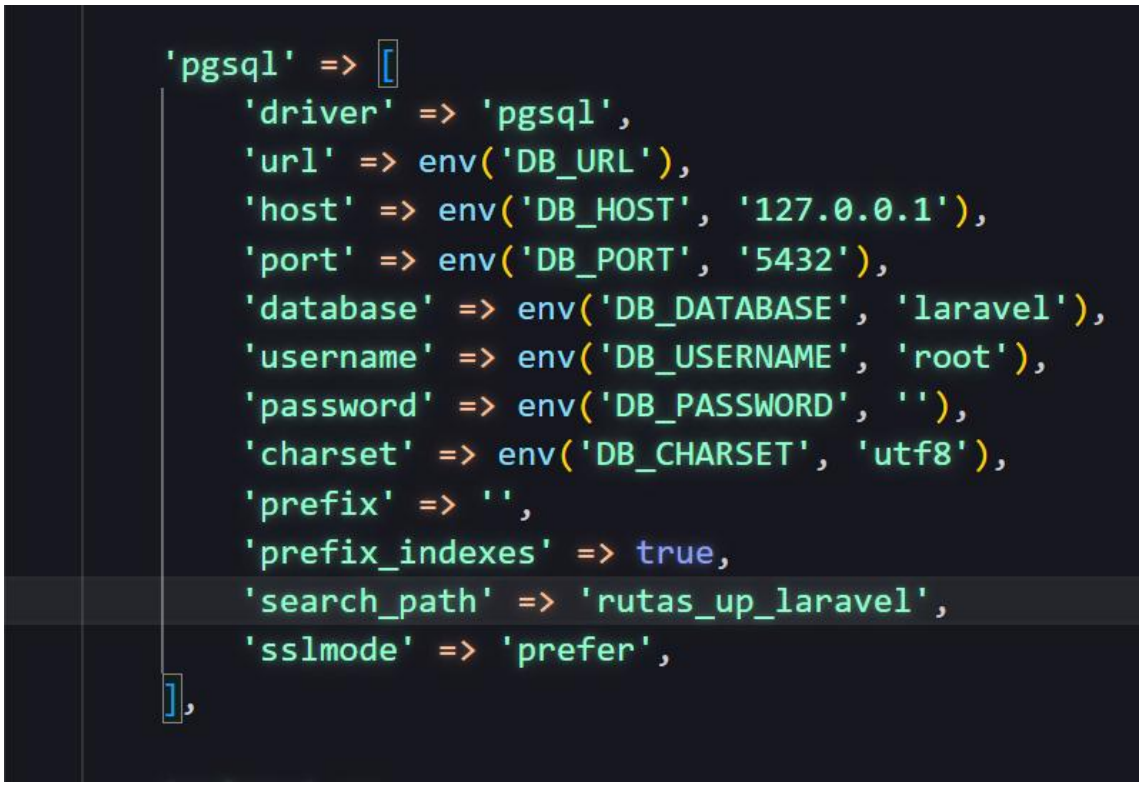
Figura 5. Conexión a Postgres.



Fuente: Miguel López (2025).

En search_path, se especifica el esquema a trabajar:

Figura 6. SCHEMA de Postgres.



Fuente: Miguel López (2025).

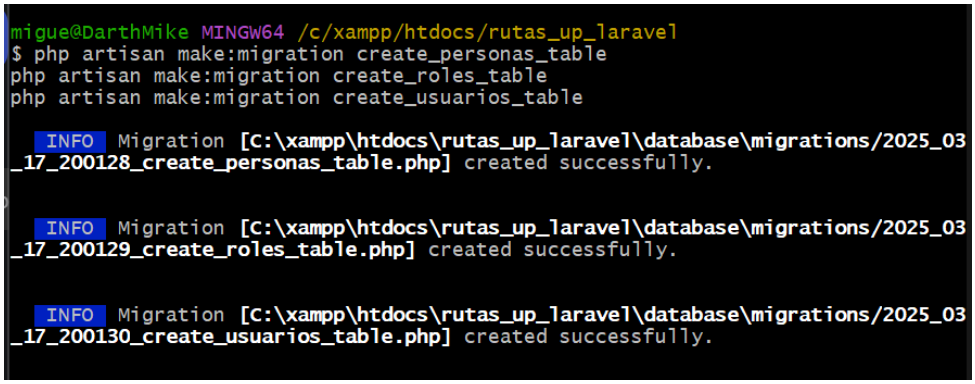
Migraciones y Tablas

Probando la configuración mediante migraciones.

