

 Universidad Francisco de Paula Santander Ocaña - Colombia Virgilio W. Rodríguez	UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA			
	FORMATO HOJA DE RESUMEN PARA TRABAJO DE GRADO	Documento F-AC-DBL-007	Código 08-07-2021	Fecha B
	Dependencia DIVISIÓN DE BIBLIOTECA	Aprobado SUBDIRECTOR ACADEMICO	Pág. 1(1)	

RESUMEN – TRABAJO DE GRADO

AUTORES	Sergio Andrés Ortiz Ibáñez
FACULTAD	Ingenierías
PLAN DE ESTUDIOS	Ingeniería de sistemas
DIRECTOR	Javier Esneider Ibáñez Castilla
TÍTULO DE LA TESIS	Migración del framework laravel a la versión actualmente estable, de manera progresiva siguiendo la documentación oficial, para el aplicativo “Gestión v1.0” de la entidad FGA Fondo de Garantías
TITULO EN INGLES	Migration of the Laravel framework to the currently stable version, progressively following the official documentation, for the “Management v1.0” application of the FGA Fondo de Garantías entity
RESUMEN (70 palabras)	
Este trabajo de grado bajo la modalidad pasantía se enfoca en el proceso de migración del framework Laravel con el que se encuentra construido el aplicativo web "Gestión v1" de FGA Fondo de Garantías. Llevando a cabo un análisis inicial para identificar aspectos cruciales y trazar una ruta que permita efectuar la migración conservando las funcionalidades y garantizando la seguridad del sistema y, además proporcionar documentación detallada del proceso.	
RESUMEN EN INGLES	
This degree work under the internship modality focuses on the migration process of the Laravel framework with which the FGA "Management v1" web application is built Guarantee Fund. Carrying out an initial analysis to identify crucial aspects and trace a route that allows the migration to be carried out while preserving the functionalities and guaranteeing the security of the system and also provide detailed documentation of the process.	
PALABRAS CLAVES	Migración, Framework, Laravel, PHP, Dependencias, Programación, optimización funcionalidades
PALABRAS CLAVES EN INGLES	Migration, Framework, Laravel, PHP, Dependencies, Programming, functionality optimization
CARACTERÍSTICAS	
PÁGINAS: 119	PLANOS: ILUSTRACIONES: 100 CD-ROM:



Vía Acolsure, Sede el Algodonal, Ocaña, Colombia - Código postal: 546552
 Línea gratuita nacional: 01 8000 121 022 - PBX: (+57) (7) 569 00 88
 atencionalciudadano@ufpso.edu.co - www.ufpso.edu.co

Migración del framework laravel a la versión actualmente estable, de manera progresiva siguiendo la documentación oficial, para el aplicativo “Gestión v1.0” de la entidad FGA Fondo de Garantías.

Sergio Andrés Ortiz Ibáñez

Facultad de Ingenierías, Universidad Francisco de Paula Santander Ocaña

Ingeniería de Sistemas

Ing. Javier Esneider Ibáñez Castilla

Junio de 2024

Índice.

Índice.....	2
Resumen.....	5
Introducción.	6
1. Migración del framework laravel a la versión actualmente estable, de manera progresiva siguiendo la documentación oficial, para el aplicativo “Gestión v1.0” de la entidad FGA Fondo de Garantías.	7
1.1 Descripción breve de la empresa.	7
1.1.1 Misión.	7
1.1.2 Visión.....	8
1.1.3 Objetivos de la empresa.	8
1.1.4 Descripción de la estructura organizacional.	9
1.1.5 Descripción de la dependencia y/o proyecto que fue asignado.	9
1.2 Diagnóstico inicial de la dependencia asignada.....	10
1.2.1 Planteamiento del problema.....	11
1.3 Objetivos de la pasantía.	14
1.3.1 General.....	14
1.3.2 Específicos.	14
1.4 Descripción de las actividades a desarrollar en la misma	15
1.5 Cronograma de actividades.....	16

2. Enfoques referenciales.....	17
2.1 Enfoque conceptual.....	17
2.2 Enfoque legal.....	21
3. Informe de cumplimiento de trabajo.....	25
3.1 Realizar un diagnóstico del estado del aplicativo web.....	25
3.1.1 Análisis de dependencias.....	25
3.1.2 Análisis de funcionalidades.....	28
3.2 Diseñar un plan para la migración del framework que usa el aplicativo considerando posibles conflictos.....	32
3.2.1 Plan de migración.....	32
3.2.2 Configuración de entornos de desarrollo.....	35
3.2.3 Control de versiones.....	47
3.2.4 Mitigación de incompatibilidades.....	50
3.3 Efectuar la migración, garantizando el funcionamiento y mitigando cualquier impacto en la funcionalidad.....	54
3.3.1 Migración del framework.....	54
3.3.2 Resolución de errores.....	65
3.3.3 Optimización de funcionalidades.....	72
3.3.4 Cierre de vulnerabilidades.....	90
3.3.5 Desarrollo de pruebas unitarias e integración.....	92

3.4 Documentar a detalle el proceso de migración y pruebas.....	106
3.4.1 Documentación general.	106
4. Diagnóstico final.....	109
5. Conclusiones.....	110
6. Recomendaciones.	111
Referencias.....	113

Resumen.

Este trabajo de grado bajo la modalidad pasantía se enfoca en el proceso de migración del framework Laravel con el que se encuentra construido el aplicativo web "Gestión v1" de FGA Fondo de Garantías. Donde se lleva a cabo un análisis inicial para identificar aspectos cruciales y trazar una ruta que permita efectuar la migración conservando las funcionalidades y garantizando la seguridad del sistema y, además proporcionar documentación detallada del proceso y aspectos relevantes.

Introducción.

El presente trabajo de grado tiene como propósito principal la migración progresiva del framework Laravel desde su versión 5.8 a la versión estable 10.40, siguiendo las directrices de la documentación oficial del framework. Este proceso se lleva a cabo en el contexto del aplicativo "Gestión V1" de la empresa FGA Fondo de Garantías. La necesidad de esta actualización surge debido a los avances tecnológicos y mejoras de seguridad y rendimiento que ofrece la nueva versión de Laravel.

Para alcanzar este objetivo, se han establecido una serie de objetivos específicos que guiarán el desarrollo del proyecto. En primer lugar, se realizará un diagnóstico exhaustivo del estado inicial del aplicativo web para identificar las características y posibles limitaciones actuales. Posteriormente, se diseñará un plan detallado para la migración del framework, considerando y previendo posibles conflictos que pudieran surgir durante el proceso. A continuación, se llevará a cabo la migración misma, garantizando que el aplicativo mantenga su funcionalidad operativa y que cualquier impacto potencial sea mitigado adecuadamente. Finalmente, se documentará meticulosamente cada etapa del proceso de migración y las pruebas realizadas, asegurando que se disponga de un registro completo y comprensible de todas las actividades efectuadas.

Este trabajo no solo busca actualizar la plataforma tecnológica del aplicativo "Gestión V1", sino también proporcionar una guía detallada y estructurada que pueda servir como referencia para futuras migraciones dentro de la organización o en contextos similares.

1. Migración del framework laravel a la versión actualmente estable, de manera progresiva siguiendo la documentación oficial, para el aplicativo “Gestión v1.0” de la entidad FGA Fondo de Garantías.

1.1 Descripción breve de la empresa.

La empresa FGA Fondo de Garantías, es una sociedad de economía mixta, creada en 1997. La cual actúa como fiador institucional para respaldar los créditos de libre inversión, libranza, comercio, motos, digitales, educativos y microcrédito; a través de garantías crediticias.

Su conocimiento de más de 27 años le permite generar valor a través de un equipo humano experto, estructurando soluciones financieras innovadoras para brindar la confianza y la seguridad que necesitan las personas y los intermediarios de crédito.

Se sienten orgullosos de ser el vínculo entre las personas y los empresarios con los intermediarios de crédito. Para apoyar y fomentar el fortalecimiento empresarial, la innovación en los procesos productivos de las organizaciones, la generación de empleo estable y el crecimiento sostenible del país, mediante garantías, promoviendo el consumo sostenible y facilitando la inclusión. (*Conócenos, s/f*)

1.1.1 Misión.

Generamos valor y confianza a través de un equipo humano experto, estructurando soluciones financieras innovadoras y contribuyendo al desarrollo económico del país.

(*Conócenos, s/f*)

1.1.2 Visión.

FGA Fondo de Garantías será la compañía de garantías de crédito referente para el mercado colombiano, reconocida por su sostenibilidad, eficiencia, orientación al cliente y compromiso con sus colaboradores. (Conócenos, s/f)

1.1.3 Objetivos de la empresa.

Figura 1

Mapa estratégico de FGA Fondo de Garantías.



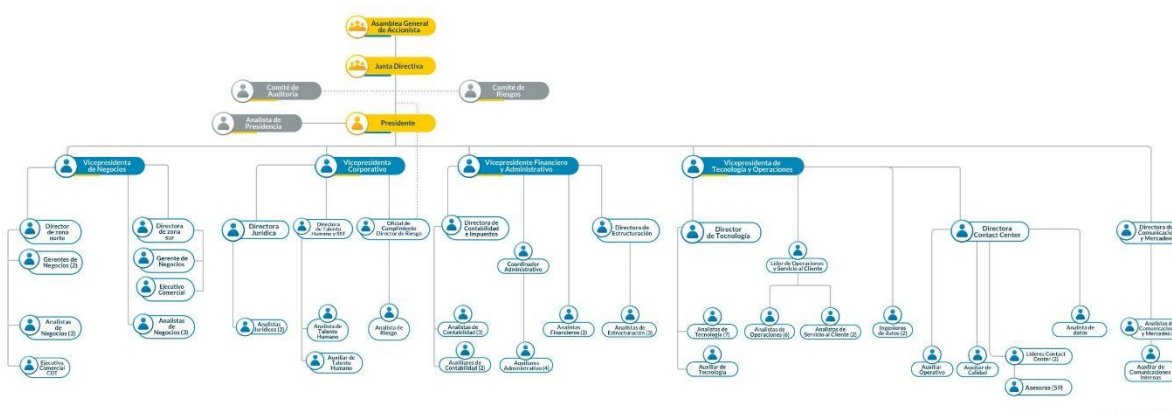
Nota: Mapa estratégico sobre los objetivos de FGA Fondo de Garantías. (FGA Fondo de Garantías | Mitigamos el riesgo de crédito, s/f)

1.1.4 Descripción de la estructura organizacional.

La empresa está constituida por un grupo de accionistas con participación público-privada. La junta directiva, compuesta por estos mismos accionistas, vela por los resultados de la compañía y define al presidente. El presidente, a su vez, tiene a su cargo cuatro vicepresidencias que se encargan de administrar las áreas más importantes: corporativa, de negocios, financiera y administrativa, tecnología y operaciones.

Figura 2

Organigrama.



Nota: Organigrama FGA Fondo de Garantías. (Organigrama FGA, s/f)

1.1.5 Descripción de la dependencia y/o proyecto que fue asignado.

En este proyecto se busca modernizar y optimizar el aplicativo web “Gestión v1.0” dispuesto para la gestión realizada con los deudores, con el fin de mantener a la vanguardia las tecnologías utilizadas para el desarrollo del aplicativo, cerrar brechas de seguridad que se generan en versiones anteriores manteniendo las mejores prácticas de desarrollo y la implementación de nuevas funcionalidades, asegurando la integridad y funcionalidad que tiene el aplicativo y familiarizarse con la arquitectura que maneja la empresa.

1.2 Diagnóstico inicial de la dependencia asignada.

Tabla 1.

Matriz DOFA y estrategias FO – DO – FA – DA.

	Fortalezas	Debilidades
Área de tecnología y operaciones	F1. Personal calificado y especializado en diversas áreas tecnológicas. F2. Infraestructura tecnológica robusta. F3. Capacidad para desarrollar soluciones tecnológicas a medida según las necesidades del cliente. F4. Experiencia en implementación exitosa de proyectos tecnológicos. F5. Buena reputación adquirida por parte de los clientes.	D1. Procesos internos ineficientes que pueden retrasar la entrega de proyectos. D2. Resistencia al cambio por parte de algunos miembros del área. D3. Poca documentación de proyectos implementados. D4. Plataformas desactualizadas en relación a las nuevas tendencias. D5. Vulnerabilidad a la fluctuación del mercado tecnológico y cambios regulatorios.
Oportunidades	Estrategias (FO)	Estrategias (DO)
O1. Mejorar la experiencia de colaboradores y clientes. O2. Aumentar la eficiencia de los proyectos tecnológicos implementados. O3. Crecimiento continuo en el negocio de las garantías. O4. Expansión de servicios hacia áreas emergentes de la tecnología.	1. Desarrollar programas de formación y capacitación. 2. Expandir la cartera de servicios tecnológicos. 3. Utilizar la experiencia del equipo para gestionar riesgos. 4. Fomentar la cultura de innovación.	1. Revisar y optimizar procesos internos. 2. Invertir en actualización tecnológica. 3. Promover una cultura de cambio. 4. Mejorar la gestión del conocimiento. 5. Implementar programas de formación interna para que el equipo adquiera nuevas habilidades.
Amenazas	Estrategias (FA)	Estrategias (DA)
A1. Competencia creciente de otros intermediarios financieros. A2. Cambios regulatorios en el sector financiero y tecnológico. A3. Riesgos de seguridad respecto al manejo de información sensible.	1. Fortalecer la seguridad de la información. 2. Diversificar la cartera de servicios. 3. Establecer alianzas estratégicas 4. Diversificar geográficamente la base de clientes.	1. Mejorar la comunicación interna y la gestión del cambio. 2. Documentar y estandarizar procesos. 3. Mejorar la monitorización y análisis de riesgos. 4. Implementar un plan de continuidad del negocio.

A4. Cambios en las condiciones macroeconómicas.

Nota: Elaboración propia.

1.2.1 Planteamiento del problema.

El uso de una versión antigua de una herramienta de software, como en el caso de la empresa "FGA Fondo de Garantías", conlleva diversas implicaciones que pueden afectar significativamente su operatividad y competitividad en el mercado.

La inestabilidad y vulnerabilidades inherentes a versiones desactualizadas representan uno de los mayores riesgos. Estas pueden resultar en un mayor riesgo de fallos, ataques cibernéticos y problemas de rendimiento, comprometiendo así la integridad y seguridad de los datos de la empresa y sus clientes. (D'Ambrosio et al., 2023; Jamshidi et al., 2014)

La falta de soporte técnico es otro aspecto crucial. Al no tener acceso a correcciones de errores críticos o nuevas funcionalidades proporcionadas en versiones más recientes, la empresa se ve limitada en su capacidad para mantener su aplicativo web funcionando de manera óptima y al día con las demandas del mercado y las expectativas de los usuarios. (Gandikota et al., 2023; Koteska & Mishev, 2014)

Las limitaciones de rendimiento también son una preocupación importante. Una versión antigua de un framework como Laravel puede resultar en una disminución en la velocidad de carga del aplicativo web y una menor capacidad para manejar un mayor volumen de tráfico, lo que impacta negativamente la experiencia del usuario y la eficiencia operativa de la empresa. (Gandikota et al., 2023; Mahmood & Ashour, 2020)

Además, el uso de versiones desactualizadas puede llevar a una pérdida significativa de competitividad. En un entorno empresarial en constante evolución, quedarse rezagado con tecnología obsoleta puede significar una desventaja frente a otras empresas que utilizan tecnologías a la vanguardia o están desarrolladas sobre nuevas tendencias y pueden ofrecer funcionalidades más avanzadas y una experiencia de usuario mejorada. (Gandikota et al., 2023)

Los costos asociados con el mantenimiento de versiones antiguas también deben tenerse en cuenta. Los gastos adicionales para corregir errores, adquirir nuevas licencias de software o compensar interrupciones del servicio pueden sumarse rápidamente, impactando negativamente en el presupuesto y los recursos de la empresa. (Koteska & Mishev, 2014)

Por último, el daño a la reputación es una preocupación válida. El deterioro de la imagen y la confianza de clientes y socios comerciales debido a problemas de seguridad, rendimiento o confiabilidad puede tener consecuencias a largo plazo para la empresa, afectando su capacidad para retener clientes existentes y atraer nuevos negocios. (Gholami et al., 2016)

La empresa se enfrenta al desafío de migrar su aplicativo web desde la versión 5.8 de Laravel a la versión 10.x de manera progresiva, aprovechando las nuevas funcionalidades y mejoras de rendimiento presentes en las últimas versiones del framework. (Chauhan et al., 2023)

Para abordar este proyecto, la empresa cuenta con valiosos recursos, como un servidor de pruebas para realizar implementaciones y pruebas exhaustivas, herramientas de control de versiones que facilitan la gestión y seguimiento de cambios en el código, y acceso a la documentación oficial de Laravel para guiar el proceso de migración. (Lalitha Devi et al., 2023; Niranjana et al., 2023)

Sin embargo, se presentan posibles desafíos durante la migración que deben abordarse adecuadamente, como lo son:

Errores durante la migración donde la posibilidad de que surjan errores durante la migración del aplicativo a las nuevas versiones de Laravel. Estos errores pueden variar desde incompatibilidades de código hasta problemas en la configuración de la aplicación. (McCall & Kölling, 2019; Sobhani et al., 2018)

Incompatibilidades con las dependencias del aplicativo, las cuáles podrían no ser compatibles con las nuevas versiones de Laravel. Esto podría generar conflictos que afecten la funcionalidad y estabilidad del sistema. (Kiesel et al., 2020)

Es posible que algunas funcionalidades del aplicativo no se puedan migrar de manera directa a las nuevas versiones de Laravel, ya sea por cambios en la arquitectura del framework o por desuso de ciertas características. (Schmiedmayer et al., 2023)

Debido a la alta probabilidad de aparición de problemas o riesgos que afecten la transición se buscan opciones las cuales puedan ayudar a mitigar estos riesgos, y buscar implementar estrategias específicas al efectuar los cambios necesarios en la aplicación. (Volynsky et al., 2022)

Realizar pruebas para identificar y corregir cualquier error o incompatibilidad que pueda surgir durante la migración. Esto asegurará que el aplicativo funcione de manera óptima en cada versión intermedia. (Matalonga et al., 2022; Salahirad et al., 2023)

Se debe establecer una comunicación proactiva con el cliente para informar sobre los avances en la migración y compartir cualquier riesgo identificado. Esta transparencia facilitará la toma de decisiones conjuntas y permitirá ajustar la estrategia según sea necesario. (Beaubouef, 2003; Galster et al., 2022)

La empresa debe estar comprometida a abordar con responsabilidad y proactividad a los posibles desafíos que puedan surgir durante la migración de su aplicativo web a las versiones más recientes de Laravel. (Dehghani et al., 2022)

Con un enfoque basado en la experiencia, la planificación cuidadosa y la comunicación transparente, se espera que el proceso sea fluido, exitoso y brinde a los usuarios una experiencia mejorada y actualizada. (Dharmaraj & Kavitha, 2023)

1.3 Objetivos de la pasantía.

1.3.1 General.

Migrar laravel 5.8 a su versión estable 10.40 de manera progresiva, siguiendo la documentación oficial proporcionada por el framework, para el aplicativo “Gestión V1” de FGA Fondo de Garantías.

1.3.2 Específicos.

1. Realizar un diagnóstico del estado inicial del aplicativo web.
2. Diseñar un plan para la migración del framework que usa el aplicativo considerando posibles conflictos.
3. Efectuar la migración, garantizando el funcionamiento y mitigando cualquier impacto en la funcionalidad.

4. Documentar a detalle el proceso de migración y pruebas.

1.4 Descripción de las actividades a desarrollar en la misma .

Tabla 2

Descripción de actividades

Objetivo General	Objetivos Específicos	Actividades a desarrollar
Migrar el aplicativo “Gestión V1.0” de Laravel 5.8 a su versión estable 10.X, siguiendo la documentación oficial proporcionada el framework, para la entidad FGA Fondo de Garantías.	Realizar un diagnóstico del estado inicial del aplicativo web.	Análisis de dependencias.
		Análisis de funcionalidades.
	Diseñar un plan para la migración del framework que usa el aplicativo considerando posibles conflictos.	Plan de migración,
		Configuración de entornos de desarrollo.
		Control de versiones.
		Mitigación de incompatibilidades.
		Migración del framework.
	Efectuar la migración, garantizando el funcionamiento y mitigando cualquier impacto en la funcionalidad.	Resolución de errores.
		Optimización de funcionalidades.
		Cierre de vulnerabilidades.
		Desarrollo de pruebas unitarias e integración.
	Documentar a detalle el proceso de migración y pruebas.	Documentación general.

Nota: Elaboración propia

1.5 Cronograma de actividades.

El cronograma está estipulado para un total de 78 días con un horario laboral de lunes a viernes, excluyendo los días festivos.

Figura 3

Diagrama de Gantt: Calendario por actividades.

Cronograma de actividades	Mes	Febrero					Marzo					Abril					Mayo					Junio		
	Semana	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3			
Análisis de dependencias.																								
Análisis de funcionalidades.																								
Configuración de entornos de desarrollo.																								
Mitigación de incompatibilidades.																								
Control de versiones.																								
Migración del framework.																								
Resolución de errores.																								
Optimización de funcionalidades.																								
Cierre de vulnerabilidades.																								
Desarrollo de pruebas unitarias e integración.																								
Documentación general.																								

Nota: Elaboración propia.

2. Enfoques referenciales.

2.1 Enfoque conceptual.

Software Libre: Software que puede ser utilizado, modificado y distribuido libremente por cualquier persona. (*¿Qué es el Software Libre? - Proyecto GNU - Free Software Foundation, s/f*)

Código Abierto: Software cuyo código fuente está disponible para que cualquier persona pueda inspeccionarlo, modificarlo y distribuirlo. (*¿Qué es el código abierto? - Explicación del código abierto - AWS, s/f*)

Framework de Desarrollo Web: Estructura o conjunto de herramientas que facilita el desarrollo de aplicaciones web. (*Framework, s/f*)

Migración de Software: Proceso de cambiar una aplicación de software de un entorno a otro, como de una versión a otra o de un sistema operativo a otro. (*¿Qué es la migración de aplicaciones? | IBM, s/f*)

Dependencia: se refiere a la relación de necesitar un componente de software o sistema de otro para funcionar, estas dependencias en general cuentan con distintas versiones y se especificada por números y puntos. (*Administración de dependencias | Software supply chain security | Google Cloud, s/f*)

Pruebas unitarias: Es el proceso en que se prueba la unidad del código más pequeña y se realizan con el fin de garantizar la calidad del código. (*¿Qué son las pruebas de software? | IBM, s/f*)

Pruebas de integración: Es una prueba que se realiza con el fin de verificar que distintos módulos o componentes de la aplicación funcionan correctamente en conjunto. (*¿Qué son las pruebas de software? | IBM, s/f*)

Modelo (Model): Los modelos hacen referencia a una tabla de la base de datos y es utilizado para interactuar con la tabla, donde se pueden recuperar registros, insertar, actualizar o eliminar. (*Laravel - The PHP Framework For Web Artisans, s/f*)

Controlador (Controller): Los controladores es donde se agrupa toda la lógica para el manejo de solicitudes que estén relacionadas en una sola clase.

Fabricas (Factories): Es un conjunto de atributos predeterminados para cada uno de los modelos que es capaz de insertar cierta cantidad de registros en la base de datos. Con el fin de probar la aplicación o inicializar la base de datos. (*Laravel - The PHP Framework For Web Artisans, s/f*)

Afirmaciones (Asserts): Es una declaración que se compara con el resultado real de una operación con el resultado que se espera, con el fin de identificar que el código se comporte como se desea y ayudar a detectar errores. (*1. Assertions — PHPUnit 11.1 Manual, s/f*)

Repositorio: Es un lugar centralizado donde se almacena información digital que los desarrolladores utilizan con el fin de administrar cambios, realizar trabajo colaborativo, acceso remoto. (*¿Qué son los repositorios?, s/f*)

Bases de datos: Es una colección organizada de información o datos, estructurados, almacenados de forma electrónica. Esta información se organiza de forma sistemática para facilitar su recuperación y administración. (*Qué es una base de datos | Oracle Colombia, s/f*)

Proveedor de servicios en la nube: Es una empresa que ofrece recursos informáticos a través de internet, que son escalables y adquiridos bajo demanda. (*¿Qué es un proveedor de servicios en la nube?, s/f*)

Amazon web services (AWS): Es el proveedor de servicios en la nube mas adoptado, el cual ofrece más de 200 servicios.

Tecnologías Involucradas

Git: Es un sistema de control de versiones gratuito y de código abierto, la cual permite la creación, fusión y eliminación de líneas de desarrollo. (*Git, s/f*)

GitLab: Es una plataforma de software de código abierto para la gestión del ciclo de vida del desarrollo de software, que proporciona herramientas para la gestión de repositorios git, seguimiento y CI/CD (integración continua y despliegue continuo). (*The most-comprehensive AI-powered DevSecOps platform | GitLab, s/f*)

Visual Studio Code: Es un editor de código fuente gratuito y multiplataforma desarrollado por Windows con un amplio ecosistema de extensiones e importantes características como soporte para múltiples lenguajes, integración con git, personalización, entre otros. (*Documentation for Visual Studio Code, s/f*)

Vagrant: Es una herramienta para construir entornos de desarrollo que reduce el tiempo de configuración del entorno y cuenta con un flujo de trabajo simple de usar y enfocado en la automatización. (*Introduction | Vagrant | HashiCorp Developer, s/f*)

Virtualbox; Es un software de virtualizaciones para arquitecturas x86 y AMD64/Interl64, la cual permite crear y ejecutar máquinas virtuales en un solo ordenador. (*Oracle VM VirtualBox, s/f*)

MySQL: Es un sistema de gestión de bases de datos relacionales de código abierto y gratuito, que se usa para almacenar, organizar y recuperar datos. Y es calificada como una de las bases de datos mas populares. (*¿Qué es MySQL?*, s/f)

Composer: Es una herramienta para la gestión de dependencias en PHP. (*Introduction - Composer*, s/f)

NVM: es un administrador de versiones de node.js a través de la línea de comando. (*nvm-sh/nvm*, 2010/2024)

NodeJS: Es un entorno de ejecución de JavaScript multiplataforma y de código abierto. (*Node.js — Introduction to Node.js*, s/f)

Postman: Es una herramienta para el desarrollo y pruebas de API, el cual proporciona interfaz para crear, enviar y recibir solicitudes API. (*What is Postman? Postman API Platform*, s/f)

TablePlus: Es una herramienta que permite a los usuarios administrar de manera simultanea múltiples bases de datos y cuenta con una capa gratuita limitada. (*Overview | TablePlus Documentation*, 2021)

Amazon S3: Es un servicio proporcionado por AWS que permite el almacenamiento de objetos que ofrece escalabilidad, disponibilidad y seguridad. (*AWS | Almacenamiento de datos seguro en la nube (S3)*, s/f)

Laravel: Framework de desarrollo web PHP gratuito y de código abierto, con una sintaxis elegante y expresiva, basado en el patrón de arquitectura modelo-vista-controlador (MVC). (*Laravel - The PHP Framework For Web Artisans*, s/f)

PHP: Lenguaje de programación de código abierto utilizado para el desarrollo web. (*PHP: ¿Qué es PHP? - Manual*, s/f)

Packagist: Es el repositorio de paquetes central para PHP. (*Packagist*, s/f)

2.2 Enfoque legal.

Considerando diversos aspectos legales y regulatorios que pueden estar implicados en el proceso migración del framework Laravel en el aplicativo “Gestión V1” de FGA asegurando que se cumpla con los términos de uso y las restricciones legales.

Artículo 15 de la Constitución Política de Colombia.

Establece el derecho fundamental a la intimidad personal y familiar y a una buena reputación. Además, otorga a las personas el derecho a conocer, actualizar y rectificar la información recopilada sobre ellas en bancos de datos y registros públicos y privados. (*Leyes desde 1992 - Vigencia expresa y control de constitucionalidad [CONSTITUCION_POLITICA_1991]*, s/f)

Ley Estatutaria 1266 de 2008.

También conocida como Ley de Habeas Data, que desarrolla el derecho fundamental de las personas a conocer, actualizar y rectificar la información que se ha recogido sobre ellas en bancos de datos y archivos de entidades públicas y privadas, y regula el manejo de la información contenida en bases de datos personales, en especial la financiera, crediticia, comercial, de servicios y la proveniente de terceros países. (*Ley 1266 de 2008 - Gestor Normativo - Función Pública*, s/f)

Ley 1273 de 2009.

conocida como Ley de Delitos Informáticos, es una ley que modifica el Código Penal para crear nuevos delitos informáticos y establecer penas para quienes los cometan. La ley también crea medidas para proteger la infraestructura crítica del país contra ataques cibernéticos. *(Ley 1273 de 2009 - Gestor Normativo - Función Pública, s/f)*

Ley Estatutaria 1581 de 2012.

También conocida como Ley de Protección de Datos Personales que establece los principios y normas generales para la protección de datos personales. Aplica al tratamiento de datos personales realizado en territorio colombiano. *(Ley 1581 de 2012 - Gestor Normativo - Función Pública, s/f)*

Ley Estatutaria 2157 de 2021.

También conocida como Ley de Borrón y Cuenta Nueva es una ley colombiana que modifica y adiciona la Ley Estatutaria 1266 de 2008 y dicta disposiciones generales sobre el habeas data en relación con la información financiera, crediticia, comercial y de servicios. *(Ley 2157 de 2021 - Gestor Normativo - Función Pública, s/f)*

Decreto 1377 de 2013.

Es un decreto colombiano que reglamenta parcialmente la Ley 1581 de 2012, la cual establece disposiciones generales para la protección de datos personales

Circular única de la Superintendencia de Industria y Comercio – Título V

Establece un conjunto de principios, requisitos y procedimientos que rigen la recolección, procesamiento, almacenamiento, transmisión y eliminación de datos personales en Colombia.

(Decreto 1377 de 2013 - Gestor Normativo - Función Pública, s/f)

Ley 23 de 1982

También conocida como Ley de Derecho de Autor, regula los derechos morales y patrimoniales que la ley concede a los autores sobre sus obras literarias, artísticas y científicas, estén publicadas o inéditas. Esta ley tiene como objetivo proteger la creación intelectual y fomentar el desarrollo. *(Ley 23 de 1982 - Gestor Normativo - Función Pública, s/f, p. 23)*

Ley 1341 de 2009

También conocida como Ley TIC, constituye un marco legal integral que establece los principios y lineamientos para el desarrollo y la organización del sector de las Tecnologías de la Información y las Comunicaciones (TIC) en Colombia. Su objetivo fundamental es promover el crecimiento y la modernización del sector TIC, garantizar el acceso universal a internet y proteger los derechos de los usuarios. *(Ley 1341 de 2009 - Gestor Normativo - Función Pública, s/f)*

Ley 527 de 1999

También conocida como ley de Comercio Electrónico, es una legislación clave que establece el marco legal para las transacciones electrónicas en el país. Esta ley reconoce la validez jurídica y la eficacia probatoria de los mensajes de datos, estableciendo así un marco

legal para el comercio electrónico y la firma electrónica. (*Ley 527 de 1999 - Gestor Normativo - Función Pública, s/f*)

Decreto 1242 de 2021.

Este decreto modifica la estructura del Fondo Nacional de Garantías S.A. (FNG) y determina las funciones de sus dependencias. Establece las funciones de sus diferentes áreas y los procedimientos para la administración de los recursos del fondo. (*Decreto 1242 de 2021 - Gestor Normativo - Función Pública, s/f*)

3. Informe de cumplimiento de trabajo.

3.1 Realizar un diagnóstico del estado del aplicativo web.

3.1.1 Análisis de dependencias.

En esta etapa del proyecto, se llevó a cabo un análisis exhaustivo de todas las dependencias que posee la aplicación en su fase inicial (Laravel 5.8), con el propósito de comprender su estructura y funcionamiento.