Se crearon tres migraciones:

- Tabla Personas: Contiene información básica de las personas, como nombres, correos electrónicos y teléfonos.
- Tabla Roles: Define los diferentes roles que pueden tener los usuarios.
- Tabla Usuarios: Almacena credenciales de acceso, asociando cada usuario con una persona y un rol.

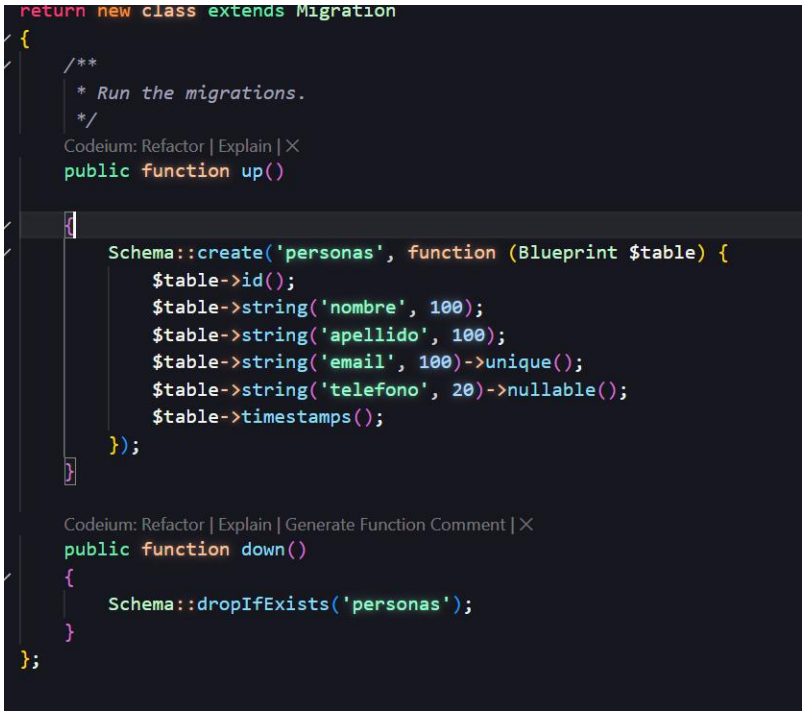
Figura 7. Migraciones.



Fuente: Miguel López (2025).

Configurando personas.

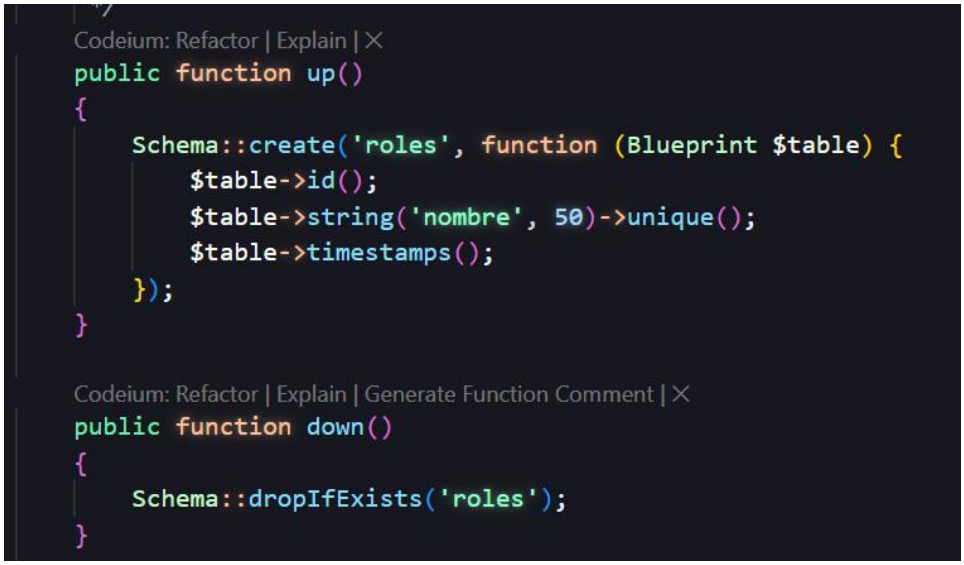
Figura 8. Configuraciones de la migración de Personas.