Figura 4

Dependencias iniciales del aplicativo “Gestión V1”

Dependencia	Laravel 5.8
php	^7.1.3
barryvdh/laravel-snappy	^0.4.6
doctrine/dbal	^2.10
fideloper/proxy	^4.0
guzzlehttp/guzzle	^6.4
h4cc/wkhtmltoimage-amd64	^0.12.x
h4cc/wkhtmltopdf-amd64	^0.12.x
laravel/framework	5.8.*
laravel/tinker	^1.0
league/flysystem-aws-s3-v3	^1.0
maatwebsite/Excel	^3.1
mpociot/laravel-apidoc-generator	^3.10
spatie/laravel-cors	^1.3
squizlabs/php_codesniffer	*
tymon/jwt-auth	~1.0
beyondcode/laravel-dump-server	^1.0
filp/whoops	^2.0
fzaninotto/faker	^1.4
mockery/mockery	^1.0
nunomaduro/collision	^3.0
phpunit/phpunit	^7.5
nesbot/carbon	^2.71.10 ^3.0.0

Nota: Elaboración propia

- **Php.** Especifica la versión mínima de PHP necesaria para que el proyecto funcione correctamente.

- **barryvdh/laravel-snappy:** Es un paquete que permite generar fácilmente archivos PDF e imágenes a partir de HTML.
- **doctrine/dbal:** Es un componente de la biblioteca Object-Relational Mapping (ORM) Doctrine para PHP. Proporciona una capa de abstracción de base de datos que permite interactuar con diferentes bases de datos.
- **fideloper/proxy:** Este paquete ofrece una manera conveniente para configurar proxies confiables.
- **guzzlehttp/guzzle:** Es una biblioteca que proporciona una interfaz fluida y consistente para realizar solicitudes HTTP.
- **h4cc/wkhtmltoimage-amd64:** Es una herramienta de línea de comandos que convierte páginas HTML en imágenes.
- **h4cc/wkhtmltopdf-amd64:** Es una herramienta de línea de comandos que permite convertir páginas HTML en archivos PDF.
- **laravel/framework:** Constituye el núcleo del framework Laravel. Proporciona una base sólida y una colección de herramientas para simplificar y agilizar el proceso de desarrollo.
- **laravel/tinker:** Proporciona una herramienta de línea de comandos interactiva que te permite interactuar con tu aplicación Laravel directamente desde la terminal.
- **league/flysystem-aws-s3-v3:** Es una herramienta poderosa para interactuar con el almacenamiento Amazon S3 desde aplicaciones PHP.
- **mattwebsite/Excel:** permite trabajar con archivos de hojas de cálculo Excel de manera sencilla y eficiente. Esta dependencia facilita la creación, lectura, modificación y exportación de archivos Excel en aplicaciones PHP.

- **mpociot/laravel-apidoc-generator:** Es una herramienta para generar automáticamente documentación API a partir de rutas de Laravel/Lumen/Dingo.
- **spatie/laravel-cors:** Es un paquete Laravel que facilita la configuración y gestión de la política CORS.
- **squizlabs/php_codesniffer:** Es una herramienta para el análisis estático y la verificación de la calidad del código en proyectos.
- **tymon/jwt-auth:** Es una biblioteca que se usa para implementar la autenticación basada en JWT (JSON Web Tokens) en aplicaciones.
- **beyondcode/laravel-dump-server:** Es una herramienta que mejora la funcionalidad del método “dump()” en aplicaciones Laravel. Este paquete proporciona un servidor de volcado (dump server) que permite visualizar los datos volcados de una manera más conveniente y legible, especialmente en entornos de desarrollo.
- **filp/whoops:** Es una herramienta de manejo de errores para aplicaciones PHP.
- **fzaninotto/faker:** Es una biblioteca muy útil para la generación de datos falsos de manera aleatoria en PHP.
- **mockery/mockery:** Es una biblioteca de pruebas unitarias (testing) para PHP que se utiliza para crear objetos simulados (mocks) y realizar afirmaciones (assertions) sobre ellos.
- **nunomaduro/collision:** Es una biblioteca utilizada principalmente en aplicaciones PHP, especialmente en el contexto de Laravel, para mejorar la experiencia de manejo de excepciones y errores de línea de comandos (CLI).

- **phpunit/phpunit:** Es un marco de pruebas unitarias para PHP que facilita la escritura y ejecución de pruebas automatizadas para verificar el correcto funcionamiento de componentes individuales de una aplicación.
- **nesbot/carbon:** Es una biblioteca de manipulación de fechas y horas en PHP. Está diseñada para facilitar la manipulación y formateo de fechas de una manera más intuitiva y legible que las funciones nativas de PHP.

3.1.2 Análisis de funcionalidades.

Para esta etapa, se toman en cuenta todos los endpoints contenidos en el aplicativo Gestión, analizando detenidamente qué hacen y cómo funcionan cada una de ellas. En total, se realizó el análisis a 110 rutas.

Figura 5

Información de las dependencias del aplicativo – Parte 1

Función	Descripción
Login	La ruta valida las credenciales utilizadas para acceder al sistema. Si el usuario es válido, se genera un JSON web token que permite el acceso a otras rutas e información del usuario. En caso contrario, se deniega el acceso.
Obtener abono	Recupera todos los registros de la tabla "abonos". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos y paginar los resultados.
Almacenar abono	Añade un nuevo registro a la tabla "abonos" con la información requerida.
Actualizar abono	Actualiza los datos de un registro en la tabla "abonos" que coincida con el id especificado.
Eliminar abono	Elimina un registro de la tabla "abonos" que coincida con el ID especificado.
Obtener acuerdo de pago	Recupera todos los registros de la tabla "acuerdos_pagos". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos y paginar los resultados.
Almacenar acuerdo de pago	Añade un nuevo registro a la tabla "acuerdos_pagos" con la información requerida.
Actualizar acuerdo de pago	Actualiza los datos de un registro en la tabla "acuerdos_pagos" que coincida con el id especificado.
Eliminar acuerdo de pago	Elimina un registro de la tabla "acuerdos_pagos" que coincida con el ID especificado.
Obtener casa cobranza	Recupera todos los registros de la tabla "casas_cobranzas". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos y paginar los resultados.
Almacenar casa cobranza	Añade un nuevo registro a la tabla "casas_cobranzas" con la información requerida.
Consultar casa cobranza	Recupera los datos de un registro de la tabla "casas_cobranzas" que coincida con el ID proporcionado.
Actualizar casa cobranza	Actualiza los datos de un registro en la tabla "casas_cobranzas" que coincida con el id especificado.
Eliminar casa cobranza	Elimina un registro de la tabla "casas_cobranzas" que coincida con el ID especificado.
Obtener árbol	Recupera la información de la tabla "modulos" que sigue una estructura jerárquica en forma de árbol.
Obtener clientes	Recupera todos los registros de la tabla "clientes". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos y paginar los resultados.
Almacenar clientes	Añade un nuevo registro a la tabla "clientes" con la información requerida.
Actualizar clientes	Actualiza los datos de un registro en la tabla "clientes" que coincida con el id especificado.
Eliminar clientes	Elimina un registro de la tabla "clientes" que coincida con el ID especificado.
Obtener codeudor	Recupera todos los registros de la tabla "codeudores". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos y paginar los resultados.
Almacenar codeudor	Añade un nuevo registro a la tabla "codeudores" con la información requerida.
Actualizar codeudor	Actualiza los datos de un registro en la tabla "codeudores" que coincida con el id especificado.
Eliminar codeudor	Elimina un registro de la tabla "codeudores" que coincida con el ID especificado.
Obtener contacto	Recupera todos los registros de la tabla "informacion_contactos". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos.

Nota: Elaboración propia.

Figura 6

Información de las dependencias del aplicativo – Parte 2

Ver contacto	Obtiene los datos de un registro de la tabla "informacion_contactos" que corresponde al ID especificado.
Almacenar contacto	Añade un nuevo registro a la tabla "informacion_contactos" con la información requerida.
Actualizar contacto	Actualiza los datos de un registro en la tabla "informacion_contactos" que coincide con el id especificado.
Eliminar contacto	Elimina un registro de la tabla "informacion_contactos" que coincide con el ID especificado.
Importar contacto	Se recopilan todos los datos de un archivo .xlsx y los almacena en la tabla "informacion_contactos".
Obtener contactos inválidos	Recupera todos los registros de la tabla "informacion_contactos" donde la columna "invalido" tiene el valor 1.
Obtener convenio	Recupera todos los registros de la tabla "convenios". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos y paginar los resultados.
Almacenar convenio	Añade un nuevo registro a la tabla "convenios" con la información requerida.
Actualizar convenio	Actualiza los datos de un registro en la tabla "convenios" que coincide con el id especificado.
Eliminar convenio	Elimina un registro de la tabla "convenios" que coincide con el ID especificado.
Obtener deudor	Recupera todos los registros de la tabla "deudores". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos y paginar los resultados.
Almacenar deudor	Añade un nuevo registro a la tabla "deudores" con la información requerida.
Actualizar deudor	Actualiza los datos de un registro en la tabla "deudores" que coincide con el id especificado.
Eliminar deudor	Elimina un registro de la tabla "deudores" que coincide con el ID especificado.
Obtener archivo	Recupera un archivo y lo descripta del bucket de S3 especificado, identificándolo mediante la ruta y el nombre enviado en el cuerpo de la solicitud.
Encriptar archivo	Guarda o sobrescribe un archivo encriptado en la ruta especificada.
Exportar usuarios	Exporta todos los registros de la tabla "usuarios" a un archivo.
Exportar deudores	Exporta todos los registros de la tabla "deudores" a un archivo.
Exportar gestiones	Exporta todos los registros de la tabla "gestiones" a un archivo.
Exportar venta cartera	Se recibe un archivo .csv, del cual extrae información y realiza consultas adicionales para generar un archivo. Posteriormente, encripta este archivo y lo guarda en un bucket de S3.
Descargar venta cartera	Descarga un archivo del bucket de S3 utilizando la ruta y el nombre especificados. Posteriormente, descripta el archivo para su descarga.
Eliminar venta cartera	Elimina un archivo almacenado en un bucket de S3 utilizando el nombre y la ruta especificada.
Obtener gestion	Recupera todos los registros de la tabla "gestiones". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos y paginar los resultados.
Almacenar gestion	Añade un nuevo registro a la tabla "gestiones" con la información requerida.
Actualizar gestión	Actualiza los datos de un registro en la tabla "gestiones" que coincide con el id especificado.
Eliminar gestión	Elimina un registro de la tabla "gestiones" que coincide con el ID especificado.

Nota: Elaboración propia.

Figura 7

Información de las dependencias del aplicativo – Parte 3

Obtener gestión obligación	Recupera todos los registros de la tabla "gestion_obligaciones". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos y paginar los resultados.
Almacenar gestión obligación	Añade un nuevo registro a la tabla "gestion_obligaciones" con la información requerida.
Actualizar gestión obligación	Actualiza los datos de un registro en la tabla "gestion_obligaciones" que coincide con el id especificado.
Eliminar gestión obligación	Elimina un registro de la tabla "gestion_obligaciones" que coincide con el ID especificado.
Generar campaña	Crea un informe en formato JSON al consultar diversos parámetros en varias tablas de la base de datos para obtener información sobre las campañas.
Asignación abono	La función presenta problemas debido a la ausencia de la tabla "asignacion_abonos". Por consiguiente, la falta de respuesta puede resultar en un error debido al límite de tiempo de la solicitud.
Informe gestión	Obtienen todos los datos asociados a las obligaciones, los cuales pueden ser filtrados mediante parámetros específicos y paginados según sea necesario.
Informe venta cartera	Recopila información de todos los archivos almacenados en un bucket de S3.
Refresh	Actualiza el JSON web token.
Logout	Destruye la sesión, donde se invalida el JSON web token.
Me	Obtiene la información almacenada en el JSON web token del usuario autenticado.
Obtener modulo	Recupera todos los registros de la tabla "modulos".
Almacenar modulo	Añade un nuevo registro a la tabla "modulos" con la información requerida.
Actualizar modulo	Actualiza los datos de un registro en la tabla "modulos" que coincide con el id especificado.
Eliminar modulo	Elimina un registro de la tabla "modulos" que coincide con el ID especificado.
Obtener municipio	Recupera todos los registros de la tabla "municipios". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos y paginar los resultados.
Almacenar municipio	Añade un nuevo registro a la tabla "municipios" con la información requerida.
Actualizar municipio	Actualiza los datos de un registro en la tabla "municipios" que coincide con el id especificado.
Eliminar municipio	Elimina un registro de la tabla "municipios" que coincide con el ID especificado.
Obtener obligación	Recupera todos los registros de la tabla "obligaciones". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos y paginar los resultados.
Almacenar obligación	Añade un nuevo registro a la tabla "obligaciones" con la información requerida.
Actualizar obligación	Actualiza los datos de un registro en la tabla "obligaciones" que coincide con el id especificado.
Eliminar obligación	Elimina un registro de la tabla "obligaciones" que coincide con el ID especificado.
Obtener presupuesto	Recupera todos los registros de la tabla "presupuestos". Además, ofrece la posibilidad de paginar los resultados.
Importar presupuesto	Se recopilan todos los datos de un archivo .xlsx y los almacena en la tabla "presupuestos".
Actualizar presupuesto	No tiene controlador presente, por ende no realiza ninguna función.
Eliminar presupuesto	No tiene controlador presente, por ende no realiza ninguna función.

Nota: Elaboración propia.

Figura 8*Información de las dependencias del aplicativo – Parte 4*

Obtener referencia	Recupera todos los registros de la tabla "referencias". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos y paginar los resultados.
Almacenar referencia	Añade un nuevo registro a la tabla "referencias" con la información requerida.
Actualizar referencia	Actualiza los datos de un registro en la tabla "referencias" que coincida con el id especificado.
Eliminar referencia	Elimina un registro de la tabla "referencias" que coincida con el ID especificado.
Obtener saldos	Recopila información sobre el saldo pendiente de las obligaciones, clasificándolas en aquellas que tienen un acuerdo de pago y las que no lo tienen.
Obtener sociodemográfica	Recupera todos los registros de la tabla "socio_demograficas". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos y paginar los resultados.
Almacenar sociodemográfica	Añade un nuevo registro a la tabla "socio_demograficas" con la información requerida.
Actualizar sociodemográfica	Actualiza los datos de un registro en la tabla "socio_demograficas" que coincida con el id especificado.
Eliminar sociodemográfica	Elimina un registro de la tabla "socio_demograficas" que coincida con el ID especificado.
Importar sociodemográfica	Se recopilan todos los datos de un archivo .csv y los almacena en la tabla "socio_demograficas".
Obtener subtipo gestión	Recupera todos los registros de la tabla "sub_tipo_gestiones". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos y paginar los resultados.
Almacenar subtipo gestión	Añade un nuevo registro a la tabla "sub_tipo_gestiones" con la información requerida.
Actualizar subtipo gestión	Actualiza los datos de un registro en la tabla "sub_tipo_gestiones" que coincida con el id especificado.
Eliminar subtipo gestión	Elimina un registro de la tabla "sub_tipo_gestiones" que coincida con el ID especificado.
Obtener tipo gestión	Recupera todos los registros de la tabla "tipo_gestiones". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos y paginar los resultados.
Almacenar tipo gestión	Añade un nuevo registro a la tabla "tipo_gestiones" con la información requerida.
Actualizar tipo gestión	Actualiza los datos de un registro en la tabla "tipo_gestiones" que coincida con el id especificado.
Eliminar tipo gestión	Elimina un registro de la tabla "tipo_gestiones" que coincida con el ID especificado.
Obtener usuario	Recupera todos los registros de la tabla "usuarios". Además, ofrece la posibilidad de filtrar estos registros según parámetros específicos y paginar los resultados.
Almacenar usuario	Añade un nuevo registro a la tabla "usuarios" con la información requerida.
Consultar usuario	Recupera los datos de un registro de la tabla "usuarios" que coincida con el ID proporcionado.
Actualizar usuario	Actualiza los datos de un registro en la tabla "usuarios" que coincida con el id especificado.
Eliminar usuario	Elimina un registro de la tabla "usuarios" que coincida con el ID especificado.
Modificar password	Actualiza la contraseña del usuario actualmente autenticado.
Obtener datos SIG	Realiza una solicitud al aplicativo SIG2. Sin embargo, debido a que el aplicativo ha sido actualizado a una versión completamente nueva (SIG3), las funciones previamente soportadas ya no son operativas.

Nota: Elaboración propia.

Figura 9*Información de las dependencias del aplicativo – Parte 5*

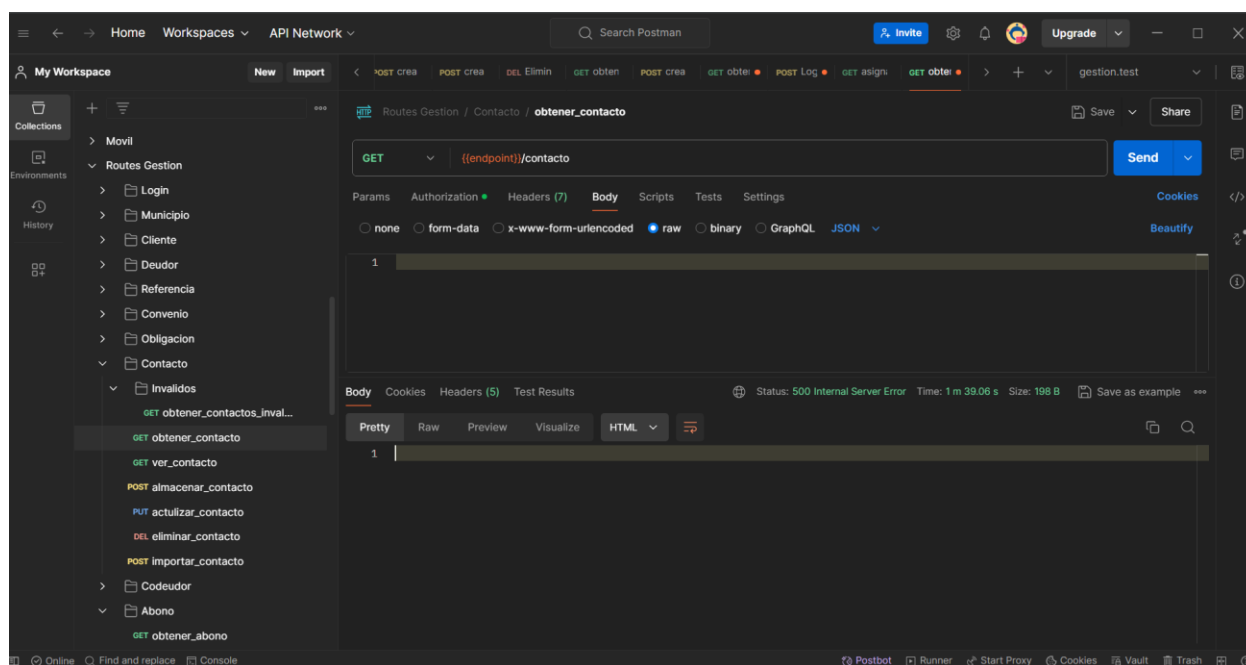
Generar documento	Cumple la función de generar un documento PDF basado en la información proporcionada en el cuerpo de la solicitud.
Datos presupuesto	Recopila los datos de la tabla "graficas" y los organiza para crear un gráfico.
Enviar SMS	Envía un mensaje de texto a través de un servicio externo que permite la personalización del mensaje y la definición de los destinatarios.
Enviar email	Envía correos electrónicos con una estructura base definida y la posibilidad de seleccionar una plantilla específica.
Intereses FGA	Recibe una lista de IDs de obligaciones para obtener y sumar sus intereses, proporcionando así el total de intereses de dichas obligaciones.
Obtener acuerdo gestión	Se recopilan todos los datos de un archivo .xlsx y los almacena en la tabla "presupuestos".
Obtener grafica presupuesto	Obtiene todos los registros de gestiones realizadas por un usuario durante un mes específico y calcula la cantidad de gestiones y acuerdos de pago realizados.
Cargue masivo gestión	Se recopilan todos los datos de un archivo .csv y los almacena en la tabla "gestiones", retornando el id asignado al registro.

Nota: Elaboración propia.

De las 110 rutas analizadas, 29 de ellas presentaban fallas al momento de realizar peticiones. Entre estas, 10 mostraban respuestas tardías debido al volumen de datos que intentaban recopilar, superando así el tiempo límite establecido y debido a los recursos limitados de hardware asignados a la base de datos del proveedor AWS. Además, 16 rutas no podían comunicarse con el servicio de S3 proporcionado por AWS debido a restricciones de seguridad. Se identificaron también 2 rutas que no cumplían ninguna función útil, así como 1 ruta que dejó de ser funcional como resultado de un cambio de versión en otro aplicativo.

Figura 10

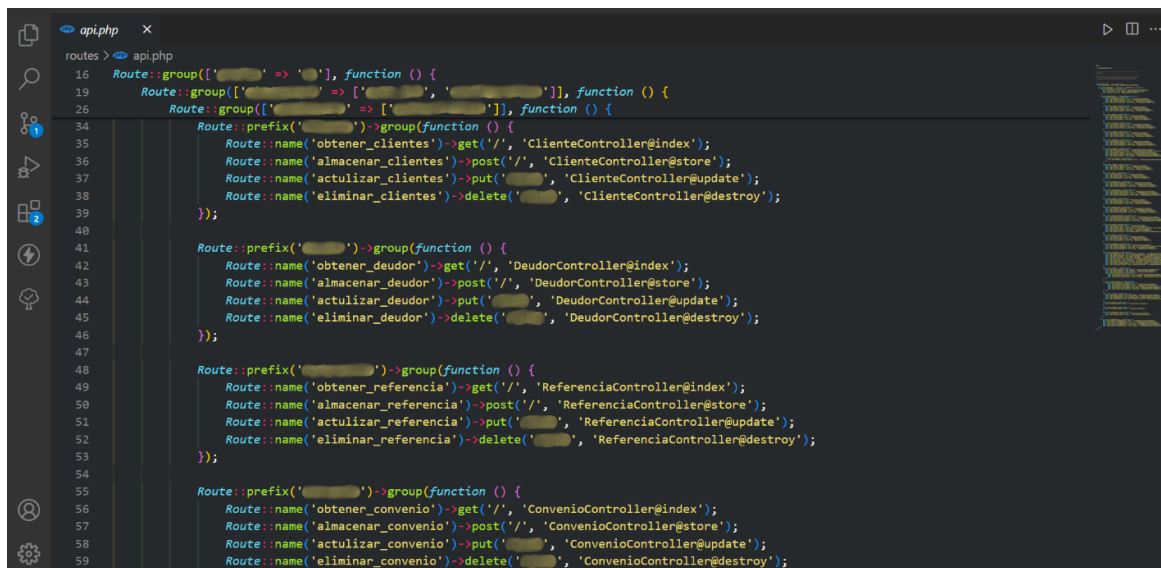
Respuesta de error en API.



Nota: Elaboración propia.

Figura 11

Archivo de rutas.



```

16 Route::group(['prefix' => ''], function () {
17     Route::group(['prefix' => ''], function () {
26         Route::group(['prefix' => ''], function () {
34             Route::prefix('')->group(function () {
35                 Route::name('obtener_clientes')->get('/', 'ClienteController@index');
36                 Route::name('almacenar_clientes')->post('/', 'ClienteController@store');
37                 Route::name('actualizar_clientes')->put('/', 'ClienteController@update');
38                 Route::name('eliminar_clientes')->delete('/', 'ClienteController@destroy');
39             });
40         });
41     });
42     Route::prefix('')->group(function () {
43         Route::name('obtener_deudor')->get('/', 'DeudorController@index');
44         Route::name('almacenar_deudor')->post('/', 'DeudorController@store');
45         Route::name('actualizar_deudor')->put('/', 'DeudorController@update');
46         Route::name('eliminar_deudor')->delete('/', 'DeudorController@destroy');
47     });
48     Route::prefix('')->group(function () {
49         Route::name('obtener_referencia')->get('/', 'ReferenciaController@index');
50         Route::name('almacenar_referencia')->post('/', 'ReferenciaController@store');
51         Route::name('actualizar_referencia')->put('/', 'ReferenciaController@update');
52         Route::name('eliminar_referencia')->delete('/', 'ReferenciaController@destroy');
53     });
54     Route::prefix('')->group(function () {
56         Route::name('obtener_convenio')->get('/', 'ConvenioController@index');
57         Route::name('almacenar_convenio')->post('/', 'ConvenioController@store');
58         Route::name('actualizar_convenio')->put('/', 'ConvenioController@update');
59         Route::name('eliminar_convenio')->delete('/', 'ConvenioController@destroy');

```

Nota: Elaboración propia.

3.2 Diseñar un plan para la migración del framework que usa el aplicativo considerando posibles conflictos.

3.2.1 Plan de migración.

Para esta etapa, se define el método para llevar a cabo la migración del framework. En este caso, la migración se realizará de manera progresiva, avanzando de una versión a la siguiente de forma controlada y positiva, comenzando desde la versión 5.8 hasta alcanzar la versión 10 de Laravel.

Figura 12

Fases en el proceso de migración.

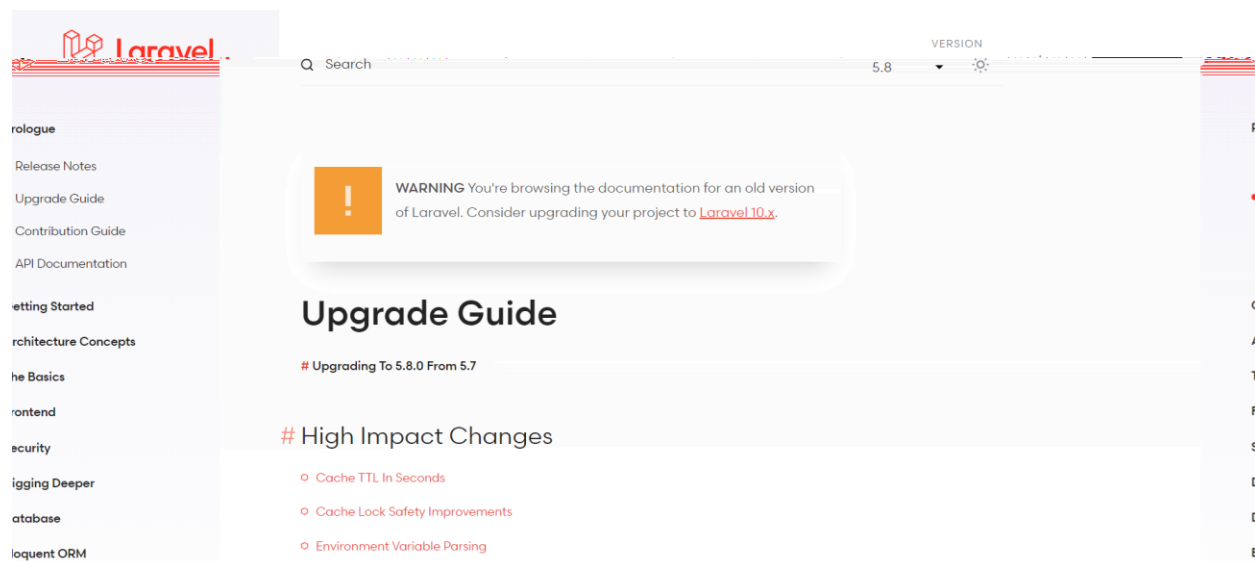


Nota: Elaboración propia.

Todo el proceso es asistido por la documentación oficial proporcionada por el framework, con el fin de obtener una guía para efectuar los cambios necesarios en cada una de las fases definidas.

Figura 13

Documentación proporcionada por el framework.

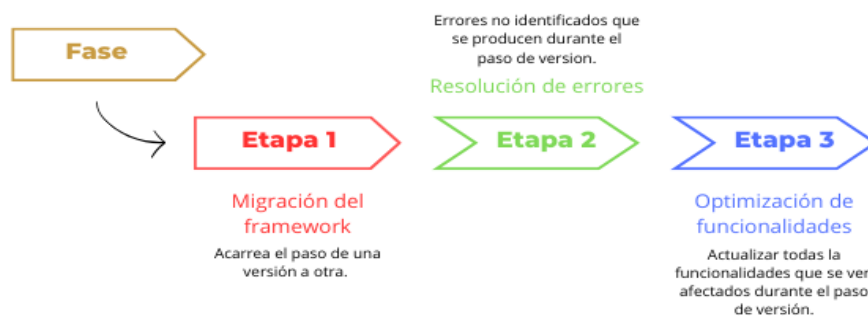


Nota: (Laravel - The PHP Framework For Web Artisans, s/f)

Cada una de estas fases consta de 3 etapas para ser completada a satisfacción.

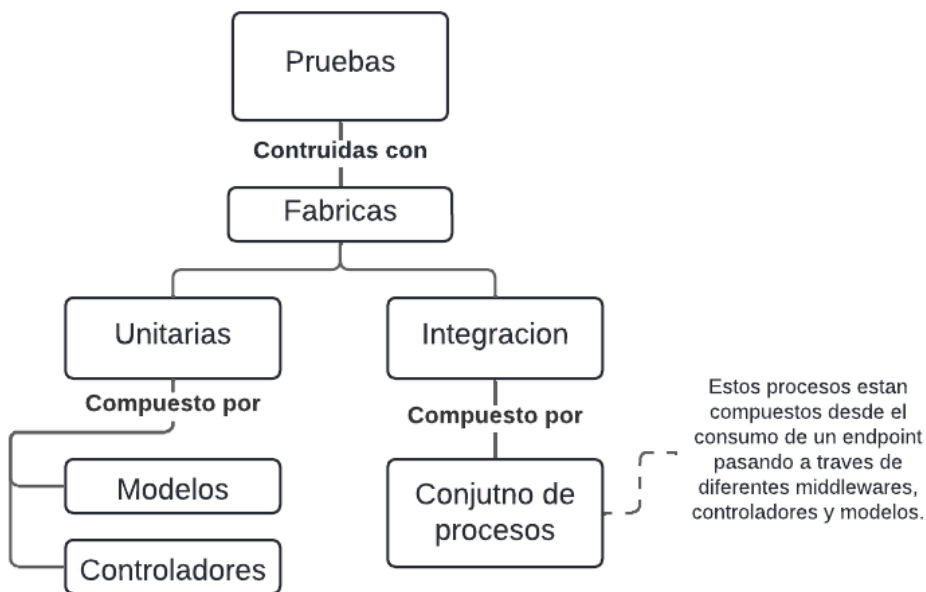
Figura 14

Etapas por fases.



Nota: Elaboración propia

Una vez finalizada la fase 6 - Laravel 10, como se muestra en la figura 12, se procederá a identificar posibles brechas de seguridad y la elaboración de pruebas unitarias y de integración. Estas pruebas se realizarán al aplicativo para garantizar el correcto funcionamiento de todas sus funcionalidades.

Figura 15*Estructura de pruebas.**Nota: Elaboración propia*

3.2.2 Configuración de entornos de desarrollo.

En esta etapa, se lleva a cabo la instalación y configuración de todas las herramientas, software y recursos necesarios para modificar, probar y ejecutar el aplicativo “Gestión V1” en un entorno de trabajo óptimo.

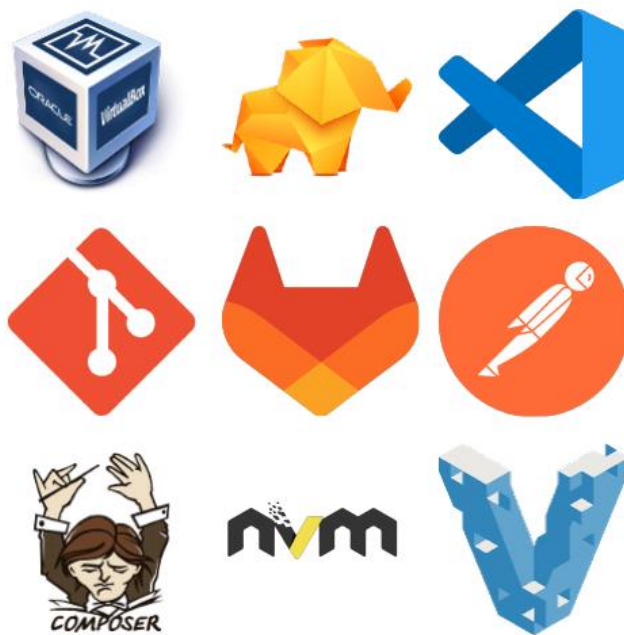
Se adquieren y configuran las siguientes herramientas para lograrlo:

- VirtualBox - (Máquina virtual)
- TablePlus – (Gestor de base de datos)
- Visual Studio Code – (Editor de código fuente)
- Postman – (Plataforma API para crear y utilizar API)

- Git – (Sistema de control de versiones)
- GitLab
- Nvm – (Administrador de versiones de Node.js)
- Vagrant

Figura 16

Herramientas de software utilizadas en el proceso.

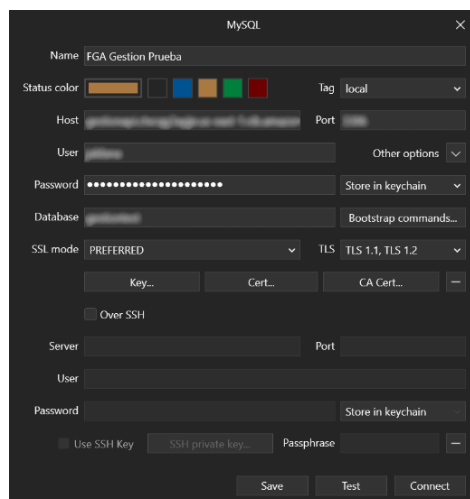


Nota: Elaboración propia.

En el gestor de bases de datos, se establece la conexión con el servicio de bases de datos de prueba proporcionado por el proveedor Amazon Web Services (AWS).

Figura 17

Conexión a base de datos del proveedor AWS.

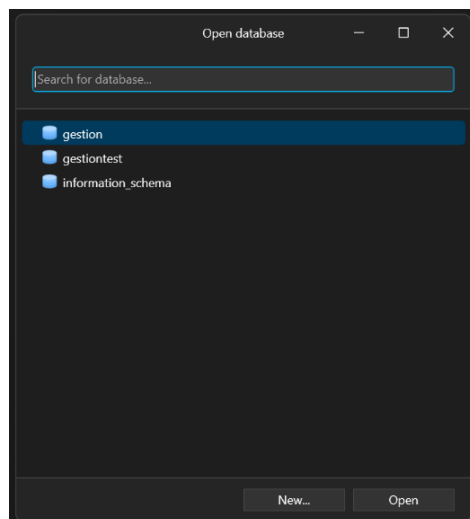


Nota: Elaboración propia.

En este servicio, se encuentran dos bases de datos disponibles, ambas destinadas a pruebas y ya pobladas con datos de ejemplo.

Figura 18

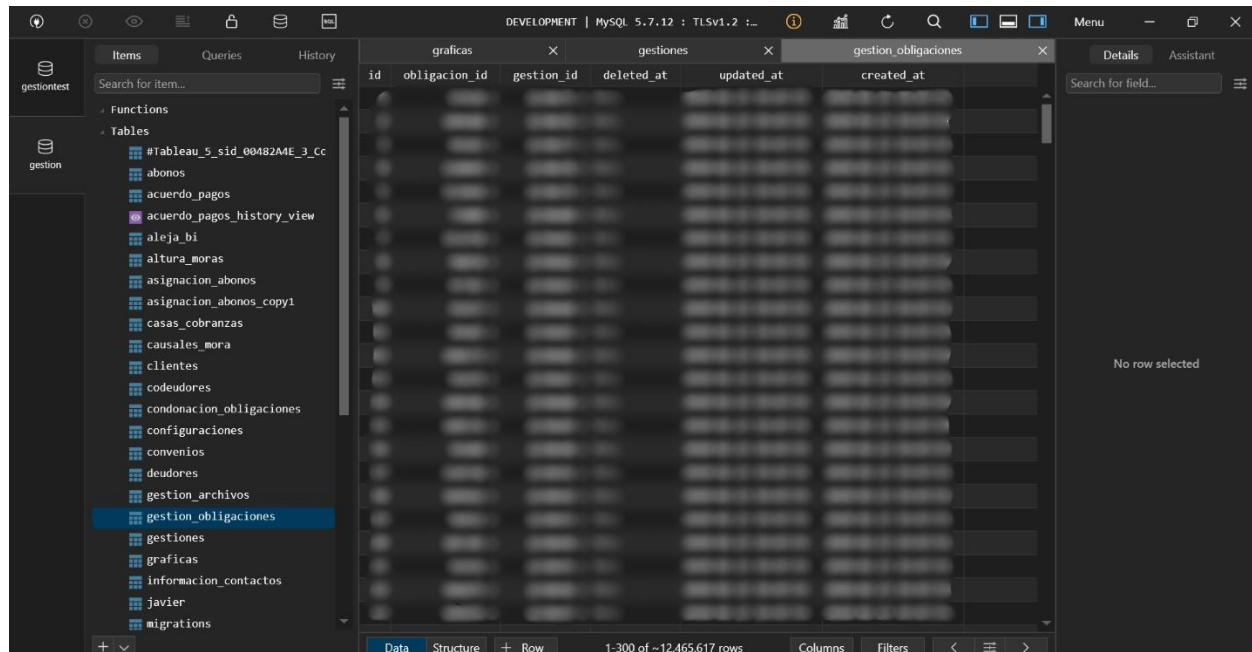
Bases de datos en el proveedor de servicios.



Nota: Elaboración propia.

Figura 19

Enlace con base de datos.

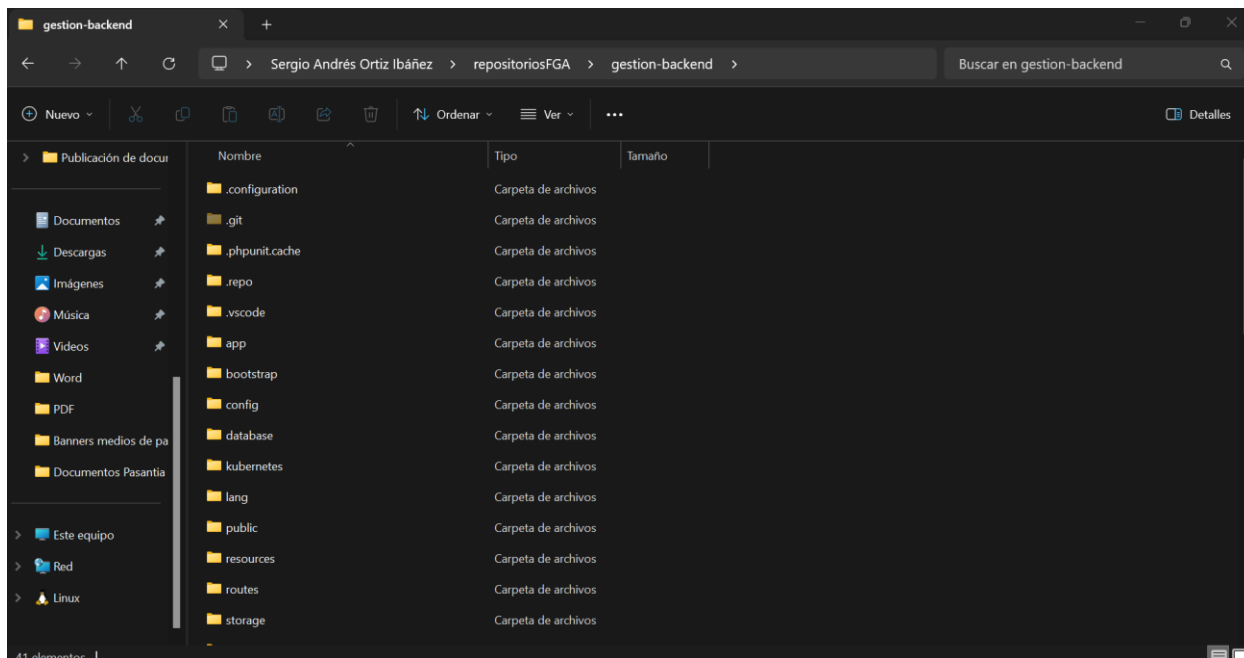


Nota: Elaboración propia.

Se crea una carpeta en la raíz del usuario donde se clonará el repositorio del aplicativo Gestión.

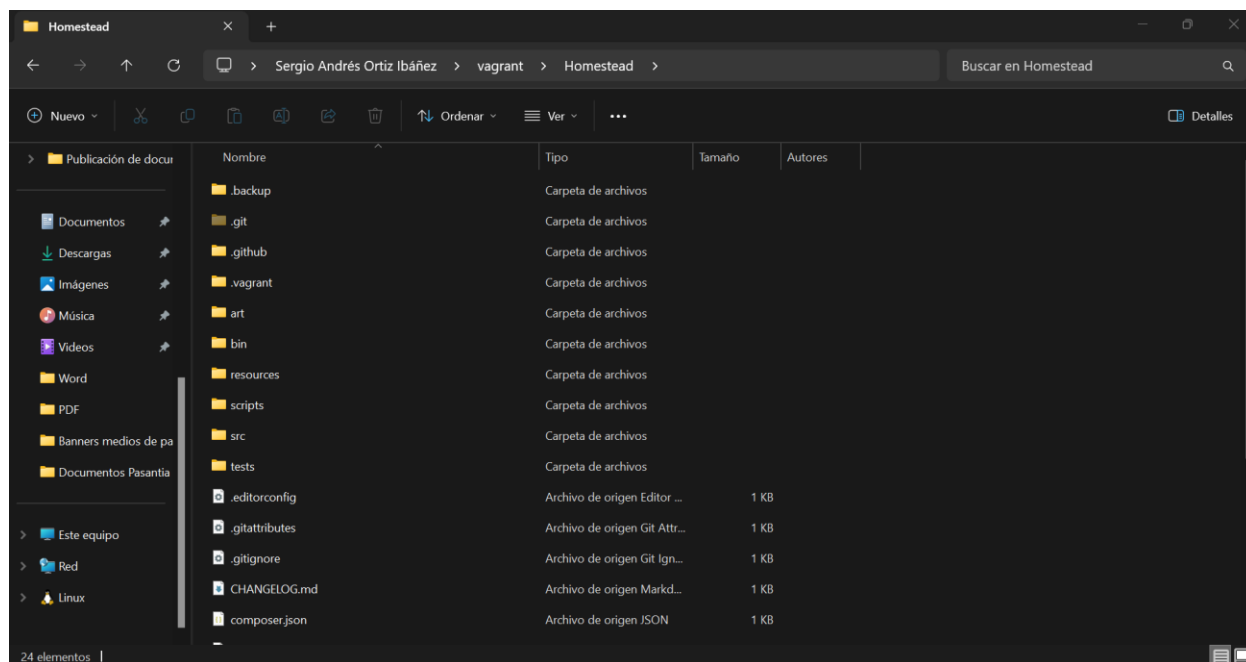
Figura 20

Repositorio aplicativo web Gestión.



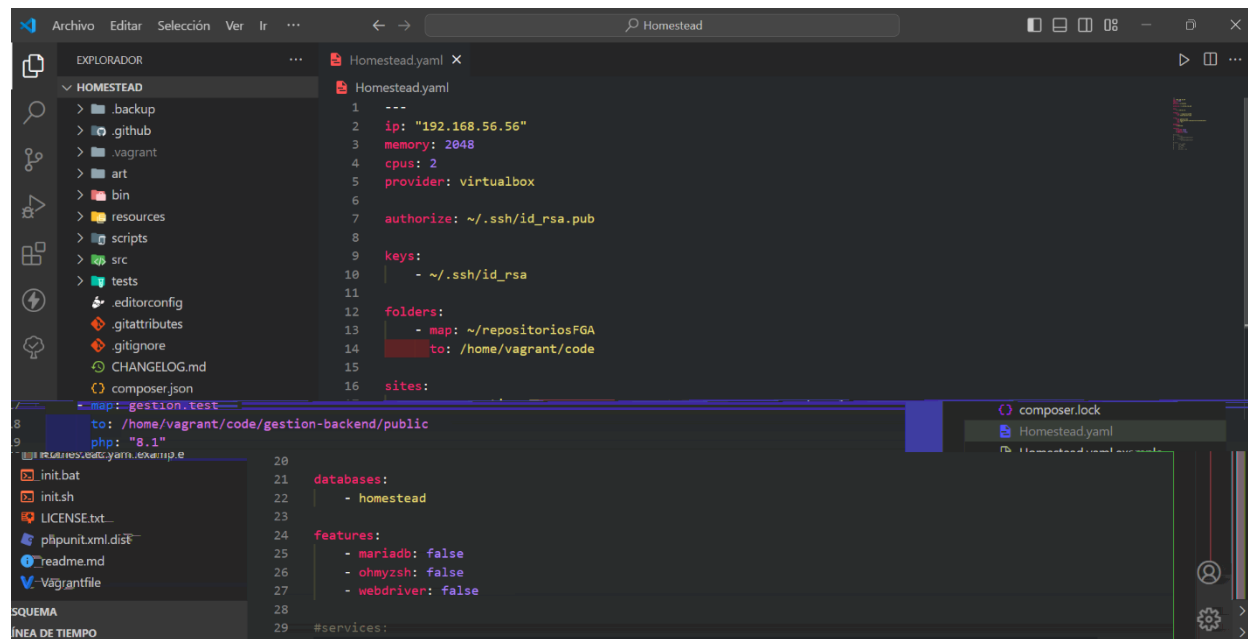
Nota: Elaboración propia.

Se procede a crear y configurar un entorno de desarrollo virtualizado utilizando Vagrant y VirtualBox, clonando el repositorio proporcionado por la documentación oficial de Laravel sobre Homestead. Esta herramienta proporciona un entorno de desarrollo completo, eliminando la necesidad de instalar PHP, un servidor web u otro software de servidor en la máquina local.

Figura 21*Repositorio Homestead*

Nota: Elaboración propia.

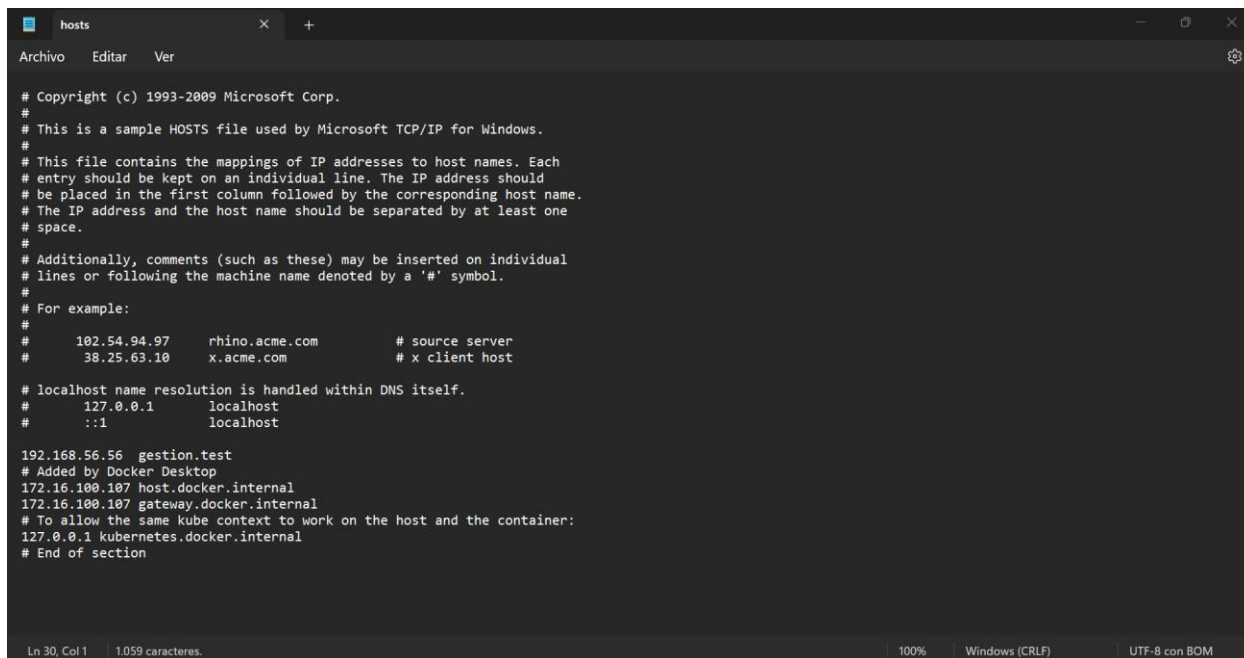
Dentro del directorio Homestead, desde la consola de comandos, se ejecuta el comando “init.bat” para crear el archivo Homestead.yaml, donde se efectúan los ajustes requeridos. En dicho archivo, se especifica la ruta de la carpeta que alberga el aplicativo “Gestión v1” para compartirla con el entorno Homestead. Asimismo, se define la versión de PHP a emplear en el sitio y se configura el servicio de base de datos MySQL.

Figura 22*Archivo de configuración Homestead**Nota: Elaboración propia.*

Para dirigir las solicitudes del sitio hacia la máquina virtual de Homestead, se realiza la configuración del archivo hosts, ubicado en C:\Windows\System32\drivers\etc\hosts.

Figura 23

Archivo hosts del sistema operativo



```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com   # source server
#       38.25.63.10       x.acme.com       # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1         localhost
#       ::1               localhost

192.168.56.56 gestion.test
# Added by Docker Desktop
172.16.100.107 host.docker.internal
172.16.100.107 gateway.docker.internal
# To allow the same kube context to work on the host and the container:
127.0.0.1 kubernetes.docker.internal
# End of section
```

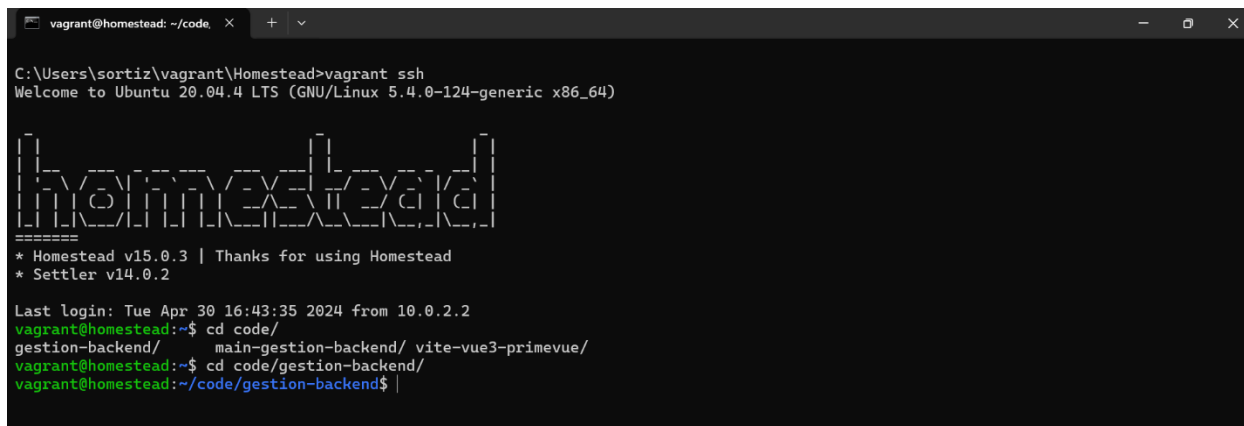
Nota: Elaboración propia.

Tras realizar las configuraciones mencionadas anteriormente, se ejecuta el comando “vagrant up” desde el directorio de homestead, lo que resulta en la creación de una máquina virtual en la aplicación VirtualBox.

Figura 24*Máquina virtual Homestead*

Nota: Elaboración propia

Para acceder a la máquina virtual homestead, se genera una clave SSH y se ejecuta el comando “vagrant ssh” desde el directorio homestead. Una vez conectados a la máquina virtual, es posible dirigirse al directorio compartido del aplicativo “Gestión v1”.

Figura 25*Conexión a la máquina virtual Homestead*


```

vagrant@homestead: ~/code
C:\Users\sortiz\vagrant\Homestead>vagrant ssh
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-124-generic x86_64)

 _____
| _ \| | | | _ \| | | | | | | | |
| |_) | | | | |_) | | | |
| |_) | | | | |_) | | | |
|_| \_|_|_|_|_| \_|_|_|_|_|

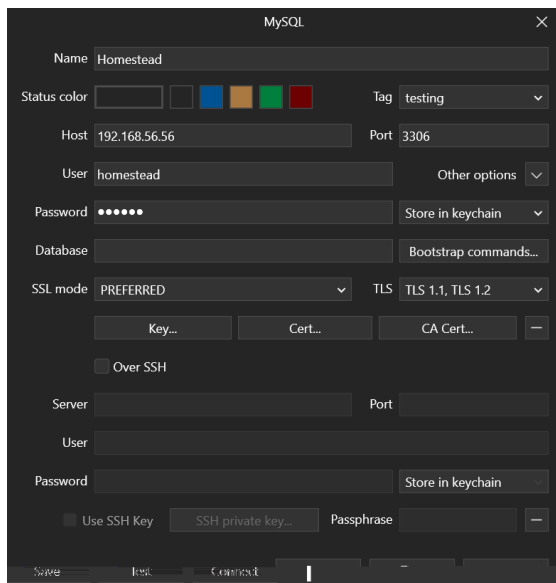
=====
* Homestead v15.0.3 | Thanks for using Homestead
* Settler v14.0.2

Last login: Tue Apr 30 16:43:35 2024 from 10.0.2.2
vagrant@homestead:~$ cd code/
gestion-backend/      main-gestion-backend/ vite-vue3-primevue/
vagrant@homestead:~$ cd code/gestion-backend/
vagrant@homestead:~/code/gestion-backend$

```

Nota: Elaboración propia

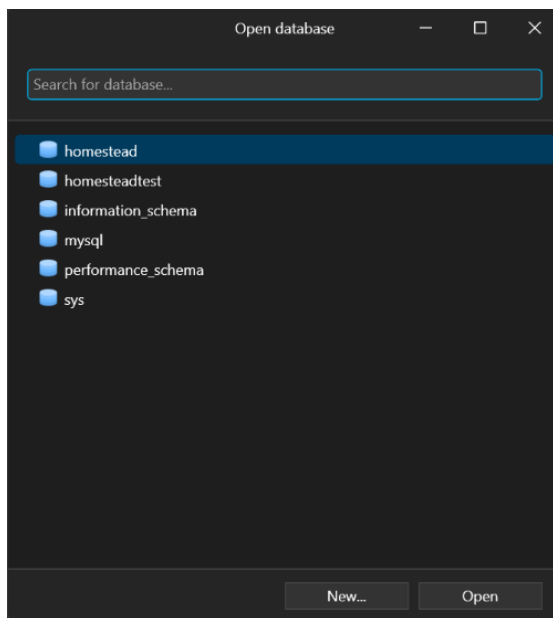
Una vez generada la máquina virtual Homestead, es posible establecer conexión con el servicio de base de datos que ésta proporciona.

Figura 26*Conexión al servicio de base de datos de la máquina virtual Homestead*

Nota: Elaboración propia

En el servicio de base de datos, se han creado dos bases de datos para simular la estructura de la base de datos de prueba ofrecida por el proveedor AWS.

Figura 27

Bases de datos de la máquina virtual Homestead

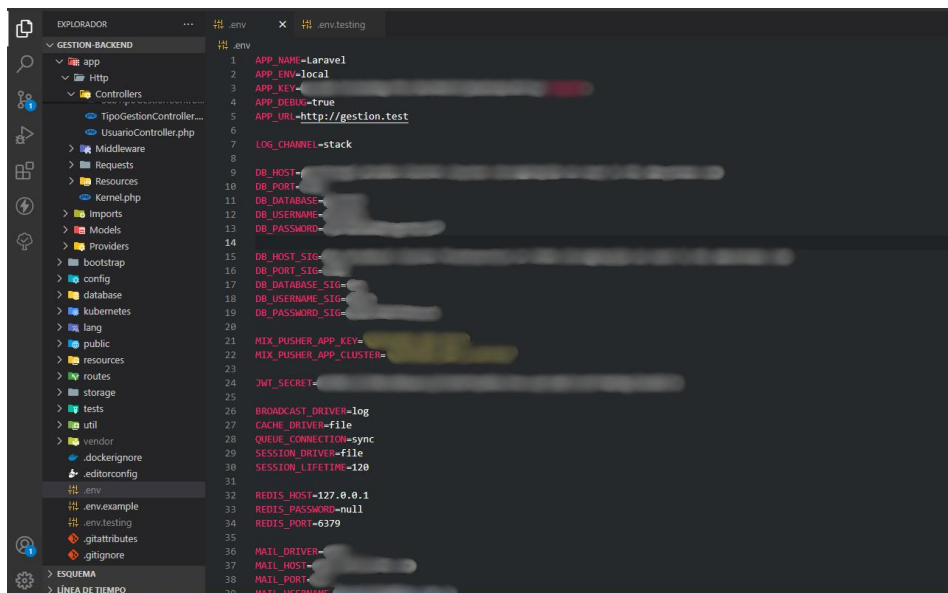


Nota: Elaboración propia

En el repositorio del aplicativo “Gestión v1”, se incluye el archivo .env para las variables de entorno, donde se configuran las credenciales para todos los servicios y la base de datos de prueba del proveedor AWS necesarias para el funcionamiento de la aplicación. Además, se ha añadido un segundo archivo llamado .env.testing con la diferencia que establecerá conexión con la base de datos proporcionada por la máquina virtual homestead.

Figura 28

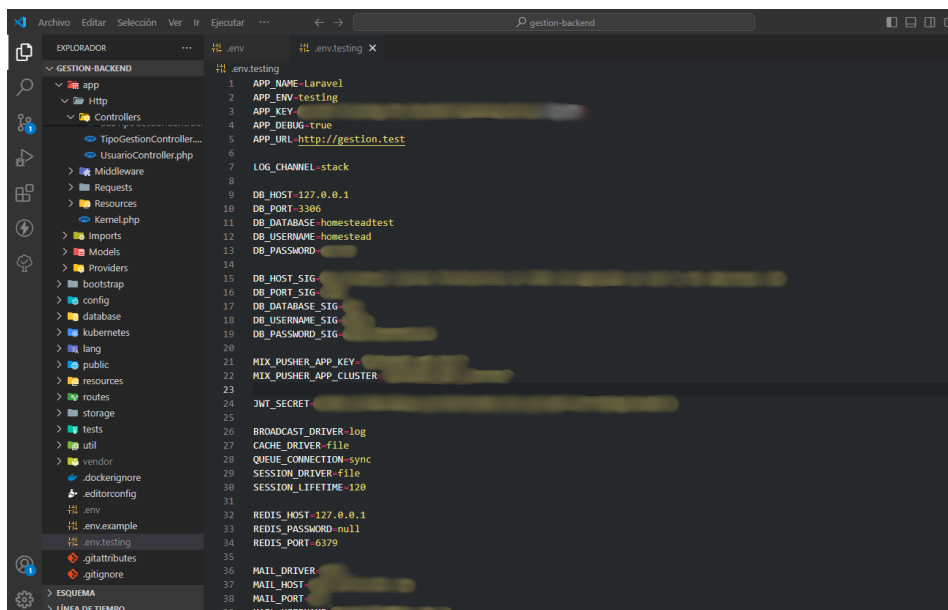
Archivo de variables de entorno (.env)



Nota: Elaboración propia

Figura 29

Archivo de variables de entorno para pruebas (.env.testing)



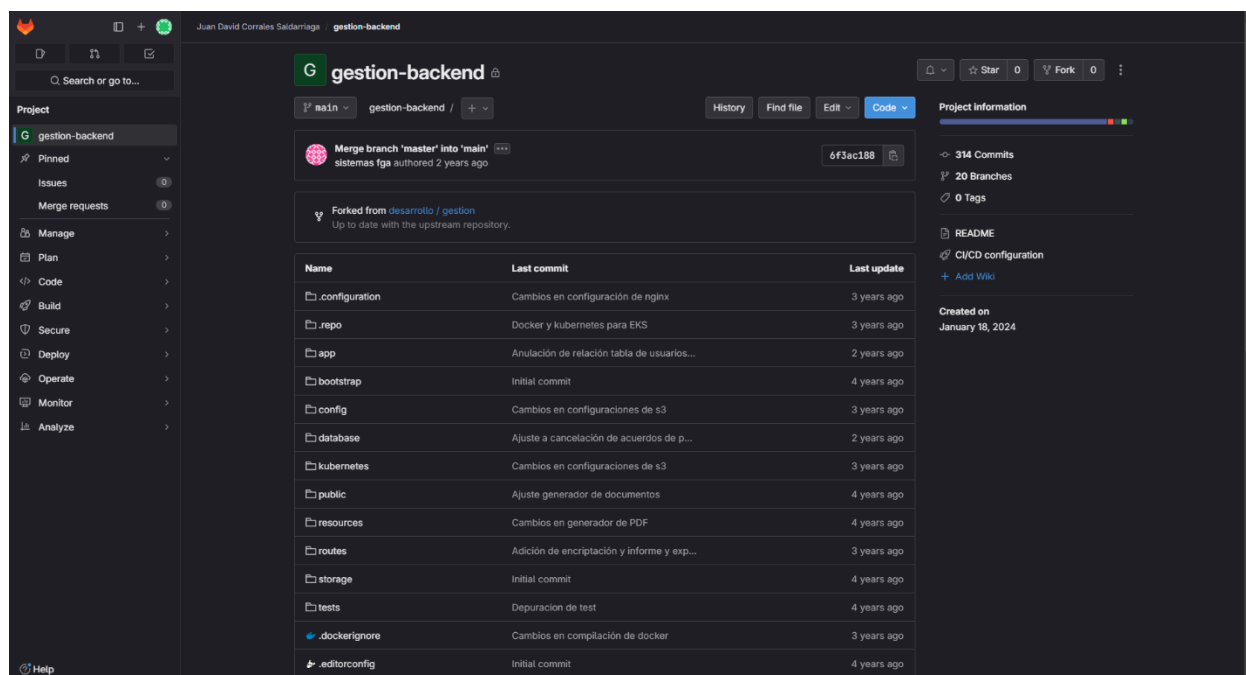
Nota: Elaboración propia

3.2.3 Control de versiones.

El control de versiones se centraliza en un repositorio de GitLab, donde se cargan los cambios de cada commit (confirmación) realizado, y generando una branch (rama) en el repositorio de manera estratégica para evitar la mezcla de versiones.

Figura 30

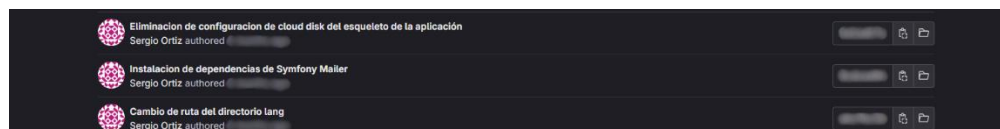
Repositorio centralizado del aplicativo “Gestión v1”



Nota: Elaboración propia

Figura 31

Confirmaciones realizadas al repositorio.