Fuente: Miguel López (2025).

Configurando Roles

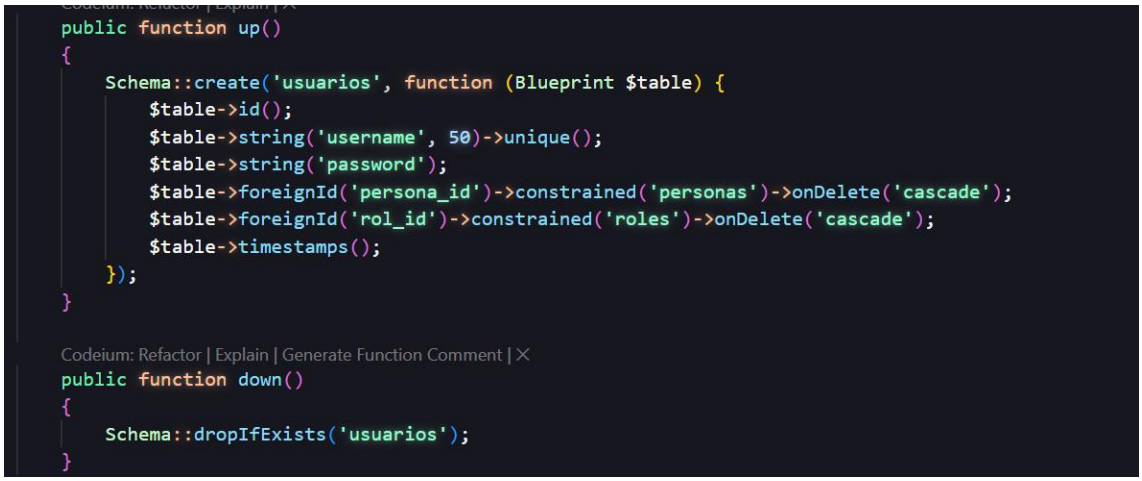
Figura 9. Configuraciones de la migración de Roles.



Fuente: Miguel López (2025).

Configurando Usuarios.

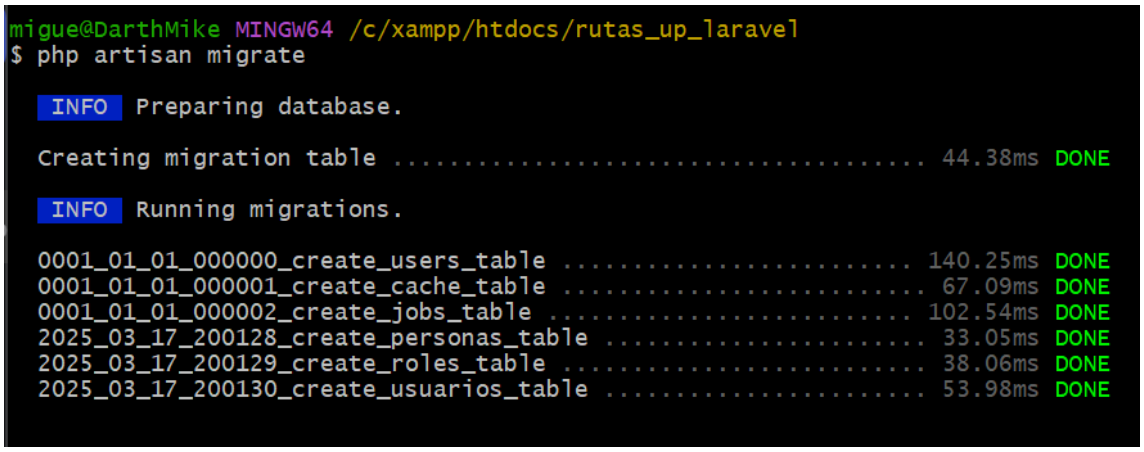
Figura 10. Configuraciones de la migración de Usuarios.



Fuente: Miguel López (2025).

Una vez configuradas las migraciones, se ejecutaron con el siguiente comando: php artisan migrate

Figura 11. Migración Completa.



Fuente: Miguel López (2025).

El sistema preparó la base de datos y creó automáticamente las tablas especificadas.

Verificación y Validación

La estructura de las tablas generadas se verificó utilizando `pgAdmin`, una herramienta gráfica para PostgreSQL. En el esquema de la base de datos, se validaron las tablas `personas`, `roles` y `usuarios`, junto con las restricciones y claves foráneas configuradas.