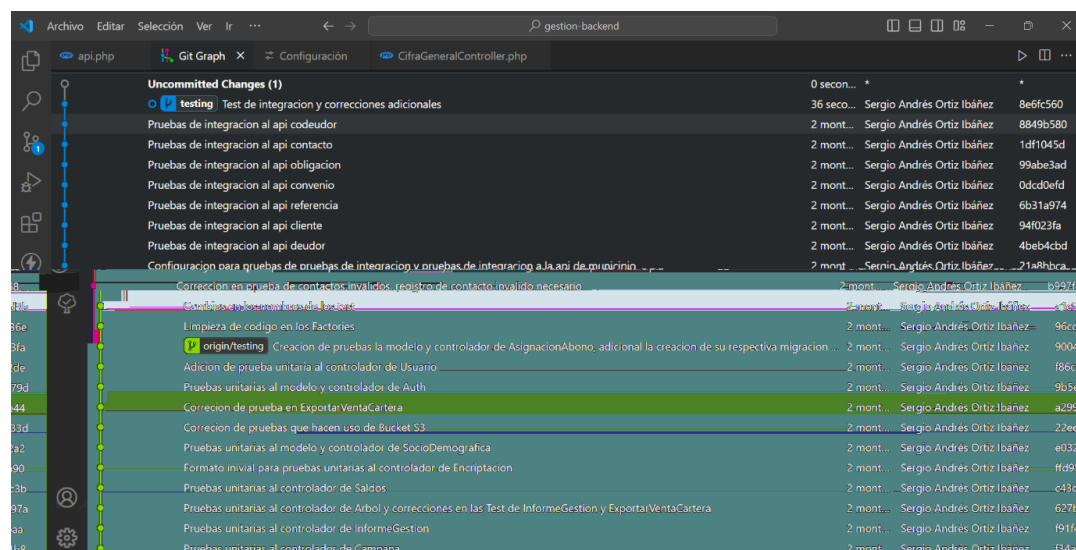


Nota: Elaboración propia

Además, se presenta como evidencia el registro de confirmaciones y ramas utilizadas mediante la extensión Git Graph en Visual Studio Code.

Figura 32

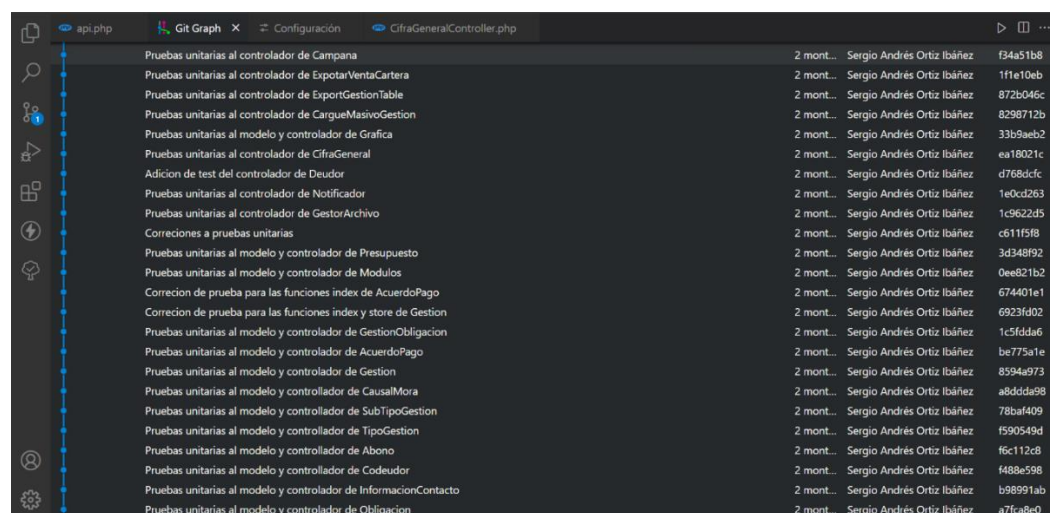
Listado de confirmaciones y ramas – Parte 1



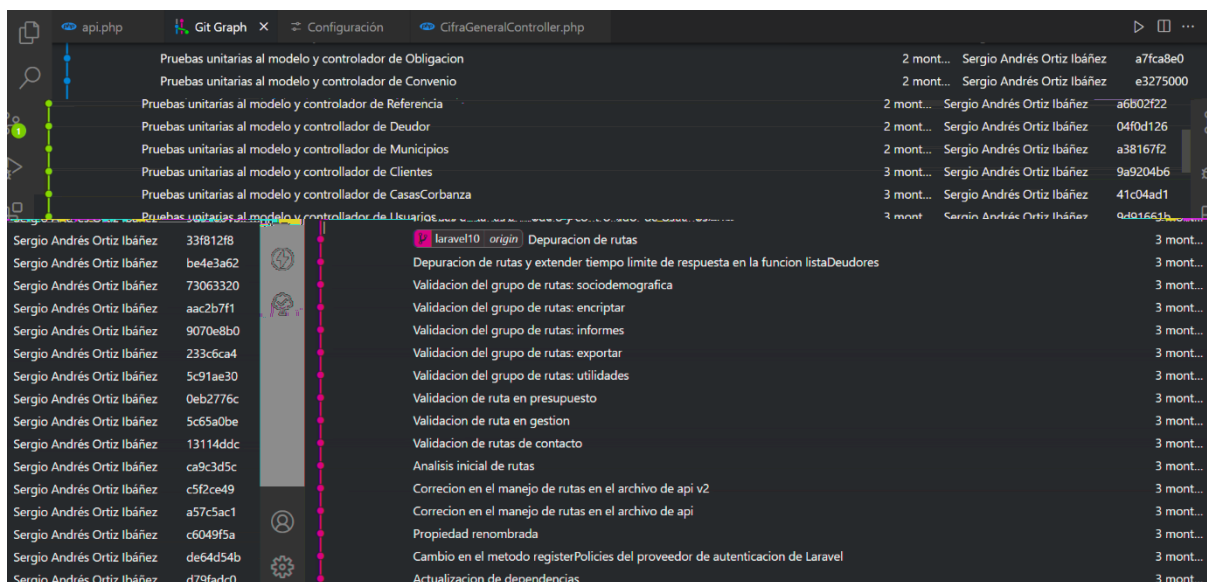
Nota: Elaboración propia

Figura 33

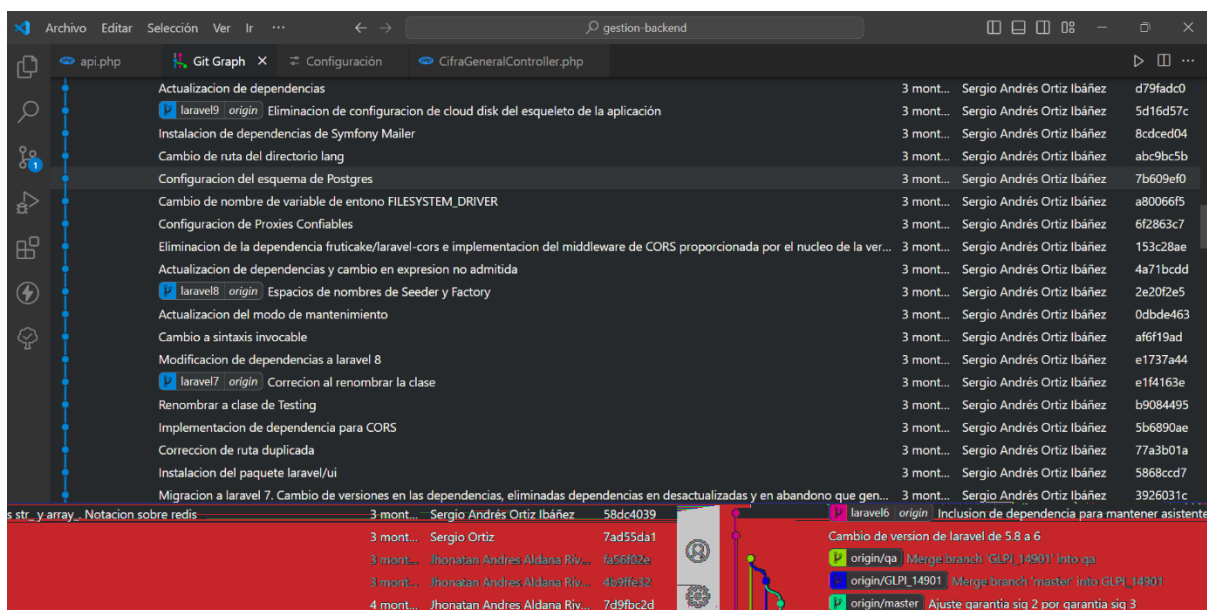
Listado de confirmaciones y ramas – Parte 2



Nota: Elaboración propia

Figura 34*Listado de confirmaciones y ramas – Parte 3*


Commit Hash	Author	Message	Date
a7fca8e0	Sergio Andrés Ortiz Ibáñez	Pruebas unitarias al modelo y controlador de Obligacion	2 mont...
e3275000	Sergio Andrés Ortiz Ibáñez	Pruebas unitarias al modelo y controlador de Convenio	2 mont...
a6802f22	Sergio Andrés Ortiz Ibáñez	Pruebas unitarias al modelo y controlador de Referencia	2 mont...
04f0d126	Sergio Andrés Ortiz Ibáñez	Pruebas unitarias al modelo y controlador de Deudor	2 mont...
a38167f2	Sergio Andrés Ortiz Ibáñez	Pruebas unitarias al modelo y controlador de Municipios	2 mont...
9a9204b6	Sergio Andrés Ortiz Ibáñez	Pruebas unitarias al modelo y controlador de Clientes	3 mont...
41c04ad1	Sergio Andrés Ortiz Ibáñez	Pruebas unitarias al modelo y controlador de CasasCorbanza	3 mont...
9d01661b	Sergio Andrés Ortiz Ibáñez	Pruebas unitarias al modelo y controlador de Usuarios	3 mont...
33f812f8	Sergio Andrés Ortiz Ibáñez	Depuracion de rutas	3 mont...
be4e3a62	Sergio Andrés Ortiz Ibáñez	Depuracion de rutas y extender tiempo limite de respuesta en la funcion listaDeudores	3 mont...
73063320	Sergio Andrés Ortiz Ibáñez	Validacion del grupo de rutas: sociodemografica	3 mont...
aac2b7f1	Sergio Andrés Ortiz Ibáñez	Validacion del grupo de rutas: encriptar	3 mont...
9070e8b0	Sergio Andrés Ortiz Ibáñez	Validacion del grupo de rutas: informes	3 mont...
233c6ca4	Sergio Andrés Ortiz Ibáñez	Validacion del grupo de rutas: exportar	3 mont...
5c91ae30	Sergio Andrés Ortiz Ibáñez	Validacion del grupo de rutas: utilidades	3 mont...
0eb2776c	Sergio Andrés Ortiz Ibáñez	Validacion de ruta en presupuesto	3 mont...
5c65a0be	Sergio Andrés Ortiz Ibáñez	Validacion de ruta en gestion	3 mont...
13114ddc	Sergio Andrés Ortiz Ibáñez	Validacion de rutas de contacto	3 mont...
ca9c3d5c	Sergio Andrés Ortiz Ibáñez	Analisis inicial de rutas	3 mont...
c5f2ce49	Sergio Andrés Ortiz Ibáñez	Correccion en el manejo de rutas en el archivo de api v2	3 mont...
a57c5ac1	Sergio Andrés Ortiz Ibáñez	Correccion en el manejo de rutas en el archivo de api	3 mont...
c6049f5a	Sergio Andrés Ortiz Ibáñez	Propiedad renombrada	3 mont...
de64d54b	Sergio Andrés Ortiz Ibáñez	Cambio en el metodo registerPolicies del proveedor de autenticacion de Laravel	3 mont...
d79fadc0	Sergio Andrés Ortiz Ibáñez	Actualizacion de dependencias	3 mont...

*Nota: Elaboración propia***Figura 35***Listado de confirmaciones y ramas – Parte 4*


Commit Hash	Author	Message	Date
d79fadc0	Sergio Andrés Ortiz Ibáñez	Actualizacion de dependencias	3 mont...
5d16d57c	Sergio Andrés Ortiz Ibáñez	Eliminacion de configuracion de cloud disk del esqueleto de la aplicación	3 mont...
8cdced04	Sergio Andrés Ortiz Ibáñez	Instalacion de dependencias de Symfony Mailer	3 mont...
abc9bc5b	Sergio Andrés Ortiz Ibáñez	Cambio de ruta del directorio lang	3 mont...
7b609ef0	Sergio Andrés Ortiz Ibáñez	Configuracion del esquema de Postgres	3 mont...
a80066f5	Sergio Andrés Ortiz Ibáñez	Cambio de nombre de variable de entorno FILESYSTEM_DRIVER	3 mont...
6f2863c7	Sergio Andrés Ortiz Ibáñez	Configuracion de Proxies Confiabiles	3 mont...
153c28ae	Sergio Andrés Ortiz Ibáñez	Eliminacion de la dependencia fruticake/laravel-cors e implementacion del middleware de CORS proporcionada por el nucleo de la ver...	3 mont...
4a71bcd5	Sergio Andrés Ortiz Ibáñez	Actualizacion de dependencias y cambio en expresion no admitida	3 mont...
2e20f2e5	Sergio Andrés Ortiz Ibáñez	Espacios de nombres de Seeder y Factory	3 mont...
0dbde463	Sergio Andrés Ortiz Ibáñez	Actualizacion del modo de mantenimiento	3 mont...
af6f19ad	Sergio Andrés Ortiz Ibáñez	Cambio a sintaxis invocable	3 mont...
e1737a44	Sergio Andrés Ortiz Ibáñez	Modificacion de dependencias a laravel 8	3 mont...
e1f4163e	Sergio Andrés Ortiz Ibáñez	Correccion al renombrar la clase	3 mont...
b9084495	Sergio Andrés Ortiz Ibáñez	Renombrar a clase de Testing	3 mont...
5b6890ae	Sergio Andrés Ortiz Ibáñez	Implementacion de dependencia para CORS	3 mont...
77a3b01a	Sergio Andrés Ortiz Ibáñez	Correccion de ruta duplicada	3 mont...
5868ccd7	Sergio Andrés Ortiz Ibáñez	Instalacion del paquete laravel/ui	3 mont...
3926031c	Sergio Andrés Ortiz Ibáñez	Migracion a laravel 7. Cambio de versiones en las dependencias, eliminadas dependencias en desactualizadas y en abandono que gen...	3 mont...
58dc4039	Sergio Andrés Ortiz Ibáñez	Notacion sobre redis	3 mont...
7ad55da1	Sergio Ortiz		3 mont...
fa56f02e	Jhonatan Andres Aldana Riv...		3 mont...
4b99fe32	Jhonatan Andres Aldana Riv...		3 mont...
7d9fbc2d	Jhonatan Andres Aldana Riv...		4 mont...
		Cambio de version de laravel de 5.8 a 6	
	origin/qa	Merge branch 'GLPL_14901' into qa	
	origin/GLPL_14901	Merge branch 'master' into GLPL_14901	
	origin/master	Ajuste garantia sig 2 por garantia sig 3	

Nota: Elaboración propia

3.2.4 Mitigación de incompatibilidades.

En esta etapa del proyecto, se desarrolla un plan detallado con el objetivo de minimizar los conflictos entre las dependencias, asegurando al mismo tiempo la preservación de las funcionalidades del aplicativo.

Para lograrlo, se llevan a cabo los siguientes cambios:

- La versión mínima requerida del lenguaje de programación “PHP” cambia según cada versión de Laravel.
- La dependencia “barryvdh/laravel-snappy”, que facilita la generación de archivos PDF e imágenes, debe actualizarse a la versión mínima requerida 1.0.2 para garantizar su compatibilidad con Laravel 9 y 10.
- La dependencia “doctrine/dbal”, que proporciona una capa de abstracción de base de datos para interactuar con diversas bases de datos, debe actualizarse a la versión mínima requerida 3.0 para ser compatible con la versión 10 de Laravel.
- La compatibilidad de la dependencia “fideloper/proxy” se ve interrumpida a partir de la versión 9 de Laravel, debido a los cambios en la gestión de los proxies de confianza por parte de Laravel.
- La dependencia “guzzlehttp/guzzle”, que proporciona un cliente HTTP, debe actualizarse para garantizar su compatibilidad a la versión mínima requerida 7.0.1 para Laravel 8 y 7.2 para Laravel 9 y 10.

- La dependencia “laravel/framework”, que constituye el núcleo del framework, debe actualizarse para cada versión, y esta proporciona información de la versión actual en uso.
- La dependencia “laravel/tinker”, que ofrece una consola interactiva para explorar y probar el código de la aplicación, debe actualizarse para garantizar su compatibilidad a la versión mínima requerida 2.0 para Laravel 7 y 8, y 2.7 para Laravel 9 y 10.
- La dependencia “league/flysystem-aws-s3-v3”, que posibilita la interacción con el servicio de almacenamiento en la nube Amazon S3, debe actualizarse a la versión mínima requerida 3.0 para garantizar su compatibilidad con Laravel 9 y 10.
- La dependencia “mpociot/laravel-apidoc-generator”, una herramienta para generar automáticamente documentación de APIs, deja de ser compatible a partir de la versión 7 de Laravel en adelante.
- La dependencia “spatie/laravel-cors”, encargada de gestionar las políticas CORS, deja de ser compatible a partir de la versión 7 de Laravel en adelante.
- La dependencia “tymon/jwt-auth”, que proporciona una forma de implementar la autenticación mediante JSON Web Token, debe actualizarse a la versión mínima requerida 2.0 para garantizar su compatibilidad con Laravel 10.
- La dependencia “fzaninotto/faker”, que cumple con la función de generar de datos falsos, deja de ser compatible con la versión 10 de Laravel.
- La dependencia “nunomaduro/collision”, que mejora la salida de errores y excepciones en la línea de comandos, debe actualizarse para garantizar su compatibilidad a la versión mínima requerida 4.1 para Laravel 7, 5.0 para Laravel 8, 6.1 para Laravel 9 y 7.0 para Laravel 10.

- La dependencia “phpunit/phpunit”, un framework de pruebas unitarias, debe actualizarse para garantizar su compatibilidad a la versión mínima requerida 8.5 para Laravel 7, 9.0 para Laravel 8 y 9, y 10.0 para Laravel 10.
- La dependencia “laravel/helpers”, que ofrece un conjunto de funciones auxiliares globales para simplificar tareas e interacciones comunes, debe implementarse a partir de la versión 6.0 de Laravel en adelante, debido a la utilización de las funciones Arr y Str.
- La dependencia “facade/ignition”, que ofrece una interfaz visual amigable para depurar errores, se implementa a partir de Laravel 7 con una versión mínima requerida de 2.0. Su compatibilidad está disponible hasta Laravel 8, con una versión mínima requerida 2.3.6.
- La dependencia “fruitcake/laravel-cors”, que permite agregar soporte para encabezados CORS y reemplaza a “spatie/laravel-cors”, se implementa en Laravel 7 y 8. Sin embargo, se considera obsoleta después debido a la inclusión de CORS en el núcleo de Laravel a partir de la versión 9.
- La dependencia “laravel/ui”, que ofrece un sistema de andamiaje para generar interfaces, se implementa en la versión mínima requerida 2.0 para Laravel 7, 3.0 para Laravel 8 y 9, y 4.0 para Laravel 10.
- La dependencia “spatie/laravel-ignition”, que mejora las páginas de error predeterminadas con información detallada y herramientas de depuración útiles, se implementa en la versión mínima requerida 1.0 para Laravel 9 y 2.0 para Laravel 10.
- La dependencia “symfony/http-client” es una herramienta que facilita la realización de solicitudes HTTP de forma sencilla y eficiente en aplicaciones PHP. Se implementa en la versión mínima requerida 6.0 a partir de Laravel 9.

- La dependencia “symfony/mailgun-mailer”, que simplifica el envío de correos electrónicos mediante Mailgun, se implementa en la versión mínima requerida 6.0 a partir de Laravel 9.
- La dependencia “symfony/postmark-mailer” permite enviar correos electrónicos de manera fácil y confiable a través de Postmark. Se implementa en la versión mínima requerida 6.0 a partir de Laravel 9.
- La dependencia “laravel/sanctum”, un controlador de autenticación de Laravel diseñado específicamente para aplicaciones de una sola página, se implementa en la versión mínima requerida 3.2 para Laravel 10.
- La dependencia “fakerphp/faker”, una herramienta para generar datos falsos de forma sencilla y eficiente en PHP y reemplazo de la dependencia “fzaninotto/faker”, se implementa en la versión mínima requerida 1.23 para Laravel 10.

Figura 36*Versiones de dependencias por versión de laravel*

Dependencia	Version de Laravel					
	Laravel 5.8	Laravel 6	Laravel 7	Laravel 8	Laravel 9	Laravel 10
php	^7.1.3	^7.2	^7.2.5	^7.3.0	^8.0.2	^8.1.0
barryvdh/laravel-snappy	^0.4.6	^0.4.6	^0.4.6	^0.4.6	^1.0.2	^1.0.2
doctrine/dbal	^2.10	^2.10	^2.10	^2.10	^2.10	^3.0
fideloper/proxy	^4.0	^4.0	^4.0	^4.0	No soportado	No soportado
guzzlehttp/guzzle	^6.4	^6.4	^6.4	^7.0.1	^7.2	^7.2
h4cc/wkhtmltoimage-amd64	^0.12.x	^0.12.x	^0.12.x	^0.12.x	^0.12.x	^0.12.x
h4cc/wkhtmltopdf-amd64	^0.12.x	^0.12.x	^0.12.x	^0.12.x	^0.12.x	^0.12.x
laravel/framework	5.8.*	^6.0	^7.0	^8.0	^9.0	^10.0
laravel/tinker	^1.0	^1.0	^2.0	^2.0	^2.7	^2.7
league/flysystem-aws-s3-v3	^1.0	^1.0	^1.0	^1.0	^3.0	^3.0
maatwebsite/Excel	^3.1	^3.1	^3.1	^3.1	^3.1	^3.1
mpociot/laravel-apidoc-generator	^3.10	^3.10	No soportado	No soportado	No soportado	No soportado
spatie/laravel-cors	^1.3	^1.3	No soportado	No soportado	No soportado	No soportado
squizlabs/php_codesniffer	*	*	*	*	*	*
tymon/jwt-auth	~1.0	~1.0	~1.0	~1.0	~1.0	^2.0
beyondcode/laravel-dump-server	^1.0	^1.0	^1.0	^1.0	^1.0	^1.0
filp/whoops	^2.0	^2.0	^2.0	^2.0	^2.0	^2.0
fzaninotto/faker	^1.4	^1.4	^1.4	^1.4	^1.4	No soportado
mockery/mockery	^1.0	^1.0	^1.0	^1.0	^1.0	^1.0
nunomaduro/collision	^3.0	^3.0	^4.1	^5.0	^6.1	^7.0
phpunit/phpunit	^7.5	^7.5	^8.5	^9.0	^9.0	^10.0
laravel/helpers	No necesario	^1.7	^1.7	^1.7	^1.7	^1.7
nesbot/carbon	^2.71.10 ^3.0.0	^2.71.10 ^3.0.0	^2.71.10 ^3.0.0	^2.71.10 ^3.0.0	^2.71.10 ^3.0.0	^2.71.10 ^3.0.0
facade/ignition	No necesario	No necesario	^2.0	^2.3.6	No soportado	No soportado
fruitcake/laravel-cors	No necesario	No necesario	^2.2	^2.2	No soportado	No soportado
laravel/ui	No necesario	No necesario	^2.0	^3.0	^3.0	^4.4
spatie/laravel-ignition	No necesario	No necesario	No necesario	No necesario	^1.0	^2.0
symfony/http-client	No necesario	No necesario	No necesario	No necesario	^6.0	^6.0
symfony/mailgun-mailer	No necesario	No necesario	No necesario	No necesario	^6.0	^6.0
symfony/postmark-mailer	No necesario	No necesario	No necesario	No necesario	^6.0	^6.0
laravel/sanctum	No necesario	No necesario	No necesario	No necesario	No necesario	^3.2
fakerphp/faker	No necesario	No necesario	No necesario	No necesario	No necesario	^1.23

Nota: Elaboración propia

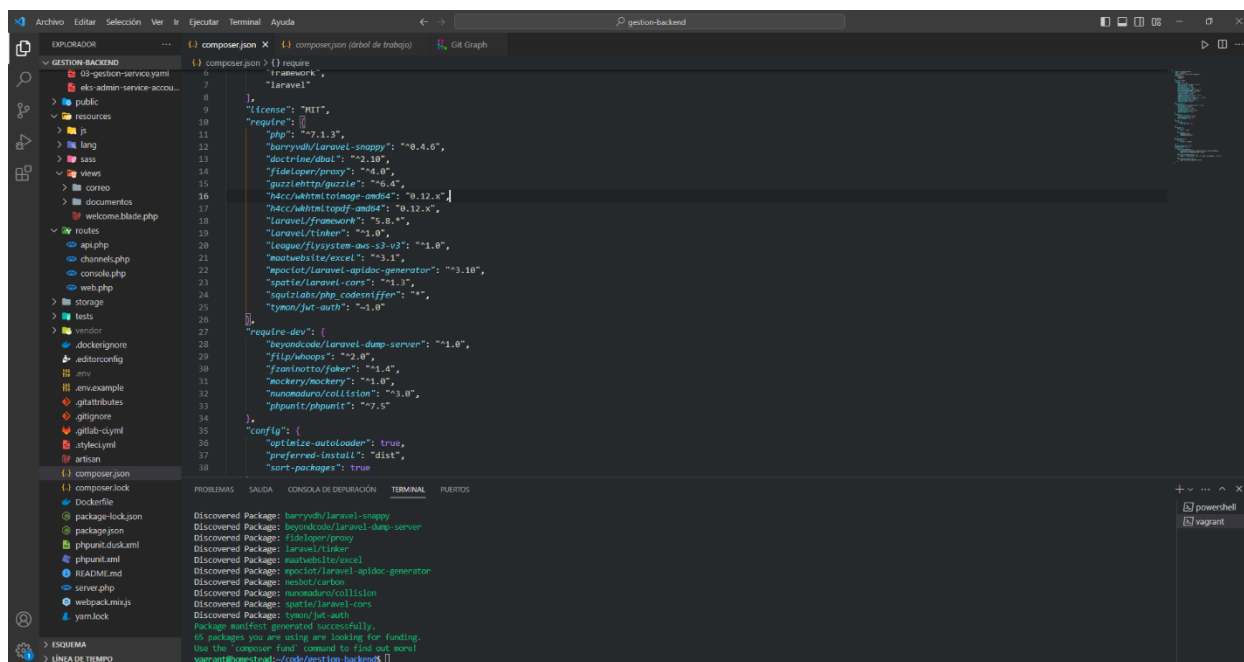
3.3 Efectuar la migración, garantizando el funcionamiento y mitigando cualquier impacto en la funcionalidad.

3.3.1 Migración del framework.

En esta etapa para actualizar el framework y dependencias, en el archivo composer.json, se deben modificar las versiones de las dependencias según corresponda con la versión a la que se está actualizando el framework.

Figura 37

Dependencias iniciales del aplicativo Gestión v1



Nota: Elaboración propia

Laravel 6. Para migrar el aplicativo de laravel 5.8 a laravel 6, se administran las dependencias que se detallan en la Tabla 3 a continuación.

Tabla 3

Dependencias modificadas o agregadas para laravel 6.

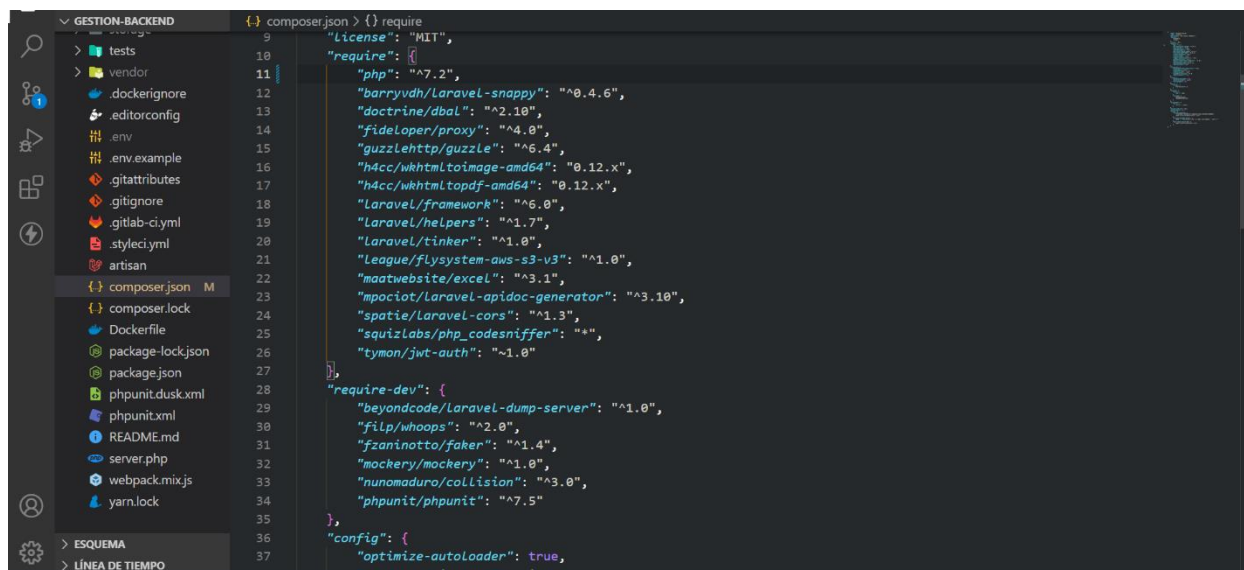
Dependencia	Versión	Acción
php	^7.2	Actualizada
laravel/framework	^6.0	Actualizada
laravel/helpers	^1.7	Agregada

Nota: Elaboración Propia

Quedando con el siguiente resultado en el archivo composer.json.

Figura 38

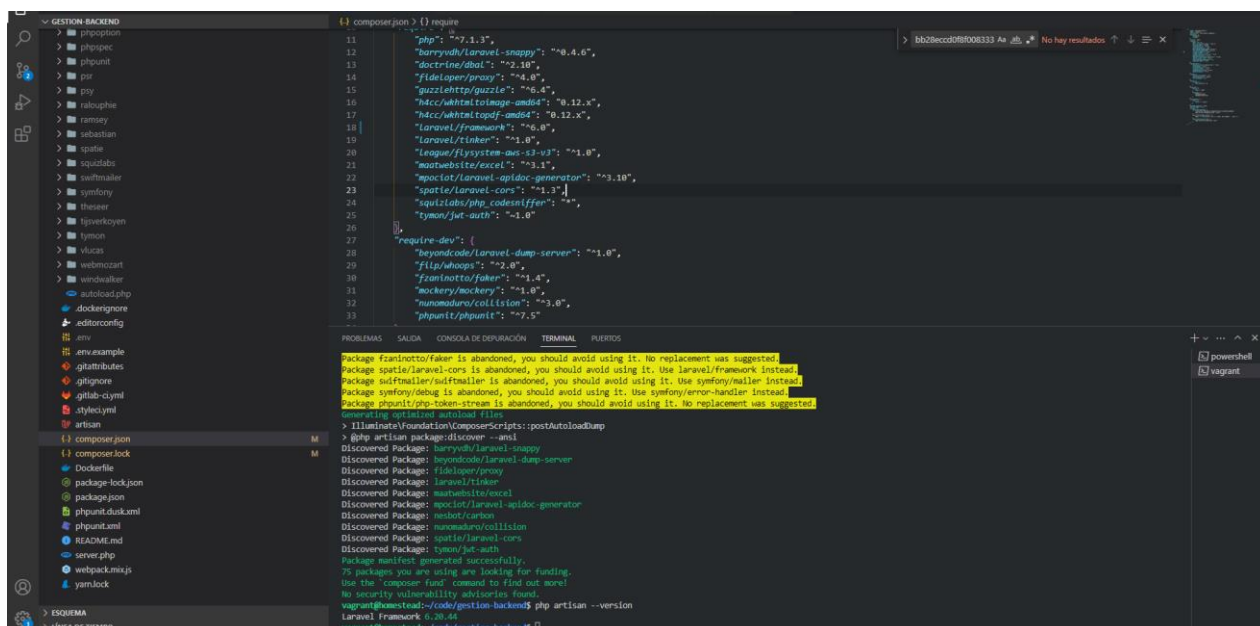
Administración dependencias para laravel 6.



Nota: Elaboración propia

Figura 39

Versión de laravel 6 expuesto por consola.



Nota: Elaboración propia.

Laravel 7. Para migrar el aplicativo de laravel 6 a laravel 7, se administran las dependencias que se detallan en la Tabla 4 a continuación.

Tabla 4

Administración dependencias para laravel 7.

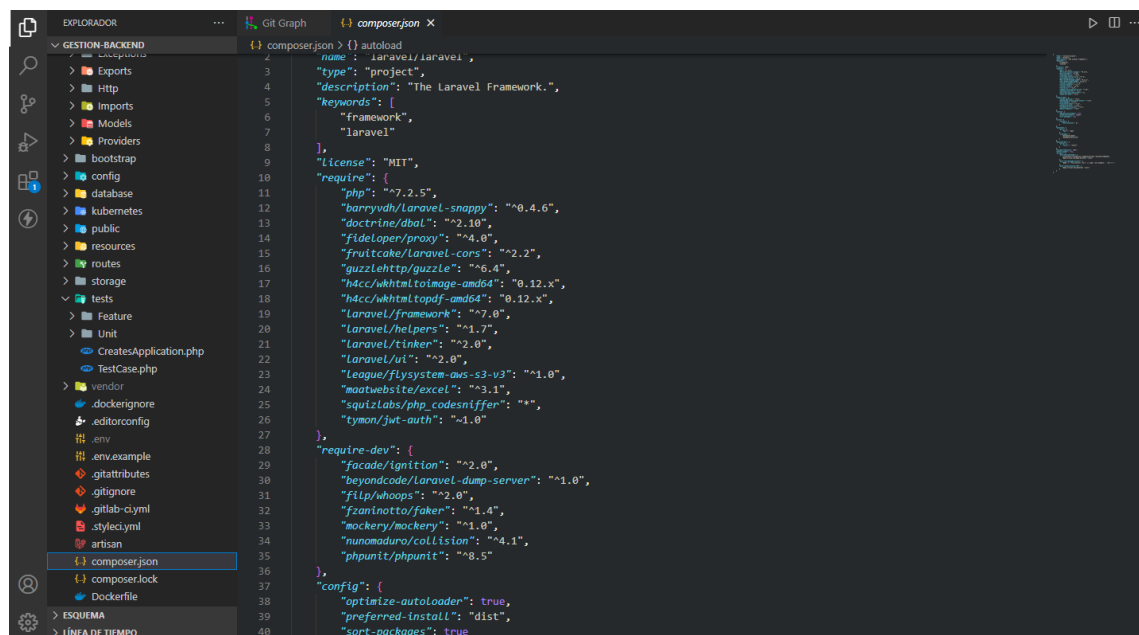
Dependencia	Versión	Acción
php	^7.2.5	Actualizada
laravel/framework	^7.0	Actualizada
laravel/tinker	^2.0	Actualizada
mpociot/laravel-apidoc-generator	^3.10	Eliminada
spatie/laravel-cors	^1.3	Eliminada
nunomaduro/collision	^4.1	Actualizada
phpunit/phpunit	^8.5	Actualizada
facade/ignition	^2.0	Agregada
fruitcake/laravel-cors	^2.2	Agregada
laravel/ui	^2.0	Agregada

Nota: Elaboración Propia

Quedando con el siguiente resultado en el archivo composer.json.

Figura 40

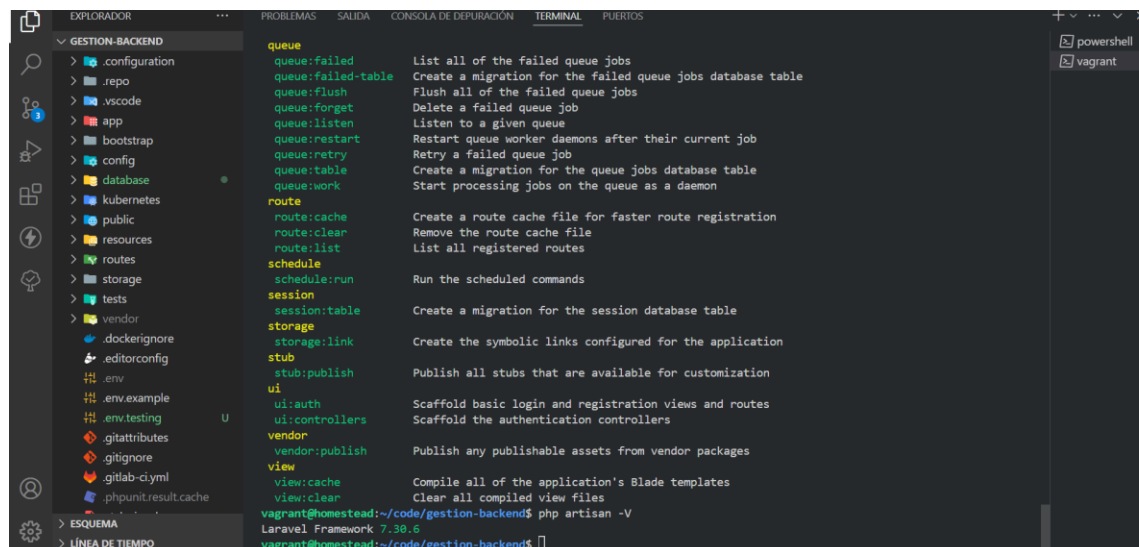
Administración dependencias para laravel 7.



Nota: Elaboración propia

Figura 41

Versión de laravel 7 expuesto por consola.



Nota: Elaboración propia

Laravel 8. Para migrar el aplicativo de laravel 7 a laravel 8, se administran las dependencias que se detallan en la Tabla 5 a continuación.

Tabla 5

Administración dependencias para laravel 8.

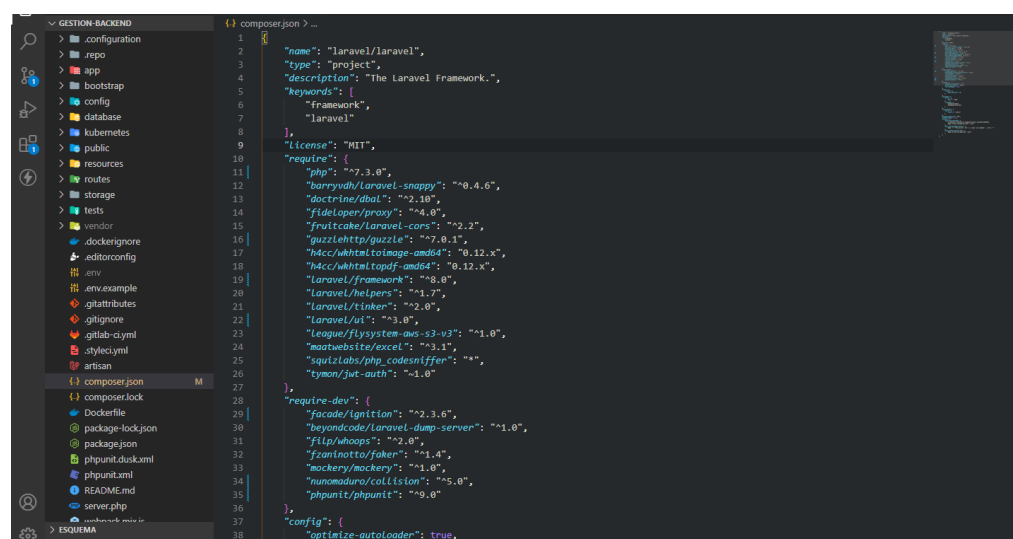
Dependencia	Versión	Acción
php	^7.3.0	Actualizada
guzzlehttp/guzzle	^7.0.1	Actualizada
laravel/framework	^8.0	Actualizada
nunomaduro/collision	^5.0	Actualizada
phpunit/phpunit	^9.0	Actualizada
facade/ignition	^2.3.6	Actualizada
laravel/ui	^3.0	Actualizada

Nota: Elaboración Propia

Quedando con el siguiente resultado en el archivo composer.json.

Figura 42

Administración dependencias para laravel 8.



Nota: Elaboración propia

Figura 43

Versión de laravel 8 expuesto por consola.

```

- Upgrading sebastian/type (1.1.4 => 3.2.1): Extracting archive
- Upgrading sebastian/resource-operations (2.0.2 => 3.0.3): Extracting archive
- Upgrading sebastian/recursion-context (3.0.1 => 4.0.5): Extracting archive
- Upgrading sebastian/object-reflector (1.1.2 => 2.0.4): Extracting archive
- Upgrading sebastian/object-enumerator (3.0.4 => 4.0.4): Extracting archive
- Upgrading sebastian/global-state (3.0.3 => 5.0.6): Extracting archive
- Upgrading sebastian/exporter (3.1.5 => 4.0.5): Extracting archive
- Upgrading sebastian/environment (4.2.4 => 5.1.5): Extracting archive
- Upgrading sebastian/diff (3.0.4 => 4.0.5): Extracting archive
- Upgrading sebastian/comparator (3.0.5 => 4.0.8): Extracting archive
- Installing sebastian/code-unit (1.0.0): Extracting archive
- Installing sebastian/cli-parser (1.0.1): Extracting archive
- Upgrading phpunit/php-timer (2.1.3 => 5.0.3): Extracting archive
- Upgrading phpunit/php-text-template (1.2.1 => 2.0.4): Extracting archive
- Installing phpunit/php-invoker (3.1.1): Extracting archive
- Upgrading phpunit/php-file-iterator (2.0.5 => 3.0.6): Extracting archive
- Installing sebastian/lines-of-code (1.0.4): Extracting archive
- Installing sebastian/complexity (2.0.3): Extracting archive
- Upgrading sebastian/code-unit-reverse-lookup (1.0.2 => 2.0.3): Extracting archive
- Upgrading phpunit/php-code-coverage (7.0.15 => 9.2.30): Extracting archive
- Upgrading phpunit/phpunit (8.5.36 => 9.6.16): Extracting archive
Package fruitcake/laravel-cors is abandoned, you should avoid using it. No replacement was suggested.
Package swiftmailer/swiftmailer is abandoned, you should avoid using it. Use symfony/mailer instead.
Package fzaninotto/faker is abandoned, you should avoid using it. No replacement was suggested.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: barryvdh/laravel-snappy
Discovered Package: beyondcode/laravel-dump-server
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: fruitcake/laravel-cors
Discovered Package: laravel/tinker
Discovered Package: laravel/ui
Discovered Package: maatwebsite/excel
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Discovered Package: tymon/jwt-auth
Package manifest generated successfully.
89 packages you are using are looking for funding.
Use the 'composer fund' command to find out more!
No security vulnerability advisories found.
vagrant@homestead:~/code/gestion-backend$ php artisan --version
Laravel Framework 8.33.2

```

Nota: Elaboración propia

Laravel 9. Para migrar el aplicativo de laravel 8 a laravel 9, se administran las dependencias que se detallan en la Tabla 6 a continuación.

Tabla 6

Administración dependencias para laravel 9.

Dependencia	Versión	Acción
php	^8.0.2	Actualizada
barryvdh/laravel-snappy	^1.0.2	Actualizada
fideloper/proxy	^4.0	Eliminada
guzzlehttp/guzzle	^7.2	Actualizada
laravel/framework	^9.0	Actualizada
laravel/tinker	^2.7	Actualizada
league/flysystem-aws-s3-v3	^3.0	Actualizada
nunomaduro/collision	^6.1	Actualizada
facade/ignition	^2.3.6	Eliminada

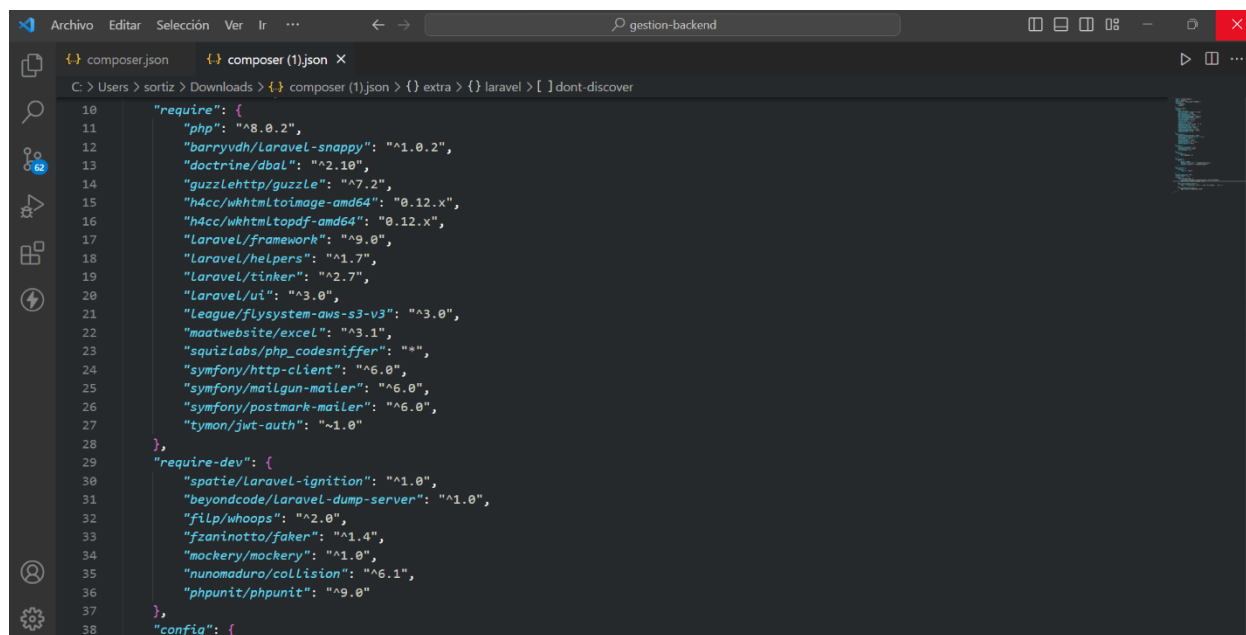
fruitcake/laravel-cors	^2.2	Eliminada
spatie/laravel-ignition	^1.0	Agregada
symfony/http-client	^6.0	Agregada
symfony/mailgun-mailer	^6.0	Agregada
symfony/postmark-mailer	^6.0	Agregada

Nota: Elaboración Propia

Quedando con el siguiente resultado en el archivo composer.json.

Figura 44

Administración dependencias para laravel 9.



```

10  "require": {
11      "php": "^8.0.2",
12      "barryvdh/laravel-snappy": "^1.0.2",
13      "doctrine/dbal": "^2.10",
14      "guzzlehttp/guzzle": "^7.2",
15      "h4cc/wkhtmltoimage-amd64": "0.12.x",
16      "h4cc/wkhtmltopdf-amd64": "0.12.x",
17      "Laravel/framework": "^9.0",
18      "Laravel/helpers": "^1.7",
19      "Laravel/tinker": "^2.7",
20      "Laravel/ui": "^3.0",
21      "League/flysystem-aws-s3-v3": "^3.0",
22      "maatwebsite/excel": "^3.1",
23      "squizlabs/php_codesniffer": "*",
24      "symfony/http-client": "^6.0",
25      "symfony/mailgun-mailer": "^6.0",
26      "symfony/postmark-mailer": "^6.0",
27      "tymon/jwt-auth": "~1.0"
28  },
29  "require-dev": {
30      "spatie/laravel-ignition": "^1.0",
31      "beyondcode/laravel-dump-server": "^1.0",
32      "filp/whoops": "^2.0",
33      "fzaninotto/faker": "^1.4",
34      "mockery/mockery": "^1.0",
35      "nunomaduro/collision": "^6.1",
36      "phpunit/phpunit": "^9.0"
37  },
38  "config": {

```

Nota: Elaboración propia

Figura 45

Actualización de dependencias para laravel 9.

```

no changes added to commit (use "git add" and/or "git commit -a")
vagrant@homestead:~/code/gestion-backend$ composer update
Loading composer repositories with package information
Updating dependencies
Lock file operations: 0 installs, 0 updates, 2 removals
  - Removing asm89/stack-cors (v2.2.0)
  - Removing fruitcake/laravel-cors (v2.2.0)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 0 installs, 0 updates, 2 removals
  - Removing fruitcake/laravel-cors (v2.2.0)
  - Removing asm89/stack-cors (v2.2.0)
Package fzaninotto/faker is abandoned, you should avoid using it. No replacement was suggested.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.

barryvdh/laravel-snappy ..... DONE
beyondcode/laravel-dump-server ..... DONE
fideloper/proxy ..... DONE
laravel/tinker ..... DONE
laravel/ui ..... DONE
maatwebsite/excel ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE
tymon/jwt-auth ..... DONE

98 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
No security vulnerability advisories found.
vagrant@homestead:~/code/gestion-backend$

```

Nota: Elaboración propia

Figura 46

Versión de laravel 9 expuesto por consola.

```

- Upgrading lcobucci/jwt (3.3.3 => 4.3.0): Extracting archive
- Upgrading tymon/jwt-auth (1.0.2 => dev-develop 014be8d): Extracting archive
Package fzaninotto/faker is abandoned, you should avoid using it. No replacement was suggested.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.

barryvdh/laravel-snappy ..... DONE
beyondcode/laravel-dump-server ..... DONE
laravel/tinker ..... DONE
laravel/ui ..... DONE
maatwebsite/excel ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE
tymon/jwt-auth ..... DONE

100 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
No security vulnerability advisories found.
vagrant@homestead:~/code/gestion-backend$ php artisan -V
Laravel Framework 9.52.16
vagrant@homestead:~/code/gestion-backend$

```

Nota: Elaboración propia

Laravel 10. Para migrar el aplicativo de laravel 9 a laravel 10, se administran las dependencias que se detallan en la Tabla 7 a continuación.

Tabla 7

Administración dependencias para laravel 10.

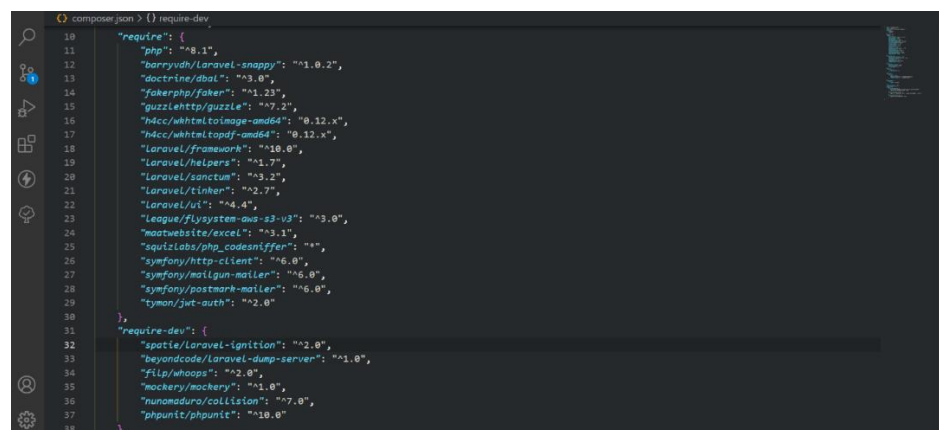
Dependencia	Versión	Acción
php	^8.1	Actualizada
doctrine/dbal	^3.0	Actualizada
laravel/framework	^10.0	Actualizada
tymon/jwt-auth	^2.0	Actualizada
fzaninotto/faker	^3.0	Eliminada
nunomaduro/collision	^7.0	Actualizada
phpunit/phpunit	^10.0	Actualizada
laravel/ui	^4.4	Actualizada
spatie/laravel-ignition	^2.0	Actualizada
laravel/sanctum	^3.2	Agregada
fakerphp/faker	^1.23	Agregada

Nota: Elaboración Propia

Quedando con el siguiente resultado en el archivo composer.json.

Figura 47

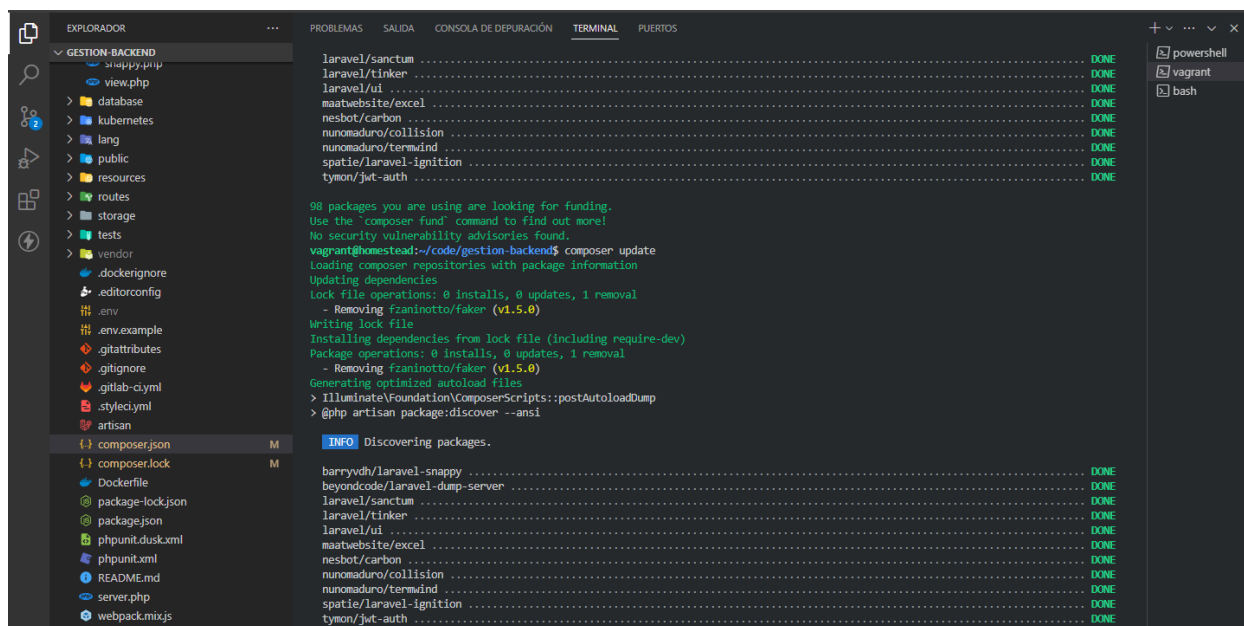
Administración dependencias para laravel 10.



Nota: Elaboración propia

Figura 48

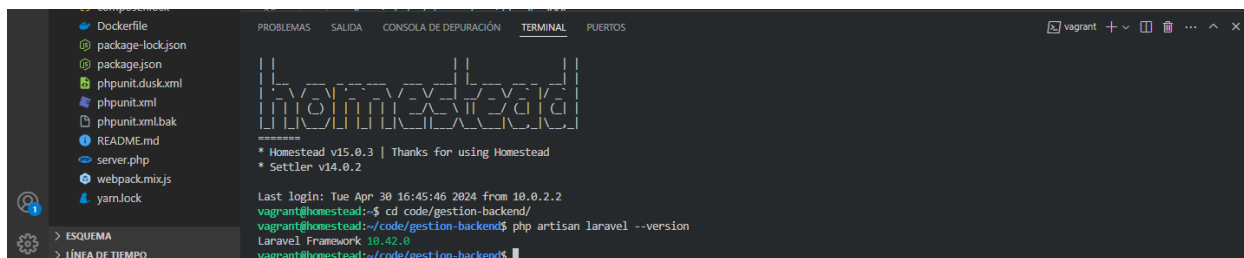
Actualización de dependencias para laravel 10.



Nota: Elaboración propia

Figura 49

Versión de laravel 10 expuesto por consola.



Nota: Elaboración propia

3.2.2 Resolución de errores.

En esta fase, se realizó la corrección de todos los errores generados durante la migración de Laravel. Este proceso se llevó a cabo cada vez que el framework se actualizaba a una nueva versión.

Laravel 7. Durante la transición de Laravel 6 a Laravel 7, surgieron errores en varios metodos dentro de la ruta App\Exceptions\Handler.php, ocasionados por la adopción de Symfony 5 en el framework.

Figura 50

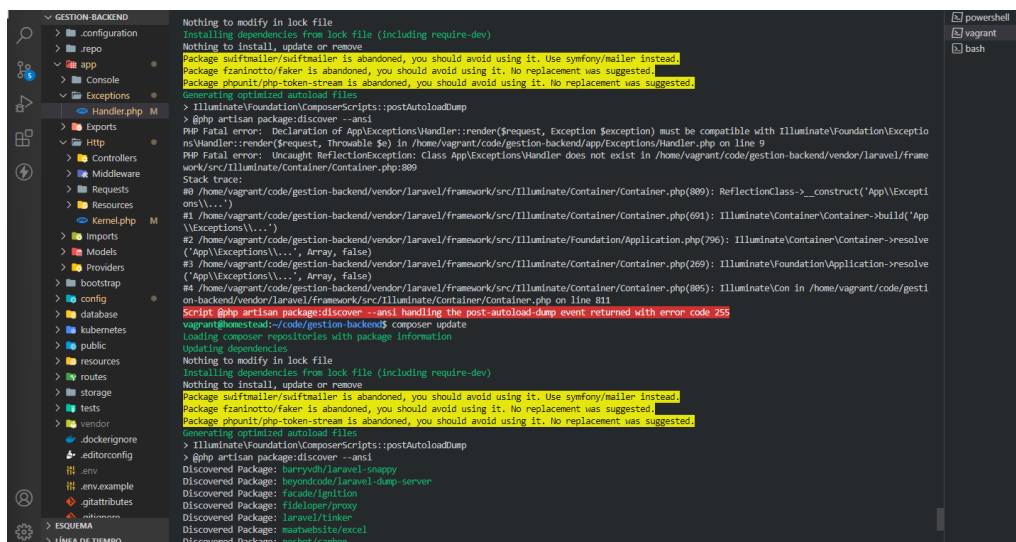
Error de la función report.

```
vagrant@homestead:~/code/gestion-backend$ composer upgrade -W --no-ansi --no-interaction &> /home/vagrant/code/log-upgrade-7.1.txt
vagrant@homestead:~/code/gestion-backend$ composer update
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Package swiftmailer/swiftmailer is abandoned, you should avoid using it. Use symfony/mailer instead.
Package fzaninotto/faker is abandoned, you should avoid using it. No replacement was suggested.
Package phpunit/php-token-stream is abandoned, you should avoid using it. No replacement was suggested.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
PHP Fatal error: Declaration of App\Exceptions\Handler::report(Exception $exception) must be compatible with Illuminate\Foundation\Exceptions\Handler::report(Throwable $e) in /home/vagrant/code/gestion-backend/app/Exceptions/Handler.php on line 8
PHP Fatal error: Uncaught ReflectionException: Class App\Exceptions\Handler does not exist in /home/vagrant/code/gestion-backend/vendor/laravel/framework/src/Illuminate/Container/Container.php:889
Stack trace:
#0 /home/vagrant/code/gestion-backend/vendor/laravel/framework/src/Illuminate/Container/Container.php(889): ReflectionClass->__construct('App\\Exceptions\\...')
#1 /home/vagrant/code/gestion-backend/vendor/laravel/framework/src/Illuminate/Container/Container.php(691): Illuminate\Container\Container->build('App\\Exceptions\\...')
#2 /home/vagrant/code/gestion-backend/vendor/laravel/framework/src/Illuminate/Foundation/Application.php(796): Illuminate\Container\Container->resolve('App\\Exceptions\\...', Array, false)
#3 /home/vagrant/code/gestion-backend/vendor/laravel/framework/src/Illuminate/Container/Container.php(269): Illuminate\Foundation\Application->resolve('App\\Exceptions\\...', Array, false)
#4 /home/vagrant/code/gestion-backend/vendor/laravel/framework/src/Illuminate/Container/Container.php(885): Illuminate\\Con in /home/vagrant/code/gesti
on-backend/vendor/laravel/framework/src/Illuminate/Container/Container.php on line 811
Script @php artisan package:discover --ansi handling the post-autoload-dump event returned with error code 255
vagrant@homestead:~/code/gestion-backend$ composer update
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Package swiftmailer/swiftmailer is abandoned, you should avoid using it. Use symfony/mailer instead.
Package fzaninotto/faker is abandoned, you should avoid using it. No replacement was suggested.
Package phpunit/php-token-stream is abandoned, you should avoid using it. No replacement was suggested.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
```

Nota: Elaboración propia

Figura 51

Error de la función render.

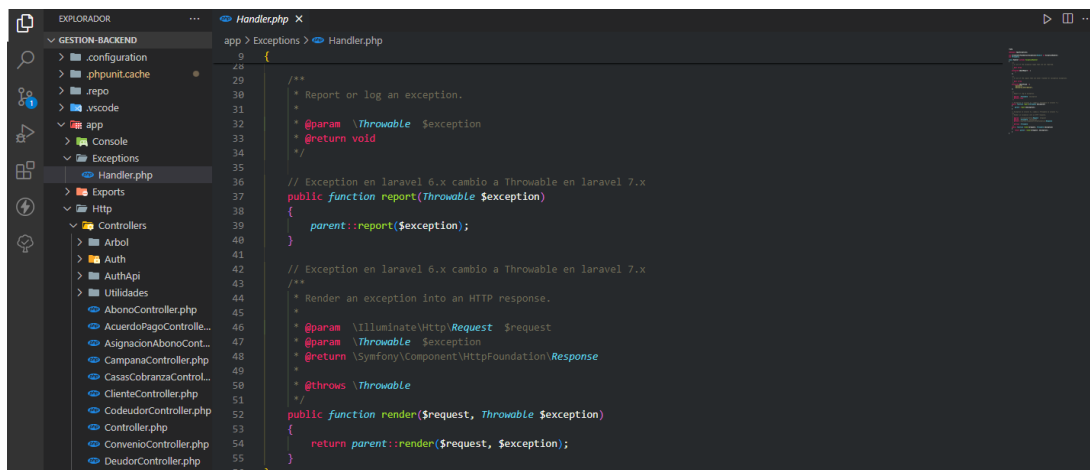


Nota: Elaboración propia

Para corregir este error, se procede a modificar el archivo en la ruta específica, cambiando el parámetro que recibe de "Exception" a "Throwable".

Figura 52

Cambios al archivo Handler.php.



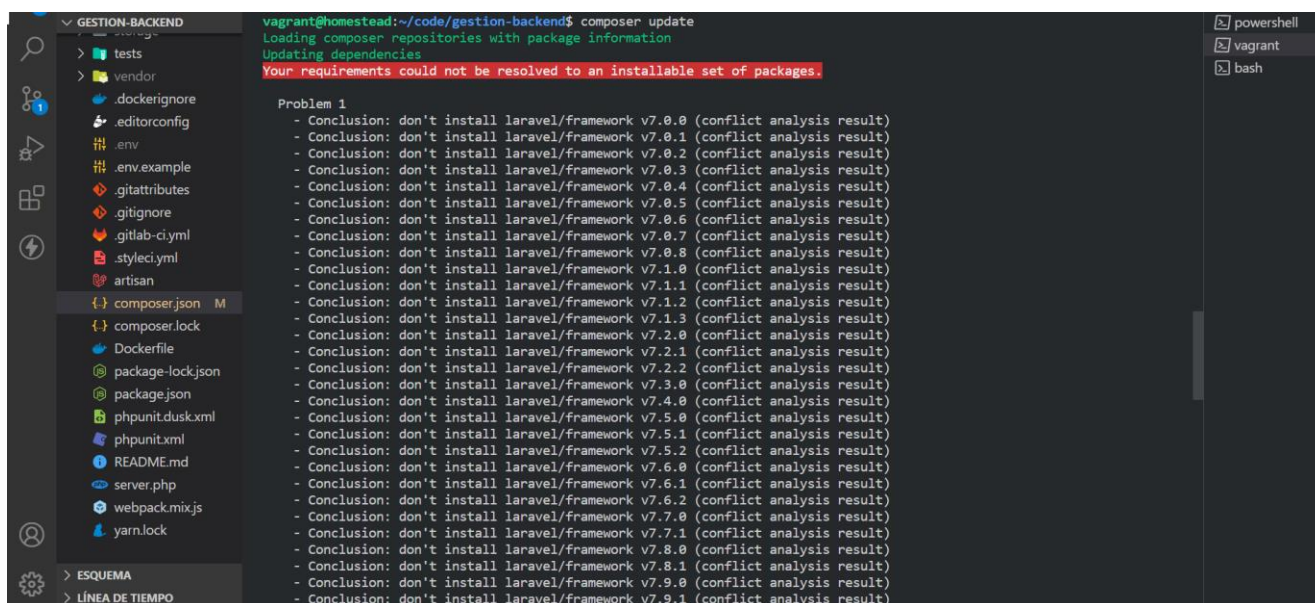
Nota: Elaboración propia

Durante la migración, también se evidenciaron los problemas generados por las dependencias que no eran compatibles.