Verificar en el pgAdmin.

Figura 12. Primera Sección de pgAdmin.

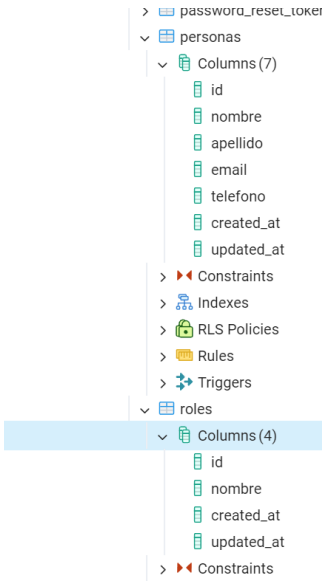
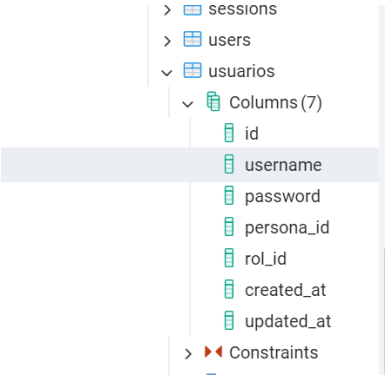


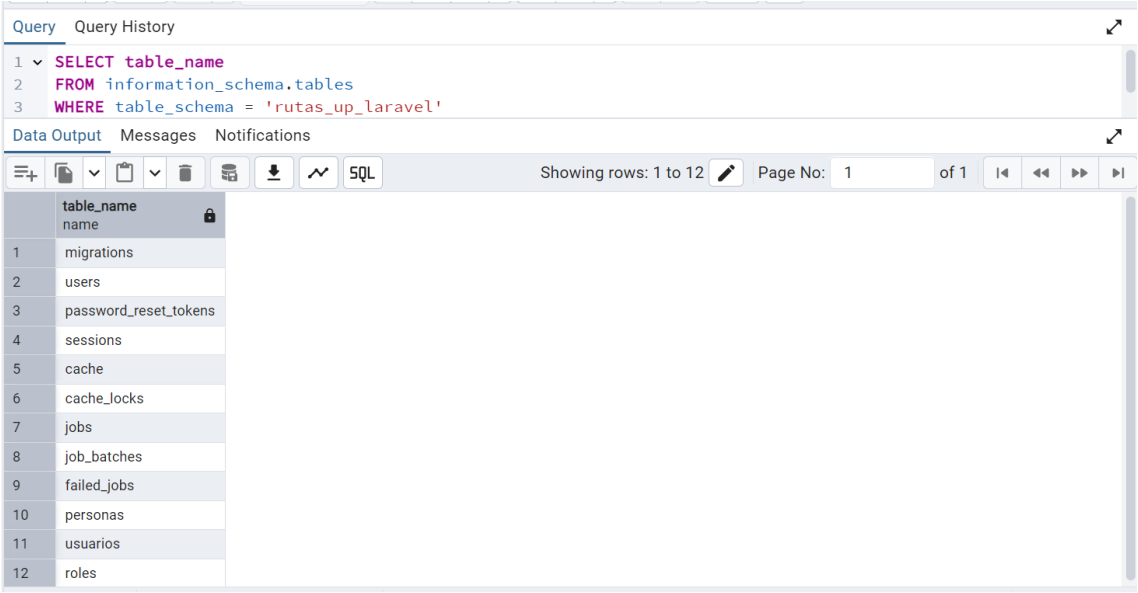
Figura 13. Segunda Sección de pgAdmin.



Fuente: Miguel López (2025).

En pgAdmin, verificar la estructura de las tablas y datos de migración. Consultar las tablas creadas:

Figura 14. Verficación Final de Postgres con Laravel.



Fuente: Miguel López (2025).

Conclusión

La integración de Laravel con PostgreSQL proporciona una solución potente y flexible para el desarrollo de aplicaciones web modernas.

Este proyecto no solo destaca la importancia de una buena configuración inicial, sino también de la capacidad de adaptabilidad de las tecnologías utilizadas. PostgreSQL permite un manejo robusto y seguro de la información, mientras que Laravel ofrece una experiencia de desarrollo fluida y eficiente.