Uno de estos casos es la dependencia "spatie/laravel-cors", que no es compatible con el framework y, por lo tanto, es sustituida por "fruitcake/laravel-cors".

Figura 53

Incompatibilidad con la dependencia spatie/laravel-cors.



```
vagrant@homestead:~/code/gestion-backend$ composer update
Loading composer repositories with package information
Updating dependencies
Your requirements could not be resolved to an installable set of packages.

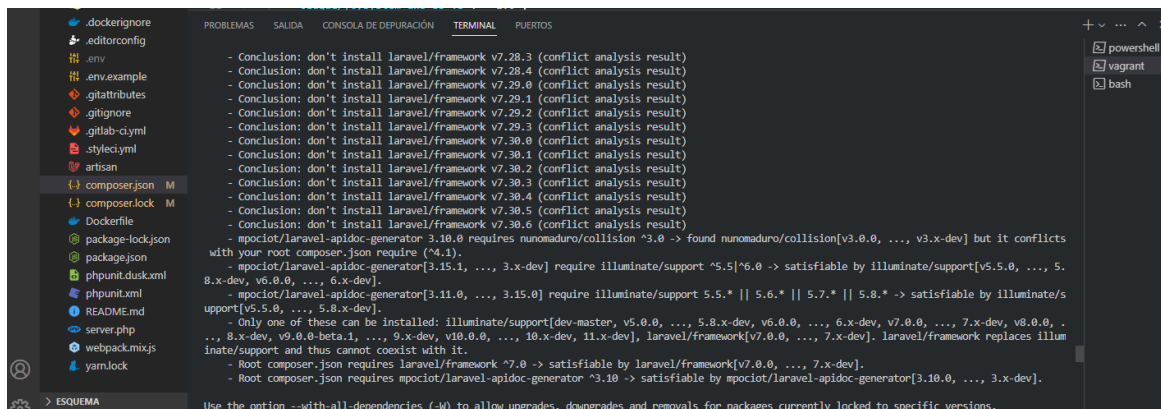
Problem 1
- Conclusion: don't install laravel/framework v7.0.0 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.0.1 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.0.2 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.0.3 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.0.4 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.0.5 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.0.6 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.0.7 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.0.8 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.1.0 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.1.1 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.1.2 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.1.3 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.2.0 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.2.1 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.2.2 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.3.0 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.4.0 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.5.0 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.5.1 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.5.2 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.6.0 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.6.1 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.6.2 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.7.0 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.7.1 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.8.0 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.8.1 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.9.0 (conflict analysis result)
- Conclusion: don't install laravel/framework v7.9.1 (conflict analysis result)
```

Nota: Elaboración propia

Y gracias al análisis de funcionalidades, se encontró que la dependencia "mpociot/laravel-apidoc-generator" la cual no era compatible, no era utilizada, por dicha razón se removió del aplicativo.

Figura 54

Conflicto con la dependencia mpociot/laravel-apidoc-generator.

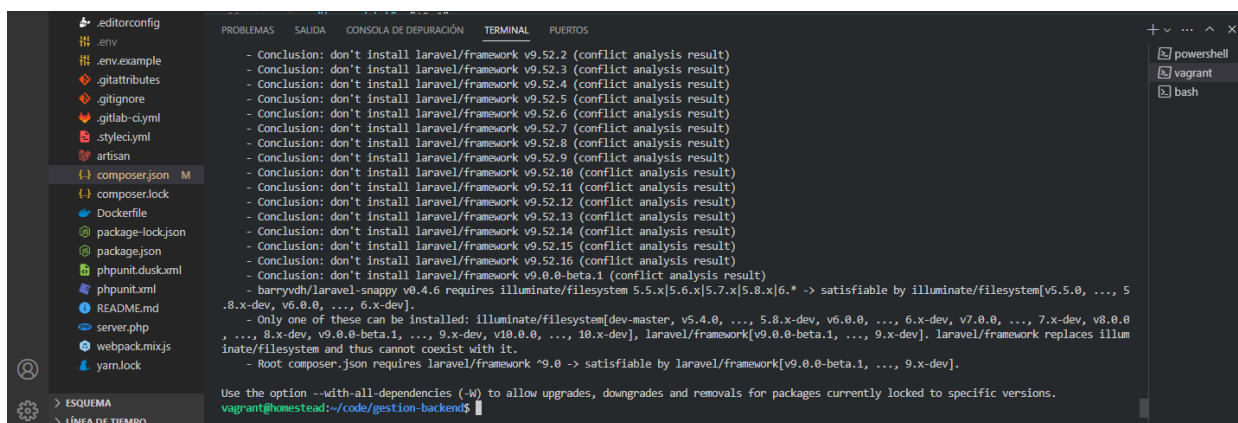


Nota: Elaboración propia

Laravel 9. Durante la transición de Laravel 8 a Laravel 9, surgió conflicto con la dependencia “barryvdh/laravel-snappy” debido al uso de una versión incompatible. Para resolver este problema, se actualizó a una nueva versión compatible.

Figura 55

Conflicto con la dependencia barryvdh/laravel-snappy.

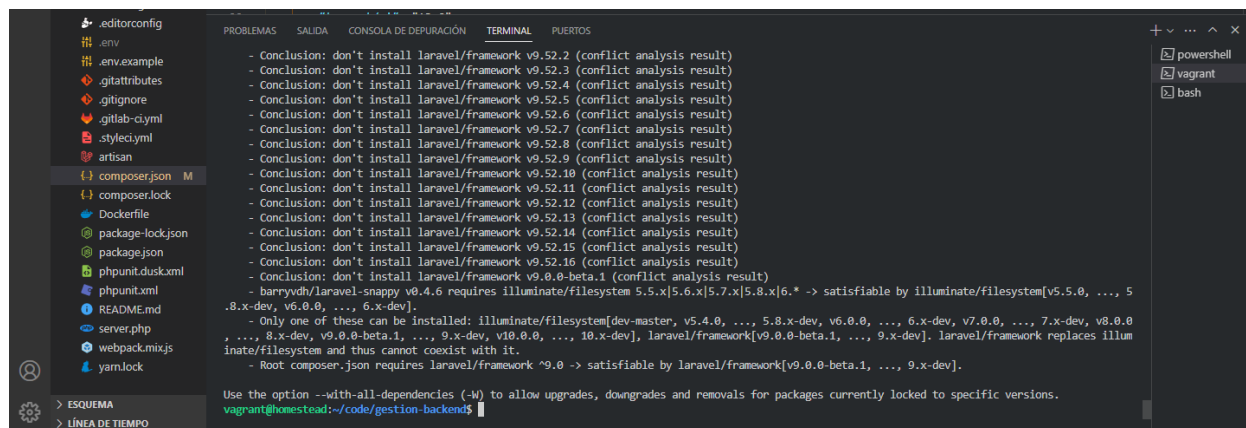


Nota: Elaboración propia

También surgió un conflicto con la dependencia “league/flysystem-aws-s3-v3” debido al uso de una versión incompatible. Para resolver este problema, se eliminó la dependencia mediante comandos y se reinstaló con una versión compatible.

Figura 56

Conflicto con la dependencia league/flysystem-aws-s3-v3.



Nota: Elaboración propia

Después surgieron problemas de incompatibilidad con algunas de ellas. Para abordar esta situación, se ejecutó el comando “composer upgrade -W”.

Figura 57

Problemas con versiones de dependencias.

```

- Conclusion: don't install laravel/framework v9.52.7 (conflict analysis result)
- Conclusion: don't install league/flysystem 3.23.0 (conflict analysis result)
- Conclusion: don't install laravel/framework v9.52.8 (conflict analysis result)
- Conclusion: don't install laravel/framework v9.52.9 (conflict analysis result)
- Conclusion: don't install laravel/framework v9.52.10 (conflict analysis result)
- Conclusion: don't install laravel/framework v9.52.11 (conflict analysis result)
- Conclusion: don't install illuminate/support v9.0.0-beta.3 (conflict analysis result)
- Root composer.json requires league/flysystem-aws-s3-v3 ^1.0 -> satisfiable by league/flysystem-aws-s3-v3[1.0.0, ..., 1.0.x-dev].

Use the option --with-all-dependencies (-W) to allow upgrades, downgrades and removals for packages currently locked to specific versions.
vagrant@homestead:~/code/gestion-backend$ composer require -W league/flysystem-aws-s3-v3 ^3.0
./composer.json has been updated
Running composer update league/flysystem-aws-s3-v3 --with-all-dependencies
Loading composer repositories with package information
Updating dependencies
Your requirements could not be resolved to an installable set of packages.

Problem 1
- Root composer.json requires barryvdh/laravel-snappy ^1.0.2, found barryvdh/laravel-snappy[dev-master, v1.0.2, 1.0.x-dev (alias of dev-master)] but the package is fixed to v0.4.8 (lock file version) by a partial update and that version does not match. Make sure you list it as an argument for the update command.
Problem 2
- Root composer.json requires nunomaduro/collision ^6.1, found nunomaduro/collision[dev-develop, v6.1.0, ..., 6.x-dev (alias of dev-develop)] but the package is fixed to v5.11.0 (lock file version) by a partial update and that version does not match. Make sure you list it as an argument for the update command.
Problem 3
- tymon/jwt-auth is locked to version 1.0.2 and an update of this package was not requested.
- tymon/jwt-auth ^5.5.9|^7.0 -> your php version (8.0.21) does not satisfy that requirement.
Problem 4
- fzaninotto/faker is locked to version v1.9.2 and an update of this package was not requested.
- fzaninotto/faker v1.9.2 requires php ^5.3.3 || ^7.0 -> your php version (8.0.21) does not satisfy that requirement.
Problem 5
- laravel/framework 9.x-dev requires symfony/http-foundation ^6.0 -> found symfony/http-foundation[v6.0.0-BETA1, ..., 6.4.x-dev] but these were not loaded, likely because it conflicts with another require.
- laravel/framework[v9.0.0-beta.1, ..., v9.22.1] require symfony/console ^6.0 -> found symfony/console[v6.0.0-BETA1, ..., 6.4.x-dev] but these were not loaded, likely because it conflicts with another require.
- laravel/framework[v9.23.0, ..., v9.33.0] require symfony/console ^6.0.3 -> found symfony/console[v6.0.3, ..., 6.4.x-dev] but these were not loaded, likely because it conflicts with another require.
- laravel/framework[v9.34.0, ..., v9.52.16] require symfony/console ^6.0.9 -> found symfony/console[v6.0.9, ..., 6.4.x-dev] but these were not loaded, likely because it conflicts with another require.
- Root composer.json requires laravel/framework ^9.0 -> satisfiable by laravel/framework[v9.0.0-beta.1, ..., 9.x-dev].

Installation failed, reverting ./composer.json and ./composer.lock to their original content.

```

Nota: Elaboración propia

Y por último, se reporta un error en el archivo

app/Http/Controllers/Utilidades/CargueMasivoGestion.php, donde se utiliza una expresión que

no es admitida: “a ? b : c ? d : e”. Se recomienda sustituirla por alguna de las siguientes formas:

“(a ? b : c) ? d : e” o “a ? b : (c ? d : e)”.

Figura 58

Error en expresión no admitida.

```

2 package suggestions were added by new dependencies, use 'composer suggest' to see details.
Package fruitcake/laravel-cors is abandoned, you should avoid using it. No replacement was suggested.
Package fzaninotto/faker is abandoned, you should avoid using it. No replacement was suggested.
Generating optimized autoload files
> illuminate/Foundation/ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
PHP Fatal error: Unparenthesized 'a ? b : c ? d : e' is not supported. Use either '(a ? b : c) ? d : e' or 'a ? b : (c ? d : e)' in /home/vagrant/cod
e/gestion-backend/app/Http/Controllers/Utilidades/CargueMasivoGestion.php on line 114

Symfony\Component\ErrorHandler\Error\FatalError

Unparenthesized 'a ? b : c ? d : e' is not supported. Use either '(a ? b : c) ? d : e' or 'a ? b : (c ? d : e)'

at app/Http/Controllers/Utilidades/CargueMasivoGestion.php:114
110
111         DB::beginTransaction();
112         try {
113             $gestion = Gestion::insertGetId([
114                 'sub_tipo_gestion_id' => isset($value["SUBTIPO"]) ? $value["SUBTIPO"] : isset($value["TIPO"]) ? $value["TIPO"] : 0,
115                 'usuario_id' => auth()->guard('api')->user()->id,
116                 'causal_mora_id' => 0,
117                 'observacion' => isset($value["MENSAJE"]) ? $value["MENSAJE"] : null,
118                 'hora_inicio' => Carbon::parse($value["FECHA"])->format('Y-m-d H:i:s'),
119             ]);
120         } catch (Whoops\Exception\ErrorException $e) {
121             //
122         }
123         DB::commit();
124         $gestion_obligacion = GestionObligacion::insert([
125             'obligacion_id' => $obligacion->id,
126             'gestion_id' => $gestion
127         ]);
128         $rows[$key]["LOG"] = "Registro almacenado correctamente. Id: " . $gestion . " ";
129         foreach ($Illuminate\Database\QueryException $e) {
130             //
131         }
132     }
133 }
134
+1 vendor frames
2 [internal]:0
Whoops\Run::handleShutdown()
Script @php artisan package:discover --ansi handling the post-autoload-dump event returned with error code 255

```

Nota: Elaboración propia

Figura 59

Expresión admitida.

```

EXPLORADOR
GESTION-BACKEND
> .configuration
> .repo
> app
> Console
> Exceptions
> Exports
> Http
> Controllers
> Arbol
> Auth
> AuthApi
> Utilidades
  Cargue... M
  CifraGeneralC...
  GestorArchivo...
  GestorInform...
  GraficaContro...
  NotificadorCo...
  AbonoControll...
  AcuerdoPagoC...
  AsignacionAbo...
  CampanaContr...

app > Http > Controllers > Utilidades > CargueMasivoGestion.php > CargueMasivoGestion > insert
184 insert($rows, $obligaciones)
185
186 ws as $key => $value) {
187     ion = $obligaciones->where('nit', $value["NIT"])->where('cedula', $value["CEDULA"])->where('pagare', $value["PAGARE"])->first();
188
189     t($obligacion) {
190
191         beginTransaction();
192         {
193             $gestion = Gestion::insertGetId([
194                 'sub_tipo_gestion_id' => (isset($value["SUBTIPO"]) ? $value["SUBTIPO"] : isset($value["TIPO"]) ? $value["TIPO"] : 0,
195                 'usuario_id' => auth()->guard('api')->user()->id,
196                 'causal_mora_id' => 0,
197                 'observacion' => isset($value["MENSAJE"]) ? $value["MENSAJE"] : null,
198                 'hora_inicio' => Carbon::parse($value["FECHA"])->format('Y-m-d H:i:s'),
199                 'hora_fin' => Carbon::parse($value["FECHA"])->format('Y-m-d H:i:s'),
200                 'origen' => $value["ORIGEN"] ? $value["ORIGEN"] : null,
201             ]);
202         }
203         DB::commit();
204         $gestion_obligacion = GestionObligacion::insert([
205             'obligacion_id' => $obligacion->id,
206             'gestion_id' => $gestion
207         ]);
208         $rows[$key]["LOG"] = "Registro almacenado correctamente. Id: " . $gestion . " ";
209         foreach ($Illuminate\Database\QueryException $e) {
210             //
211         }
212     }
213 }

```

Nota: Elaboración propia

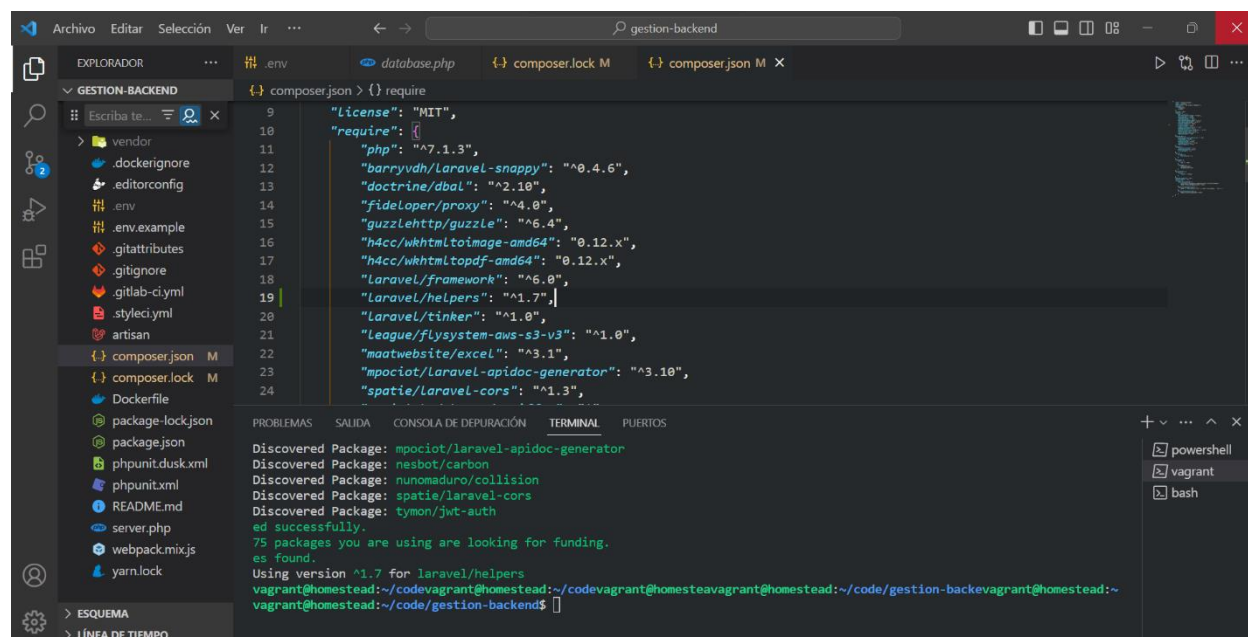
3.3.3 Optimización de funcionalidades.

En esta etapa, se buscó implementar las características proporcionadas por las nuevas versiones del framework y sus dependencias en la funcionalidad del sistema. Para lograrlo, se llevaron a cabo las modificaciones necesarias en los directorios correspondientes para cada uno de ellos, analizando y siguiendo todos los cambios especificados en la documentación oficial de Laravel según la versión especificada.

Laravel 6. Durante la migración de Laravel 5.8 a Laravel 6, se procedió a trasladar los helpers `str_` y `array_` a nuevas ubicaciones. Dado el uso extensivo de estos helpers en la aplicación y la necesidad de mantener sus funcionalidades, se optó por instalar el paquete `laravel/helpers`.

Figura 60

Adición de la dependencia `laravel/helpers`.

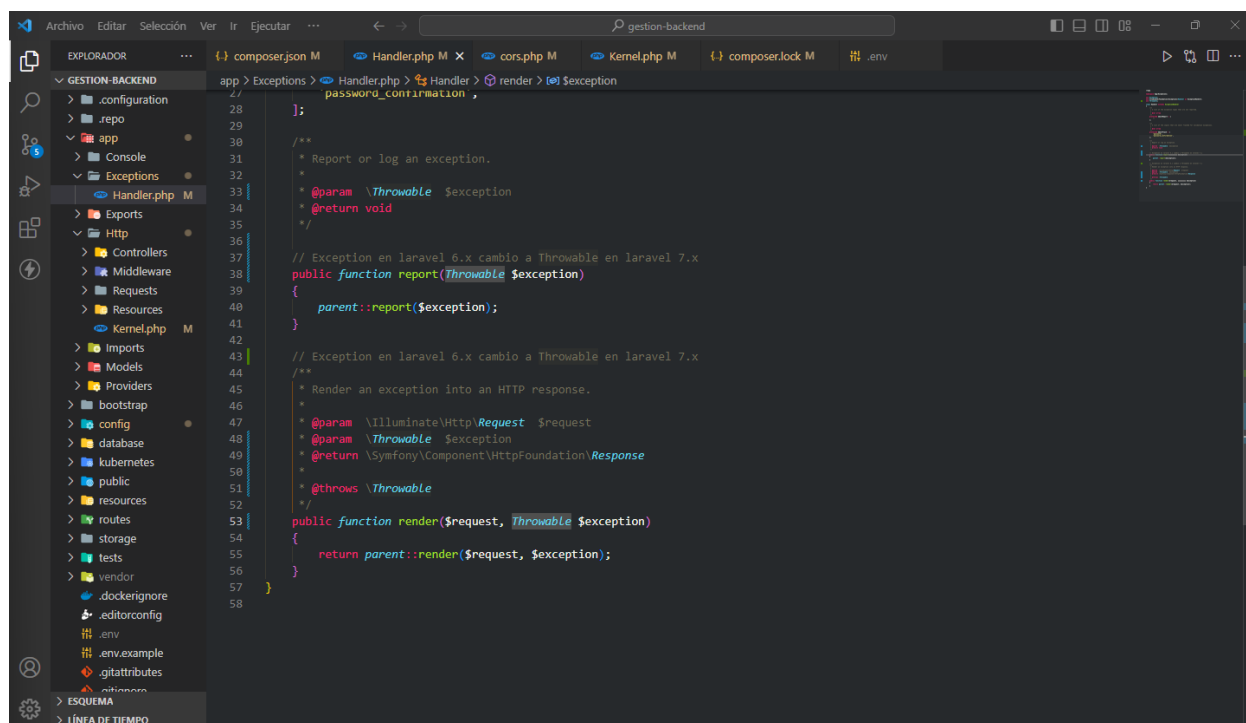


Nota: Elaboración propia

Laravel 7. Durante la migración de Laravel 6 a Laravel 7, en la implementación de Symfony 5, se actualizó el manejo de excepciones en la ruta App\Exceptions\Handler. Se modificaron los métodos report, render y shouldReport de la clase renderForConsole para que ahora acepten instancias de la interfaz Throwable en lugar de instancias de Exception.

Figura 61

Cambios asociados a Symfony 5.

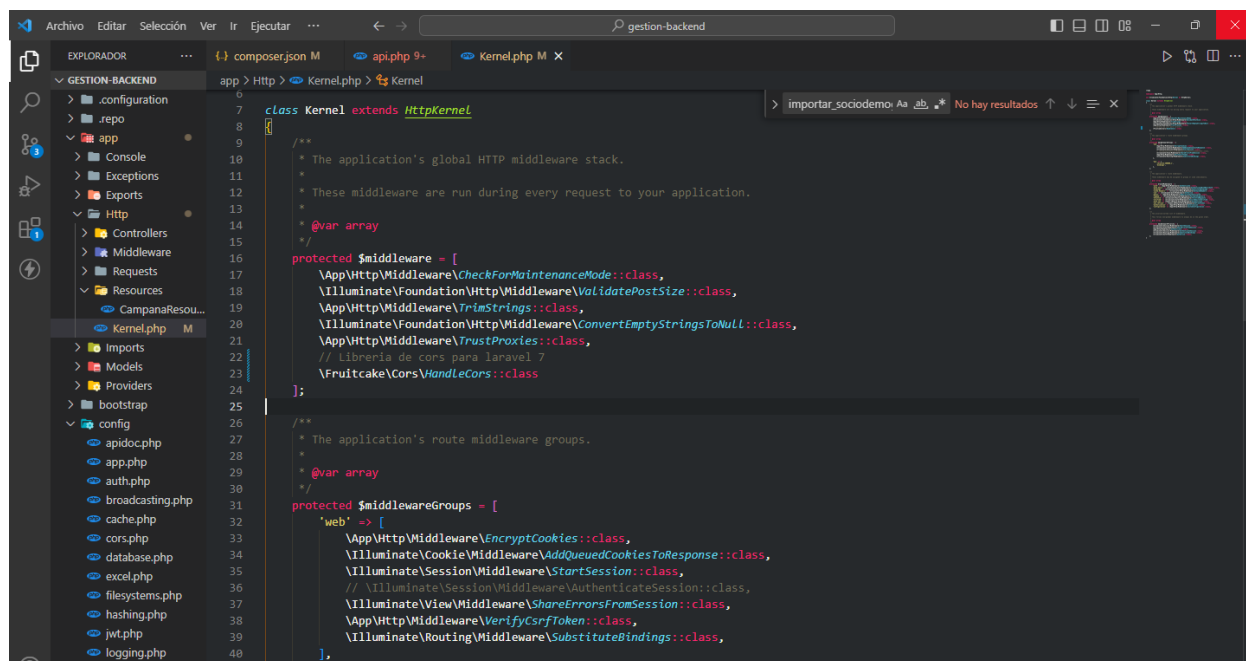


Nota: Elaboración propia

También se implementaron rutas únicas. Durante una revisión, se detectó y corrigió la duplicidad de una de estas rutas para garantizar que no se perdiera su funcionalidad o generara algún error.

Figura 62*Rutas duplicadas.**Nota: Elaboración propia*

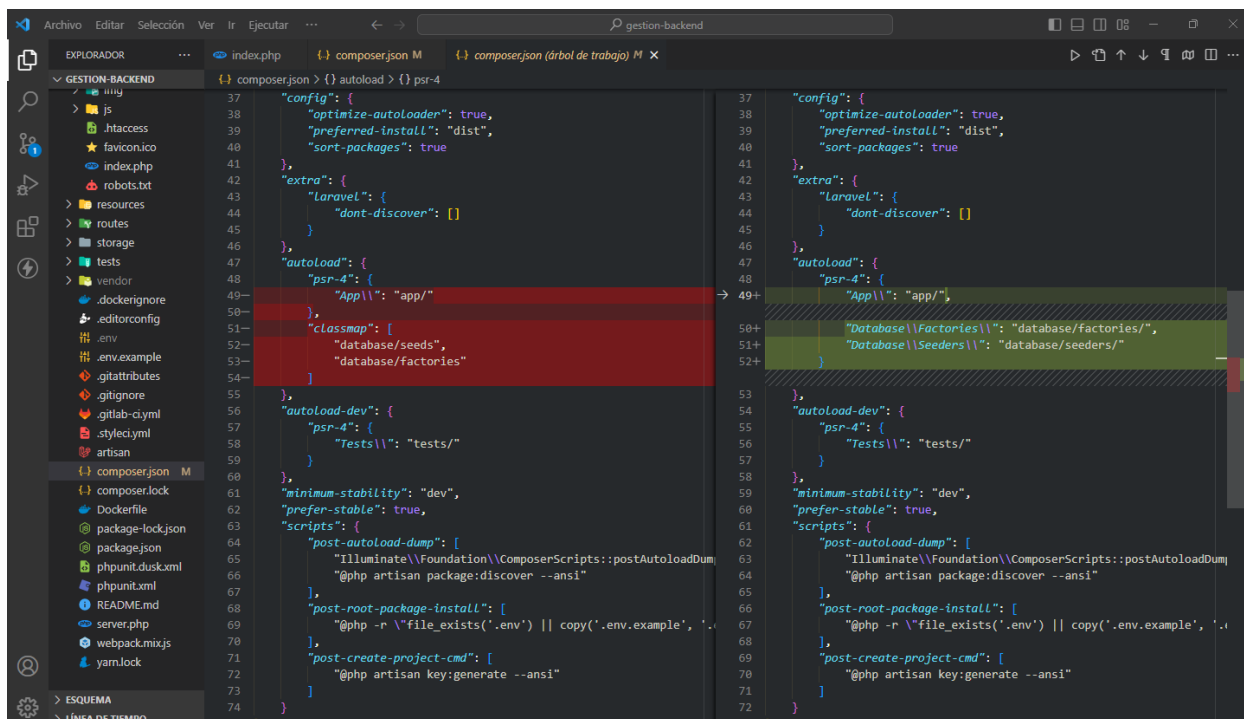
Dado que la dependencia spatie\cors ya no cuenta con soporte, se optó por implementar la dependencia fruitcake/laravel-cors. Esto permite a los navegadores realizar solicitudes desde orígenes diferentes al dominio principal. La dependencia se incluyó en la lista global de middleware del archivo App\Http\Kernel.

Figura 63*Configuración de CORS de la dependencia fruitcake/laravel-cors.**Nota: Elaboración propia*

Laravel 8. Durante la migración de Laravel 7 a Laravel 8, la clase `DataBase\Seeds` debe ser reemplazada por `Database\Seeder`. Para actualizar las funcionalidades de los factories, se ha modificado el espacio de nombres a `Database\Factories` y `Database/Seeders`. Además, en el archivo `composer.json` se requiere eliminar el bloque `classmap` de la sección `autoload` y agregar las nuevas asignaciones de directorio con sus respectivos espacios de nombres.

Figura 64

Cambios en espacios de nombres.

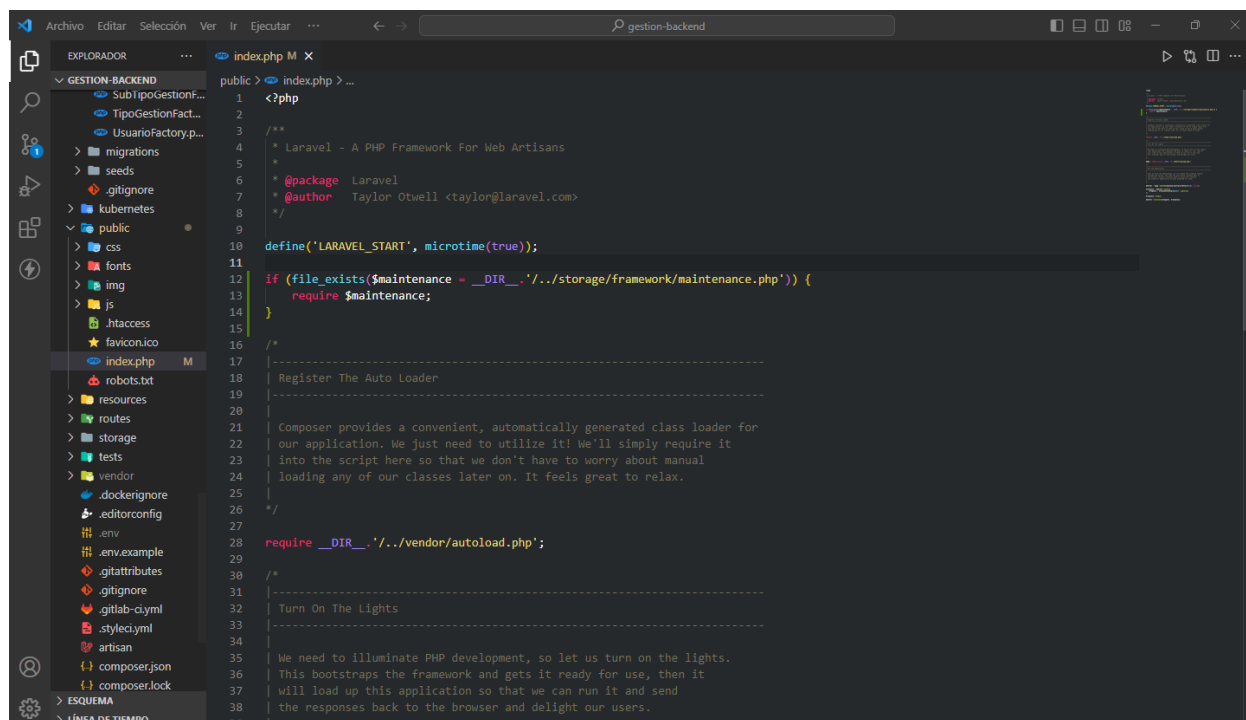


Nota: Elaboración propia

Con el propósito de mantener todos los componentes y modos del framework actualizados, se ha implementado una mejora en el modo de mantenimiento. Para lograrlo, se requiere agregar las siguientes líneas de código en el archivo `public/index.php`.

Figura 65

Configuración para el modo mantenimiento.

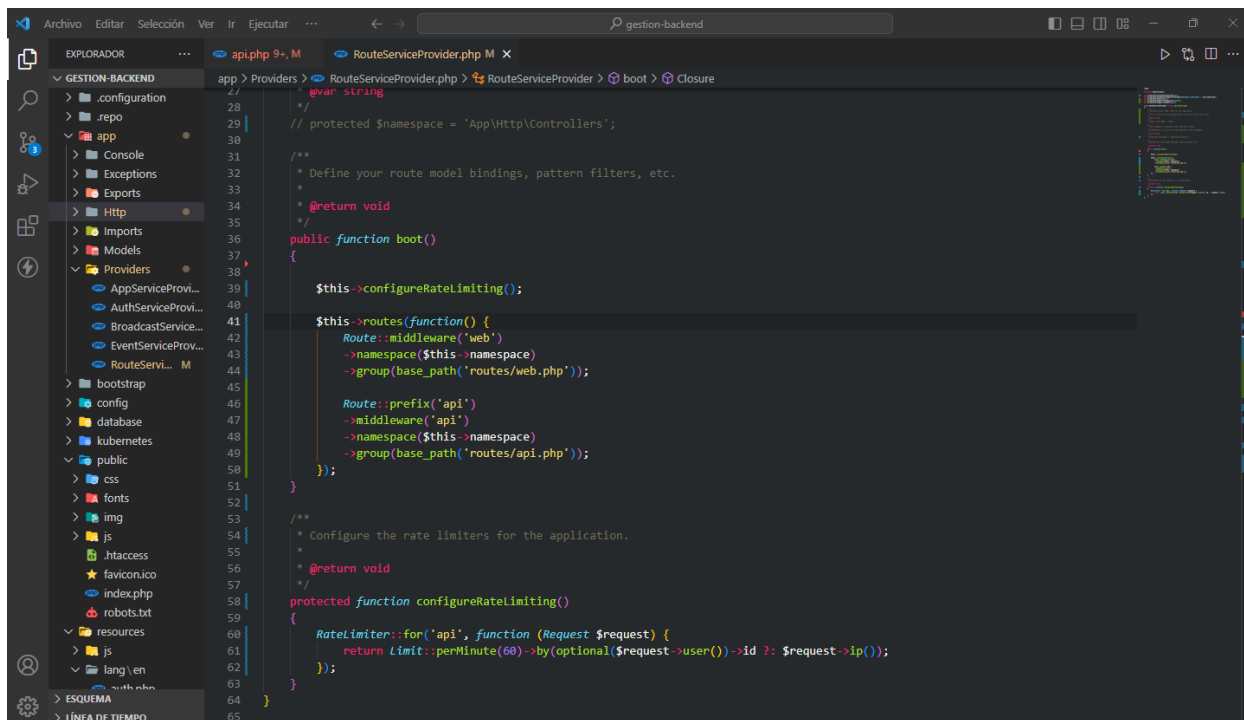


Nota: Elaboración propia

Para adoptar la sintaxis estándar de PHP para las declaraciones de ruta del controlador, que proporciona un mejor soporte para la navegación al controlador correspondiente, se procede a un cambio en la definición del controlador que usará una ruta específica. Este cambio requiere una configuración previa para deshabilitar el uso del \$namespace, evitando así la prefijación automática de las declaraciones de ruta del controlador.

Figura 66

Configuración para declaraciones de ruta del controlado.



Nota: Elaboración propia

Además de esto, se realiza la configuración de la definición de las rutas que posee la aplicación.

Figura 67

Cambio en la definición de rutas.

```

48 Route::group(['prefix' => 'api'], function () {
49     Route::name('login')->post('login', [AuthController::class, 'login']);
50
51     Route::group(['prefix' => 'auth'], function () {
52         Route::prefix('refresh')->group(function () {
53             Route::name('refresh')->post('refresh', 'AuthApi\AuthController@refresh');
54             Route::name('logout')->post('logout', 'AuthApi\AuthController@logout');
55             Route::name('me')->get('me', 'AuthApi\AuthController@me');
56         });
57     });
58
59     Route::group(['prefix' => 'municipios'], function () {
60         Route::prefix('municipios')->group(function () {
61             Route::name('obtener_municipio')->get('/', 'MunicipioController@index');
62             Route::name('almacenar_municipio')->post('/', 'MunicipioController@store');
63             Route::name('actualizar_municipio')->put('/', 'MunicipioController@update');
64             Route::name('eliminar_municipio')->delete('/', 'MunicipioController@destroy');
65         });
66     });
67
68     Route::prefix('clientes')->group(function () {
69         Route::name('obtener_clientes')->get('/', [ClienteController::class, 'index']);
70         Route::name('almacenar_clientes')->post('/', [ClienteController::class, 'store']);
71         Route::name('actualizar_clientes')->put('/', [ClienteController::class, 'update']);
72         Route::name('eliminar_clientes')->delete('/', [ClienteController::class, 'destroy']);
73     });
74
75     Route::prefix('deudores')->group(function () {
76         Route::name('obtener_deudor')->get('/', [DeudorController::class, 'index']);
77         Route::name('almacenar_deudor')->post('/', [DeudorController::class, 'store']);
78         Route::name('actualizar_deudor')->put('/', [DeudorController::class, 'update']);
79         Route::name('eliminar_deudor')->delete('/', [DeudorController::class, 'destroy']);
80     });
81
82     Route::prefix('referencias')->group(function () {
83         Route::name('obtener_referencia')->get('/', [ReferenciaController::class, 'index']);
84         Route::name('almacenar_referencia')->post('/', [ReferenciaController::class, 'store']);
85         Route::name('actualizar_referencia')->put('/', [ReferenciaController::class, 'update']);
86         Route::name('eliminar_referencia')->delete('/', [ReferenciaController::class, 'destroy']);
87     });
88 });

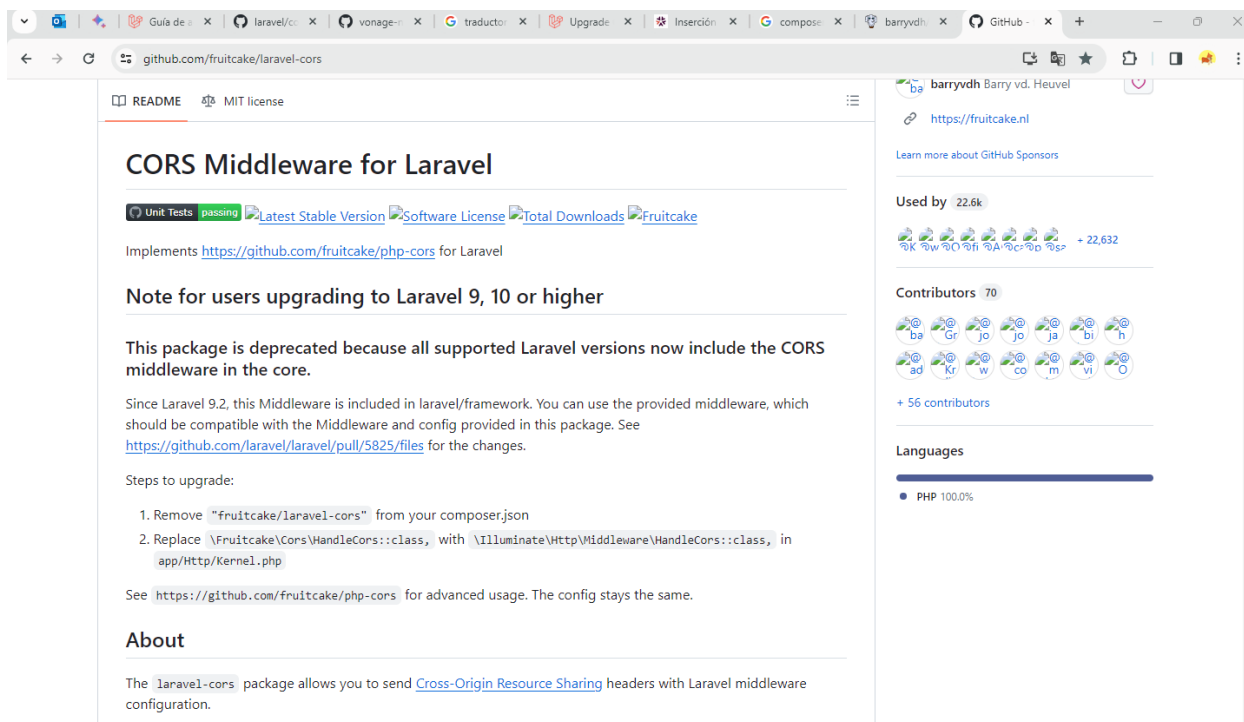
```

Nota: Elaboración propia

Laravel 9. Durante la migración de Laravel 8 a Laravel 9, la dependencia fruitcake/laravel-cors fue abandonada debido a la integración de CORS en el núcleo del framework. Para realizar la integración del nuevo middleware de CORS, es proporcionado otro apartado con documentación oficial explicando los pasos a seguir, anexando como dejar de usar la dependencia fruitcake/laravel-cors.

Figura 68

Documentación oficial sobre CORS para laravel 9.



Nota: Elaboración propia

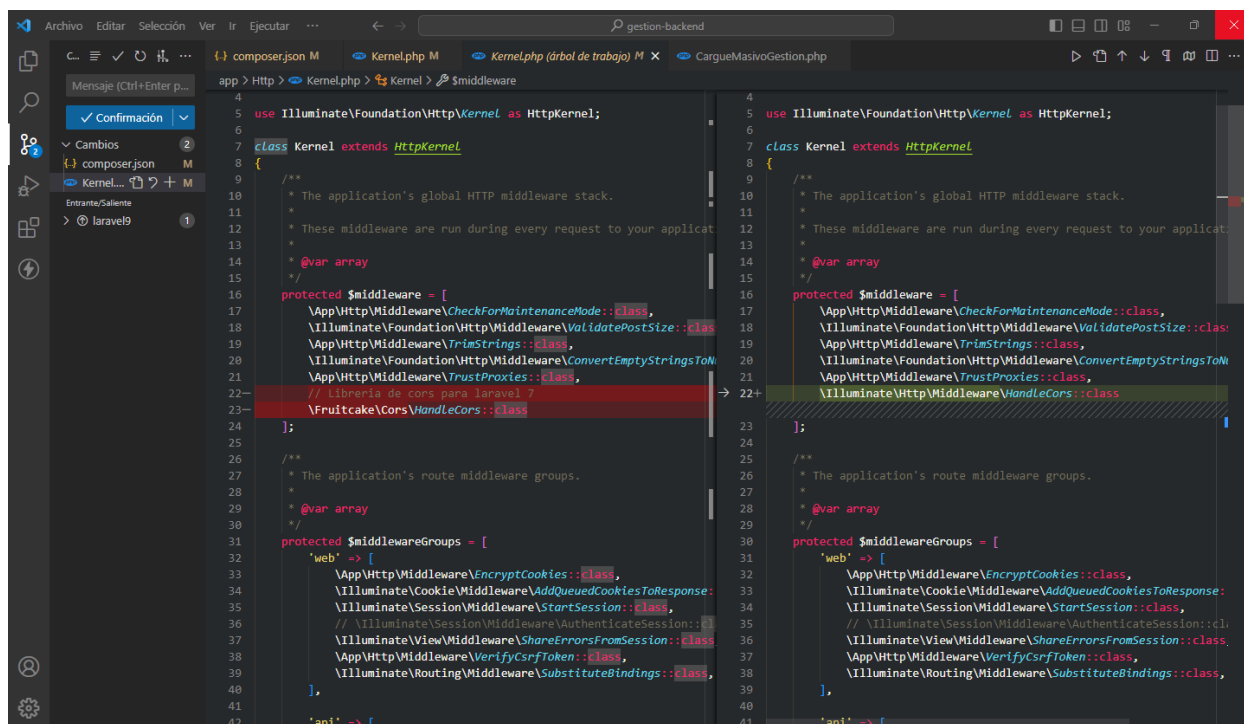
Inicialmente, se eliminó la dependencia `fruitcake/laravel-cors` del archivo `composer.json`.

Para luego, se sustituyó la clase `\Fruitcake\Cors\HandleCors::class` por

`\Illuminate\Http\Middleware\HandleCors::class` en el archivo `app/Http/Kernel.php`.

Figura 69

Configuración del middleware CORS del núcleo del framework.



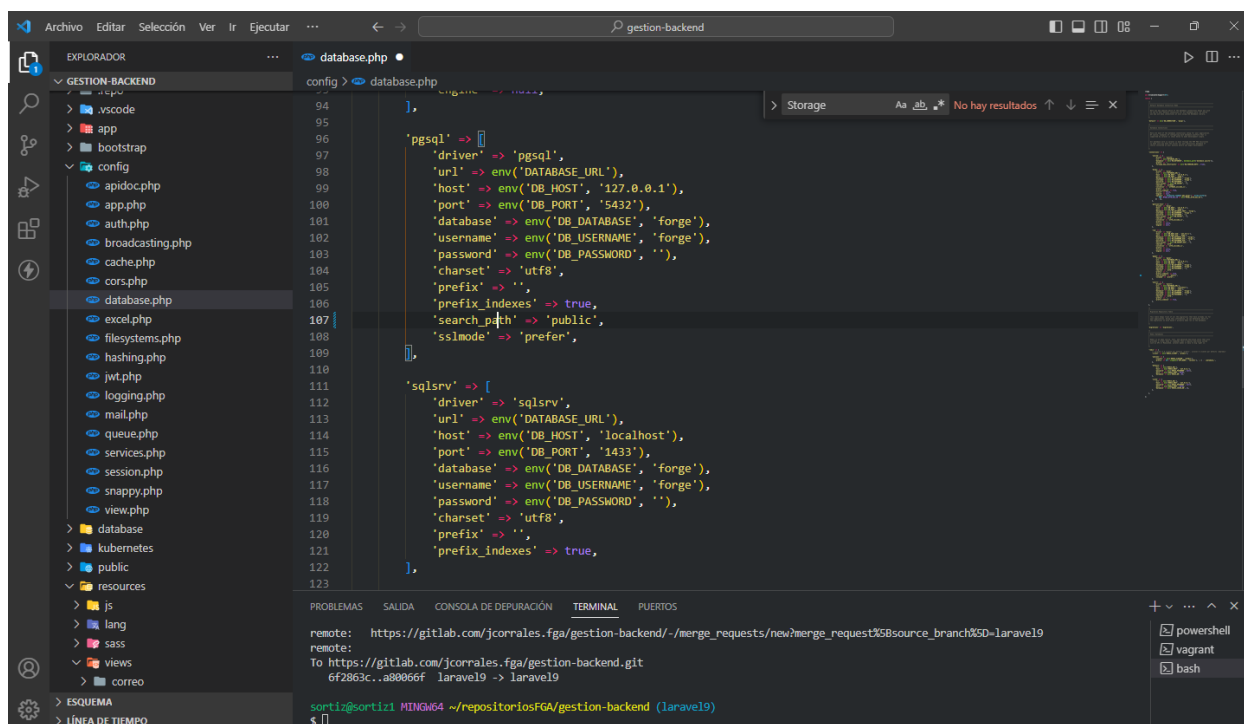
Nota: Elaboración propia

Además, se integró el sistema de mensajería de Symfony mediante la inclusión de las dependencias symfony/postmark-mailer, symfony/http-client y symfony/postmark-mailer.

También se realizó un cambio en la configuración para los esquemas de Postgres, donde se configuró la opción “schema” para establecer las rutas de búsqueda de conexiones Postgres en el archivo config/database.php de la aplicación, sustituyendo dicha opción por “search_path”.

Figura 70

Configuración para esquemas de Postgres.

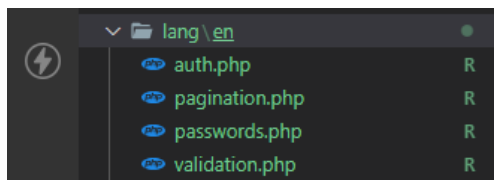


Nota: Elaboración propia

También se trasladó el directorio de traducciones del directorio "resources/lang" al directorio raíz de la aplicación.

Figura 71

Directorio de traducciones.

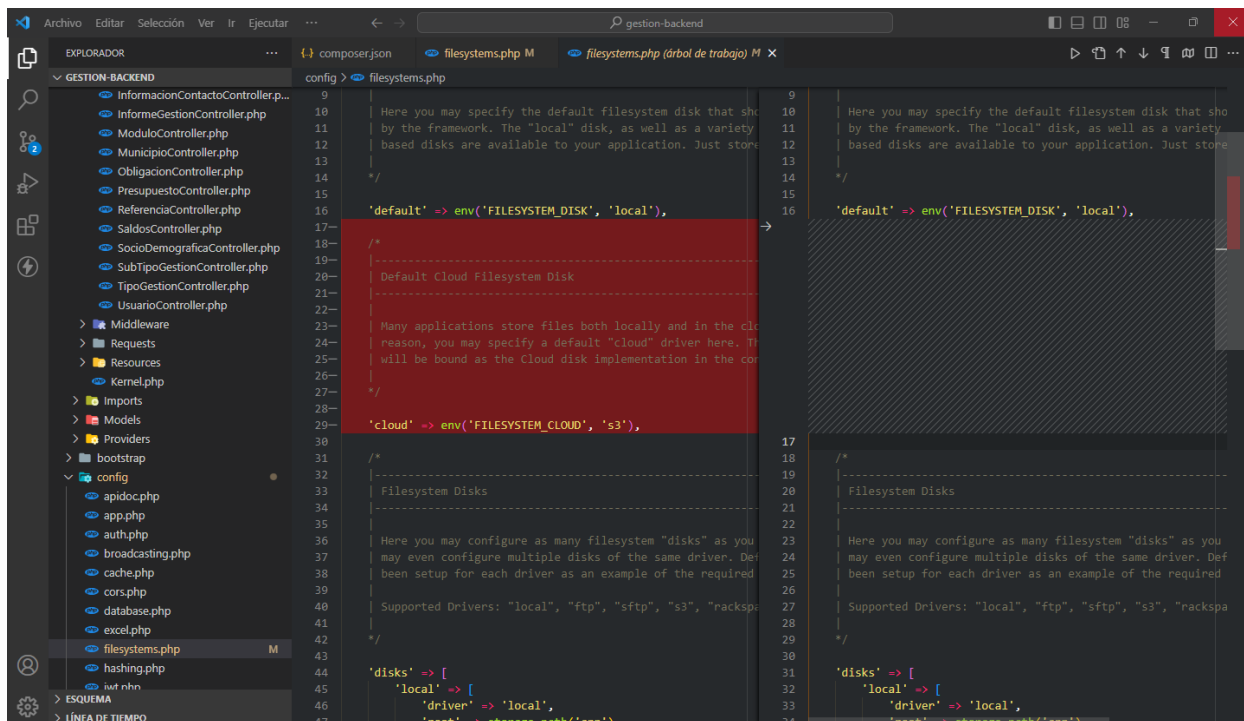


Nota: Elaboración propia

La opción de configuración del disco en la nube se eliminó del esqueleto de la aplicación, para ello se modificó el archivo “config/filesystems.php”.

Figura 72

Configuración del disco en la nube.

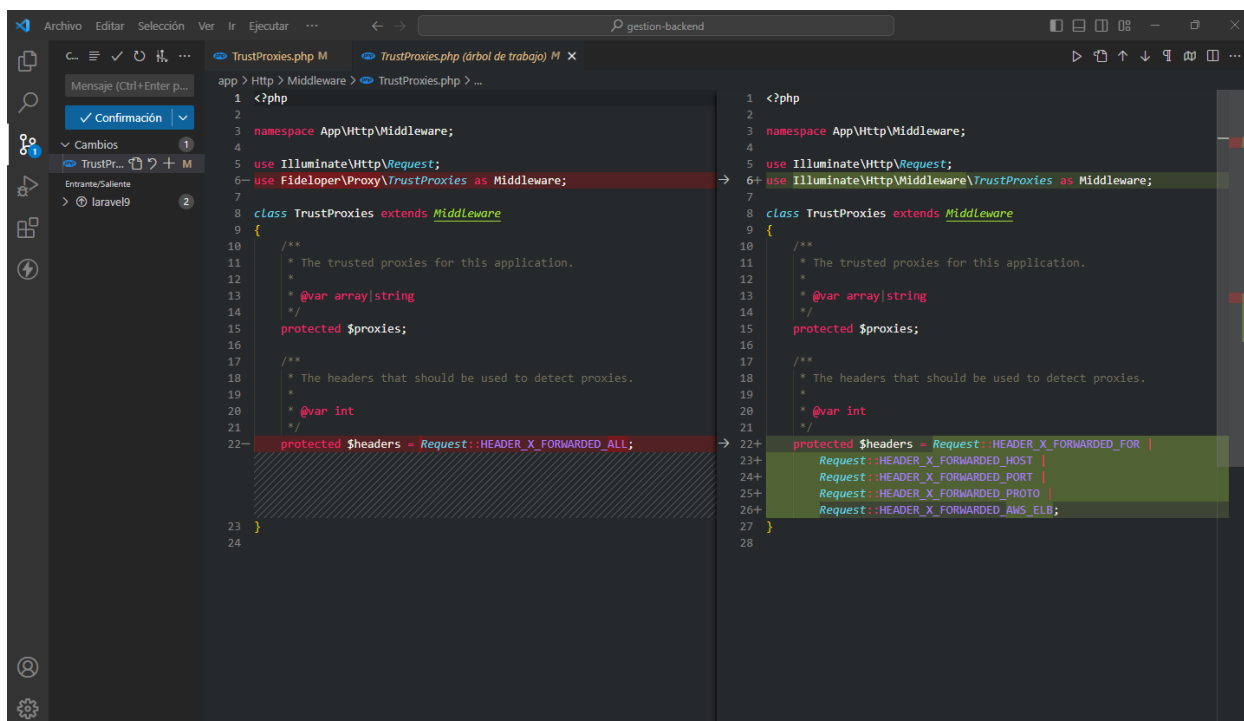


Nota: Elaboración propia

Es necesario configurar el middleware trusted proxy de la aplicación. Dentro del archivo “app/Http/middleware/TrustedProxies.php” donde se sustituyó “use Fideloper\Proxy\TrustProxies.php” por “use Illuminate\Http\Middleware\TrustProxies as Middleware”, adicionalmente cambia la propiedad “\$headers” y se elimina la dependencia “fideloper/proxy” de la aplicación.

Figura 73

Configuración middleware trusted proxy.

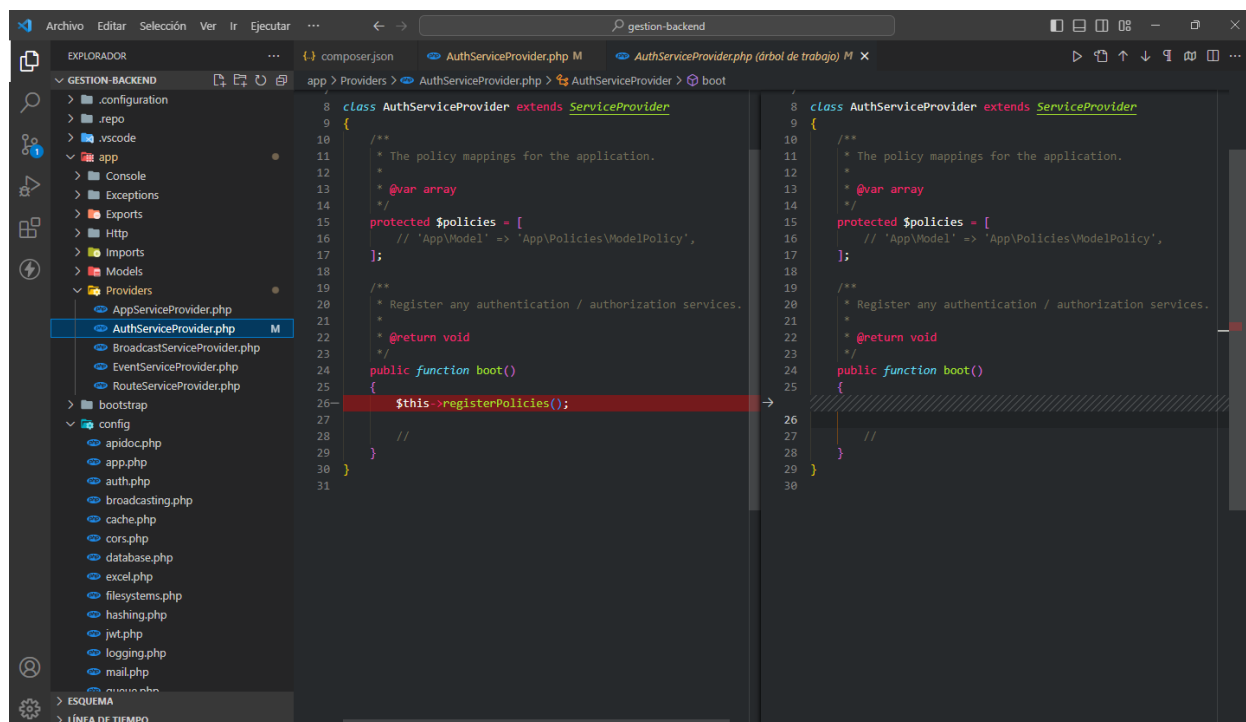


Nota: Elaboración propia

Laravel 10. Durante la migración de Laravel 9 a Laravel 10, el marco invoca automáticamente el método registerPolicies de AuthServiceProvider. Por lo tanto, se eliminó el llamado a este método de inicio de AuthServiceProvider.

Figura 74

Método de registro de políticas.

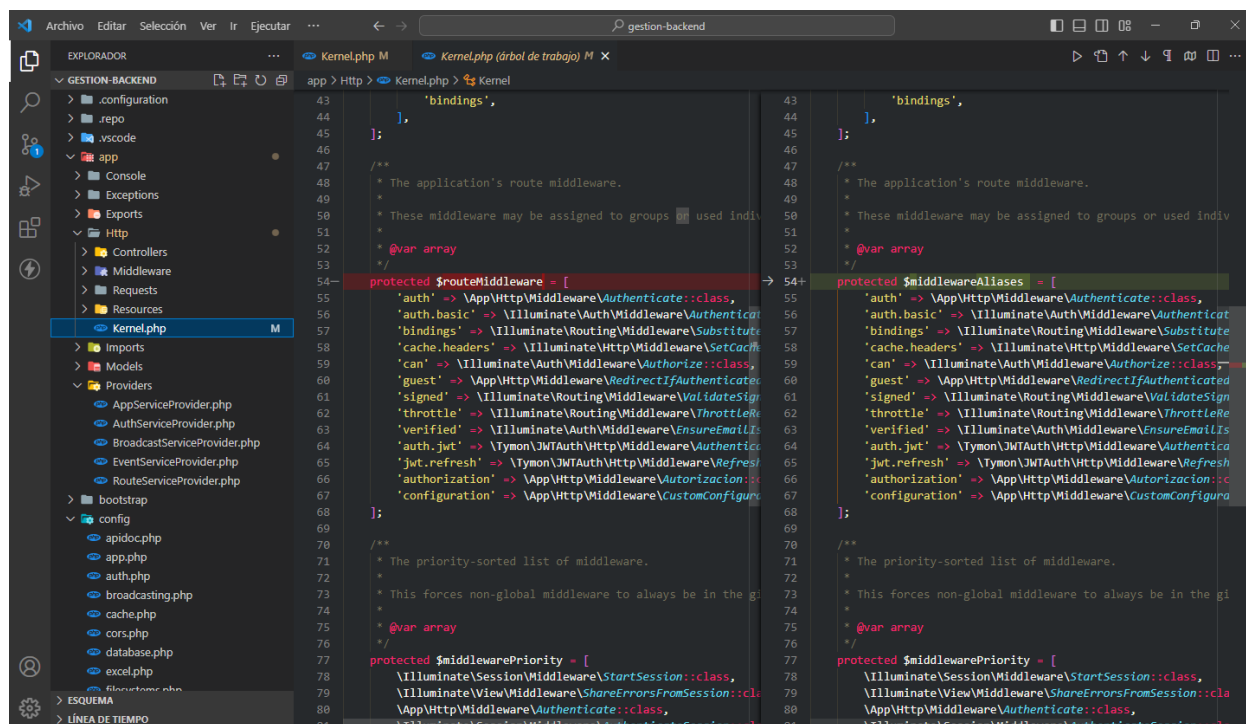


Nota: Elaboración propia

También se realizó un cambio en la propiedad “\$routeMiddleware” de la clase “App\Http\Kernel”. Esta propiedad ha sido renombrada a “\$middlewareAliases”.

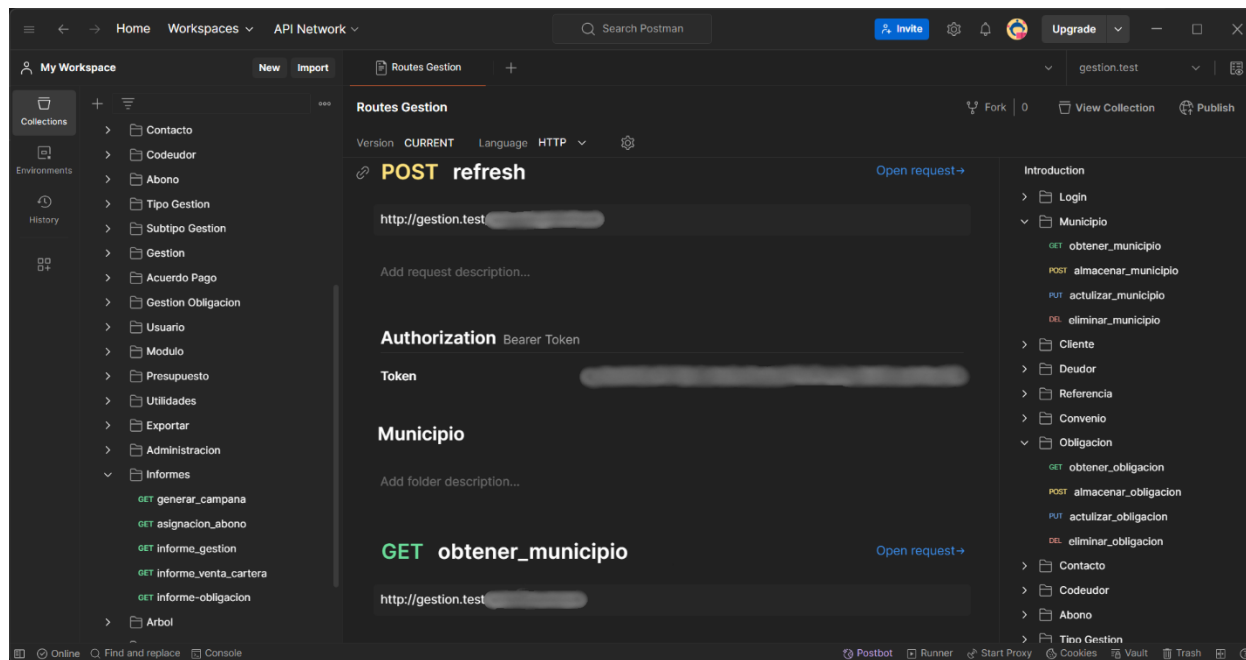
Figura 75

Alias de middleware.



Nota: Elaboración propia

Para finalizar esta etapa y garantizar que todas las funcionalidades hayan quedado como inicialmente estaban planteadas se realiza un mapeo de todas las rutas de la aplicación, en búsqueda de posibles errores

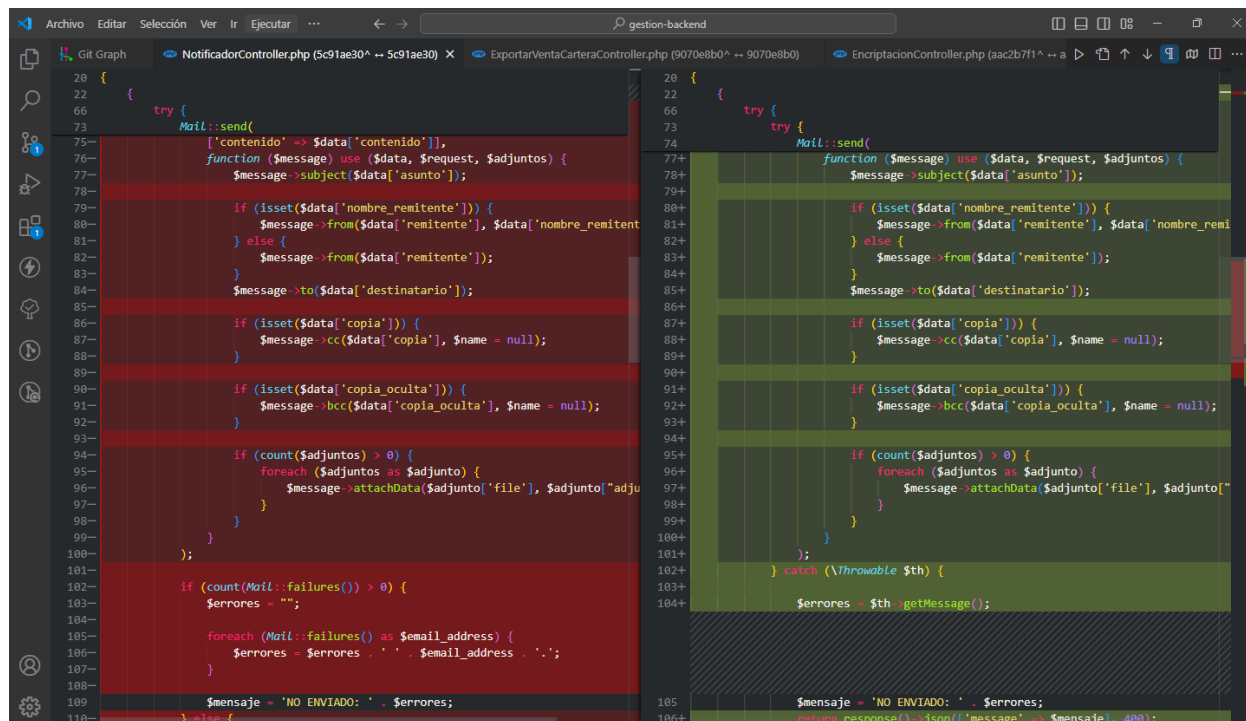
Figura 76*Mapeo de rutas.**Nota: Elaboración propia*

Durante este mapeo se identifican las siguientes situaciones:

La funcionalidad de notificación por correo electrónico se vio afectada por un conflicto, al no encontrar ya que la función “failures” no estaba presente en el nuevo sistema de mensajería incorporado en Symfony. Para resolver este problema, se modificó el controlador al que apuntaba la ruta correspondiente. Se eliminó y se reorganizó un nuevo método para manejar los correos electrónicos que generaban fallos.

Figura 77

Reorganización de método de mensajería vía email.

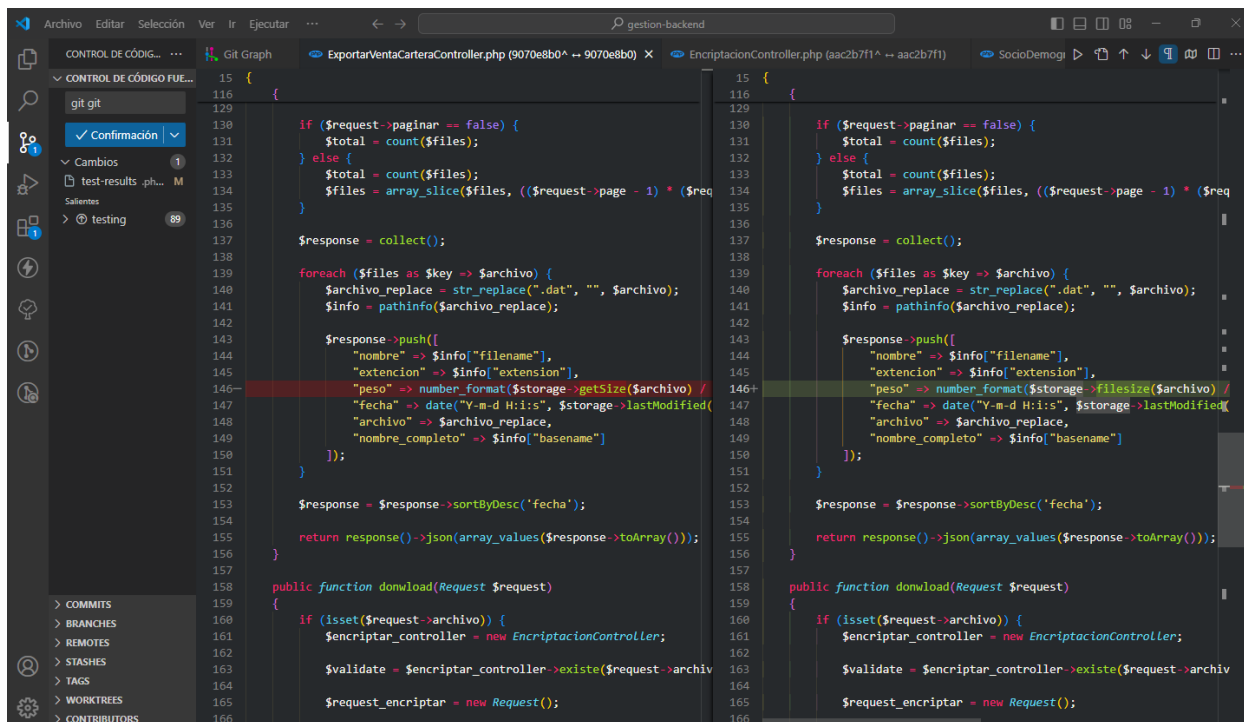


Nota: Elaboración propia

La función “getSize” de la clase Storage quedó obsoleta. Esta función solía devolver el tamaño del archivo, pero fue reemplazada por otra función llamada “filesize”, que proporciona la misma información solicitada.

Figura 78

Función para obtener tamaño del archivo.

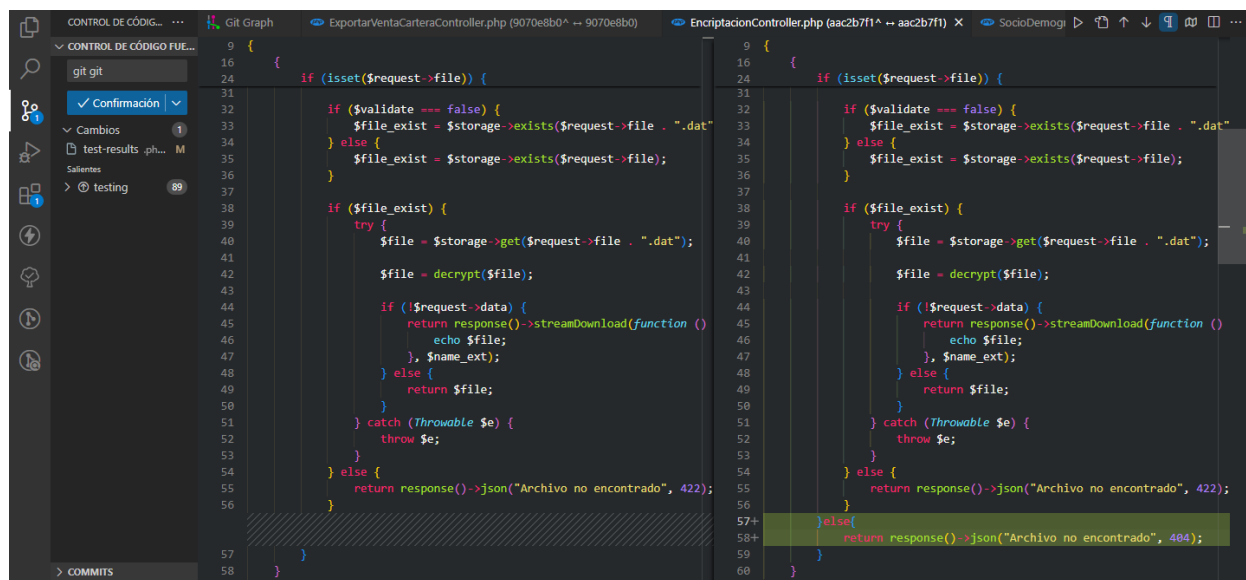


Nota: Elaboración propia

Durante la validación de los parámetros recibidos para la ruta de desencriptar archivo, se detectaron fallos en la respuesta devuelta por el controlador.

Figura 79

Manejo de errores en archivo Encriptacion.

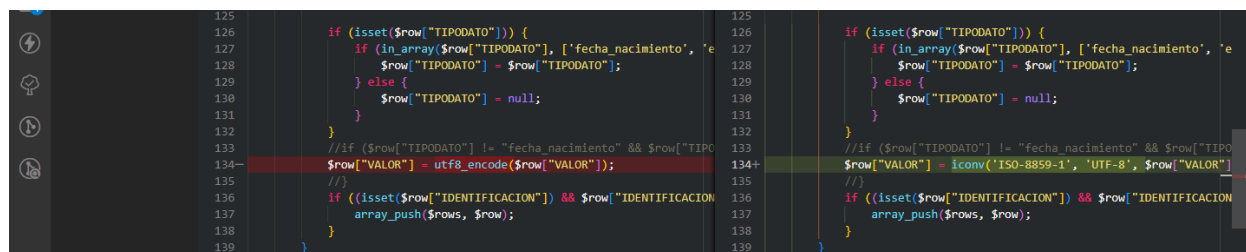


Nota: Elaboración propia

Una de las funciones nativas del lenguaje php, “utf8_encode”, que se utilizaba para especificar la codificación de los datos de un archivo, ha sido marcada como obsoleta. En su lugar, se sustituye por “iconv”, que proporciona la misma funcionalidad.

Figura 80

Cambio de funcionalidad depreciada por php.

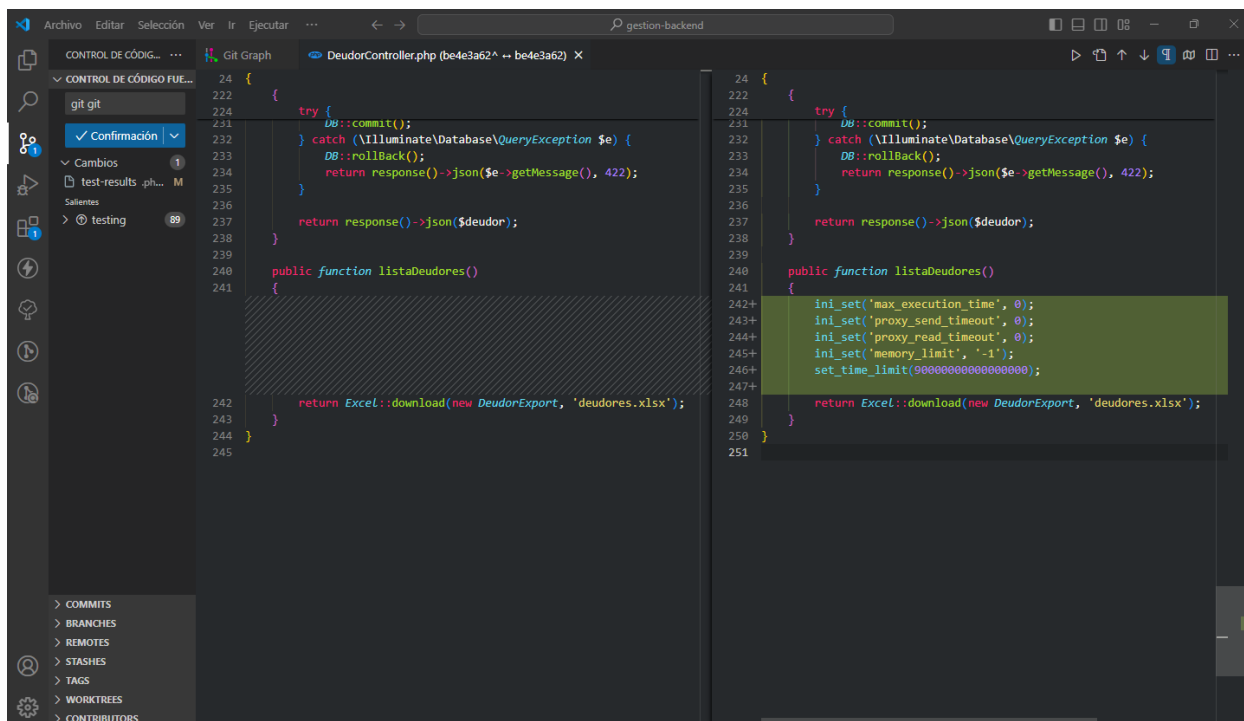


Nota: Elaboración propia

La ruta encargada de devolver un archivo que enumera a los deudores que superan el tiempo límite de respuesta por defecto necesitaba ajustes. Por lo tanto, se configuró el controlador de la ruta para permitir que se excediera dicho límite de tiempo.

Figura 81

Exportar deudores.



Nota: Elaboración propia

3.3.4 Cierre de vulnerabilidades.

En esta actividad, se tiene en consideración la migración entre versiones de Laravel, ya que muchos de los cambios realizados en el framework están dirigidos a cerrar brechas de seguridad o corregir errores que podrían ser explotados por actores externos.

Además de eso, se identificaron rutas que podrían ser explotadas debido a su implementación deficiente o a fallos en su funcionamiento. Algunas de estas incluyen las rutas

“actualizar presupuesto” y “eliminar presupuestos”, así como la ruta “obtener datos sig” que perdió su funcionalidad al realizar la transición de versión a otra aplicación con la que se comunicaba.

Figura 82

Depuración de rutas.

```

50 Route::group(['prefix' => ''], function () {
51     Route::group(['prefix' => ''], function () {
52         Route::group(['prefix' => ''], function () {
165 Route::prefix('')->group(function () {
172     Route::name('modificar_password')->post('');
173
174 });
175
176 Route::prefix('')->group(function () {
177     Route::name('obtener_modulo')->get('/', [ModuloControl
178     Route::name('almacenar_modulo')->post('/', [ModuloCont
179     Route::name('actualizar_modulo')->put('/', [ModuloC
180     Route::name('eliminar_modulo')->delete('/', [Modul
181 });
182
183 Route::prefix('')->group(function () {
184     Route::name('obtener_presupuesto')->get('/', [Presupue
185     Route::name('importar_presupuesto')->post('/', [Presupue
186
187 // // Funcion inexistente en el controlador Eliminar
188 // Route::name('actualizar_presupuesto')->put('/{id}',
189 // Route::name('eliminar_presupuesto')->delete('/{id}',
190 // // -----
191 });
192
193 Route::prefix('')->group(function () {
194     // Desuso
195     // // TODO: Ruta con error 401 -> No se realiza autent
196     // Route::name('obtener_datos_sig')->post('/informacio
197
198 Route::name('generar_documento')->post('');
199 Route::name('datos_presupuesto')->post('');
200 Route::name('enviar_sms')->post('');
201 Route::name('enviar_email')->post('');
202 Route::name('interes_fga')->post('');
203 Route::name('obtener_acuerdo_gestion')->get('');
204 Route::name('obtener_grafica_presupuesto')->post('');
205 Route::name('cargue_masivo_gestion')->post('');
206
207 });
208
209 Route::prefix('')->group(function () {
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

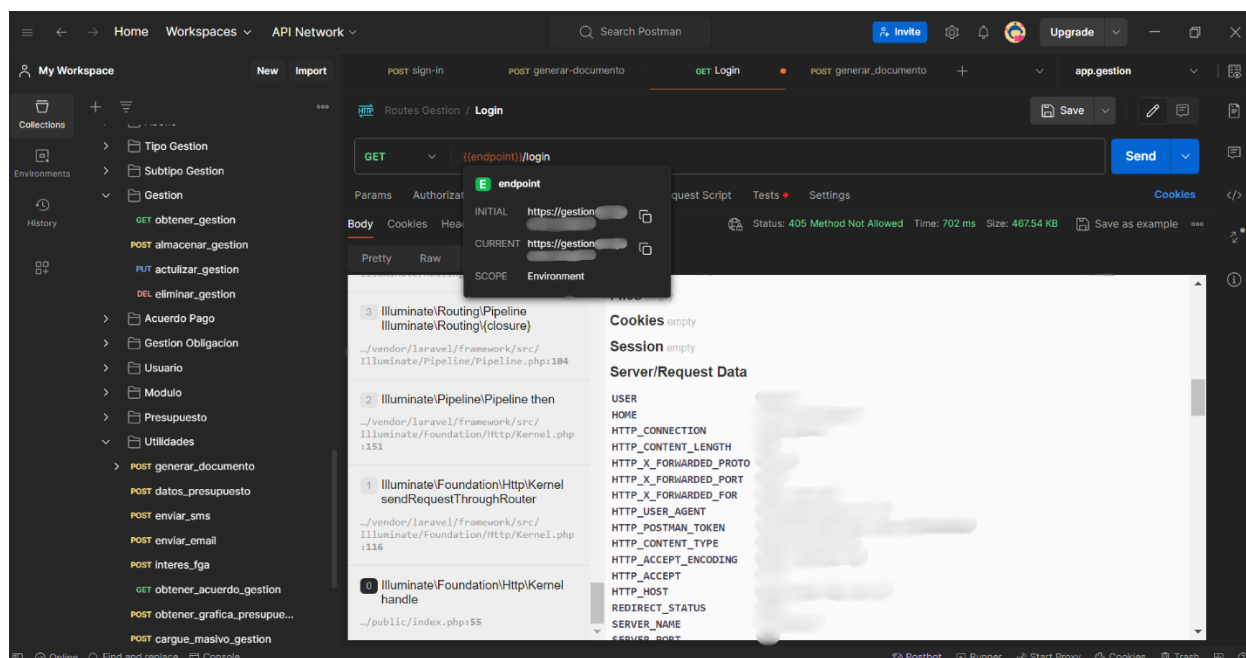
Nota: Elaboración propia

Después de completar las tareas anteriores, se llevó a cabo la fusión con la rama principal del repositorio para su despliegue. Después de realizado dicho proceso, se detectó una vulnerabilidad de seguridad en el sitio de producción, que dejaba expuestos fragmentos del código donde se generó la falla y el archivo de variables de entorno (.env) para cualquier persona, revelando las credenciales de conexión de todos los servicios de los que hace uso el aplicativo. Esta vulnerabilidad se originó al dejar activado el modo de depuración, el cual está

controlado la variable de entorno “APP_DEBUG” con valor booleano. Esta variable debe estar configurada como “true” en un ambiente de producción.

Figura 83

Modo depuración activo.



Nota: Elaboración propia

3.3.5 Desarrollo de pruebas unitarias e integración.

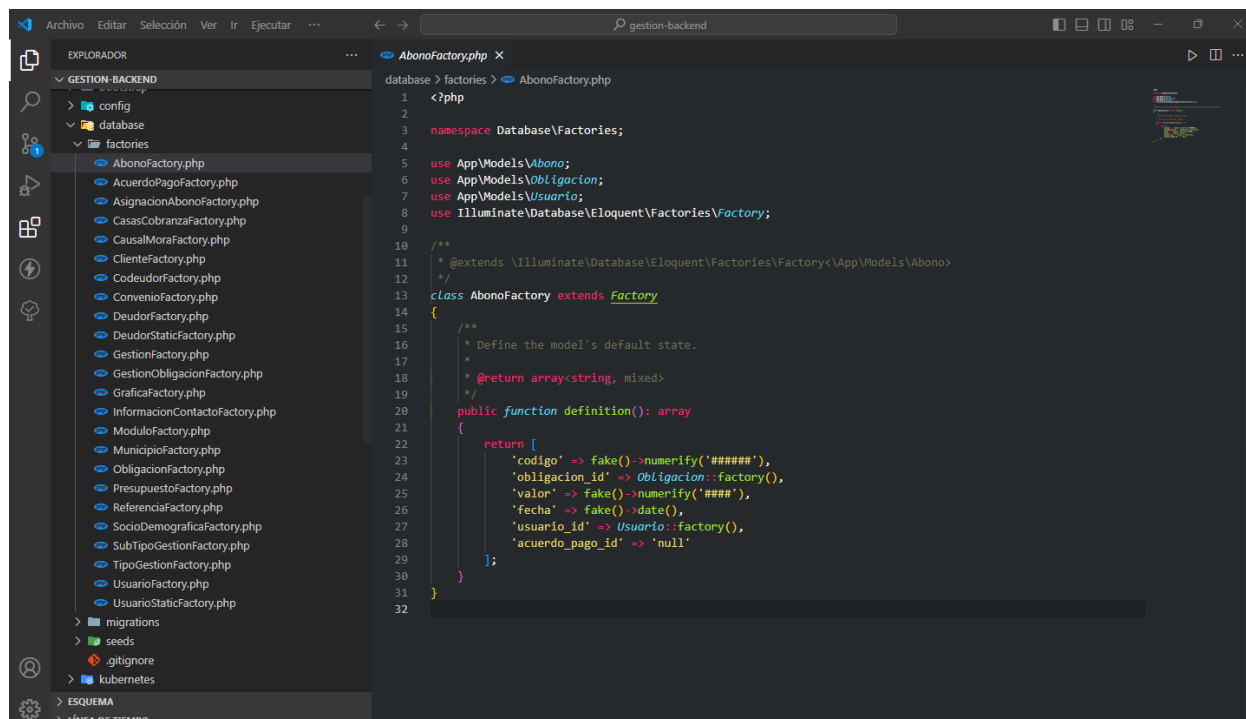
Para esta actividad, se emplearon en gran medida las herramientas proporcionadas por el framework (Factories, Models, Tests, Migrations, Asserts), con el propósito de automatizar y mejorar la eficacia de las pruebas.

Configuración. Con el propósito de automatizar la generación de datos falsos para enriquecer las pruebas, se establecen las fábricas (factory) correspondientes a cada modelo

(model) en la aplicación, empleando la dependencia "fakerphp/faker", lo que culmina en un total de 24 fábricas.

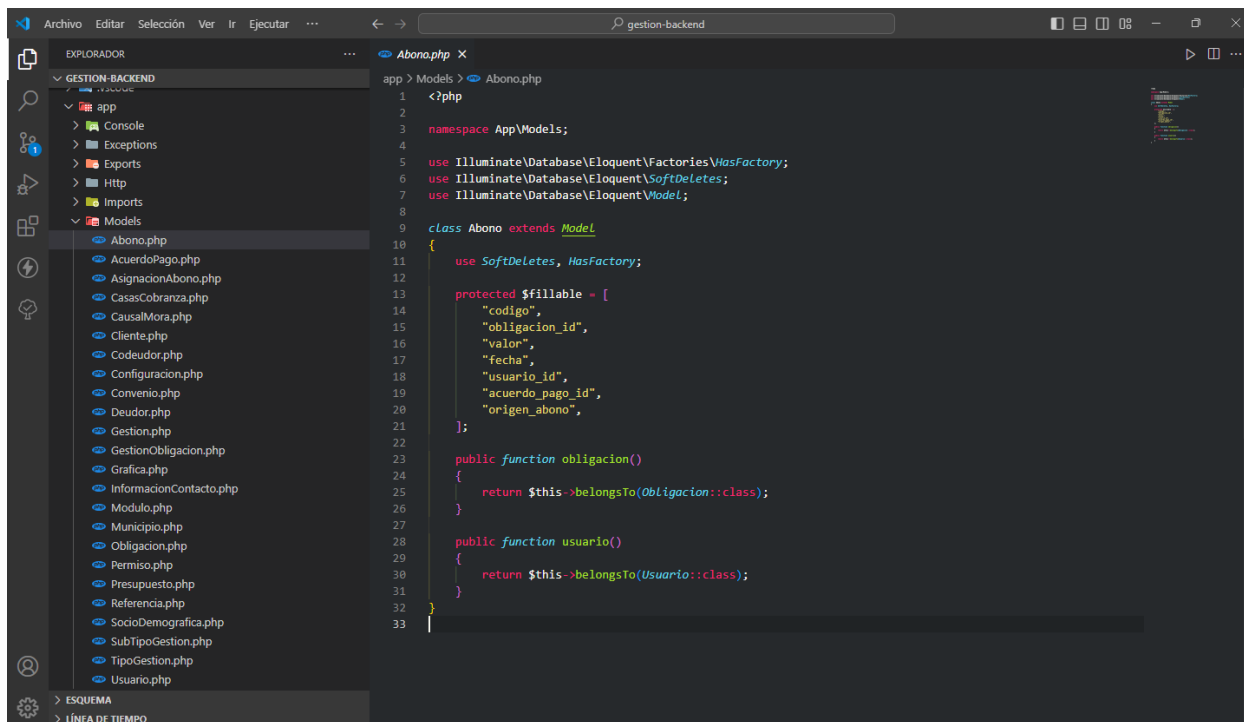
Figura 84

Fábricas de modelos.



Nota: Elaboración propia

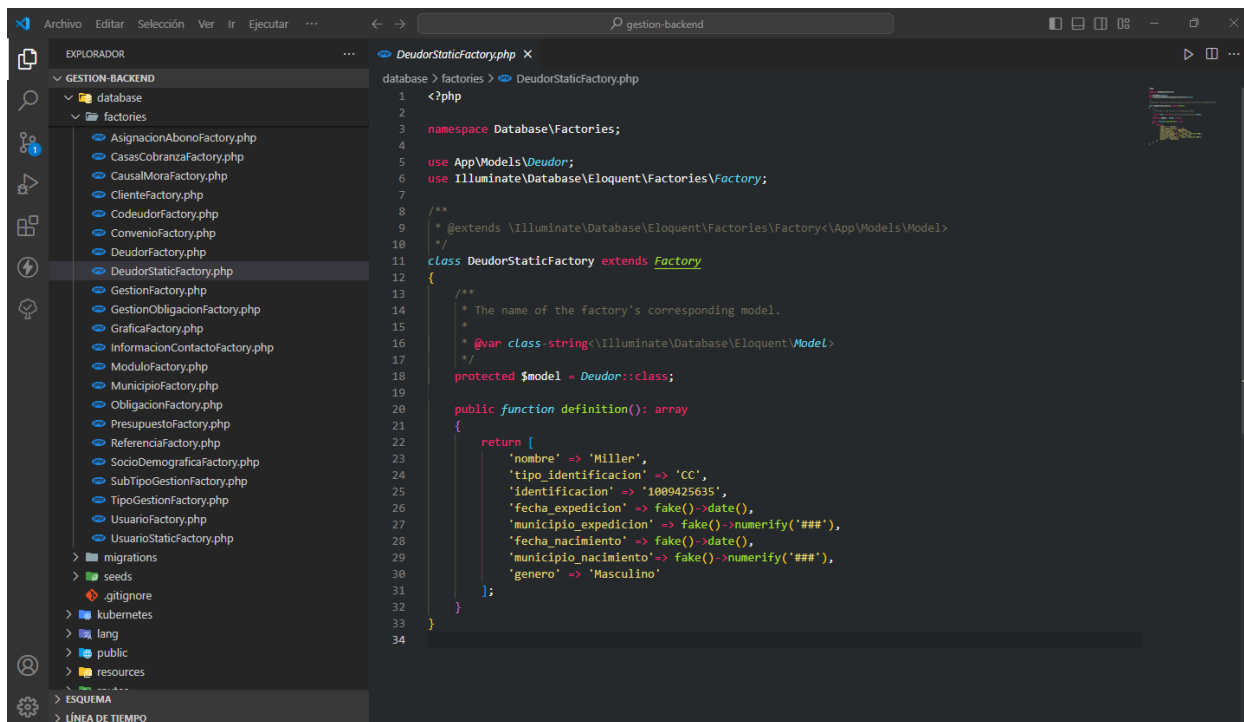
Estas fábricas para ser asociadas automáticamente por el framework deben llamarse del mismo modo que el modelo incluyendo al final la palabra “Factory” y adicionalmente para poder hacer uso del método “factory”, es necesario que el archivo del modelo utilice la clase “Illuminate\Database\Eloquent\Factories\HasFactory”.

Figura 85*Modelos.**Nota: Elaboración propia*

Adicionalmente se necesitó generar fabricas adicionales a un modelo, ya que estas fábricas no podían llamarse de la misma manera que el modelo, la asignación automática no se generaba, para solucionar este inconveniente se asociaba el archivo de la fábrica manualmente haciendo uso de la siguiente declaración “protected \$model = Model::class” donde el “Model” es el modelo al que se asocia dicha fabrica.

Figura 86

Fabrica adicional asociada al modelo Deudor.

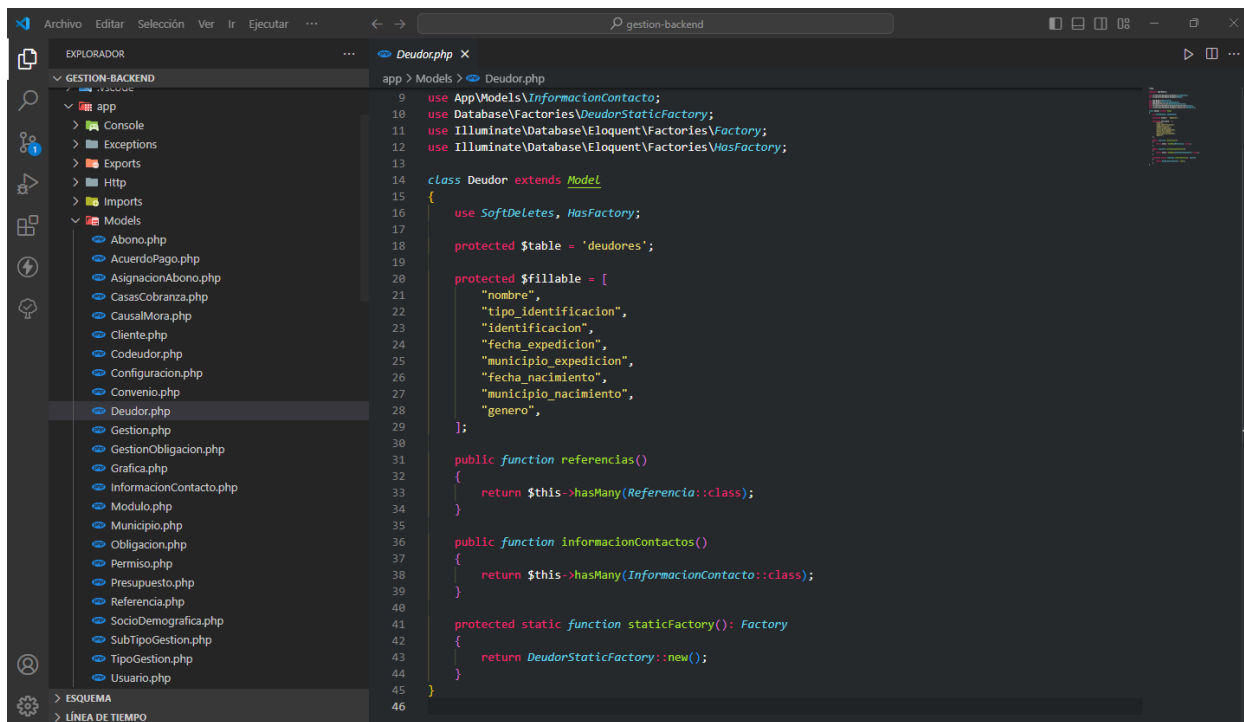


Nota: Elaboración propia

Para esta nueva fábrica, se requiere que modelo al que se asoció, utilice la clase “Illuminate\Database\Eloquent\Factories\Factory”, y se defina un método estático protegido que devuelva un objeto “Factory”, permitiendo así el uso de dicha fábrica.

Figura 87

Método para uso de fabrica adicional en el modelo Deudor.

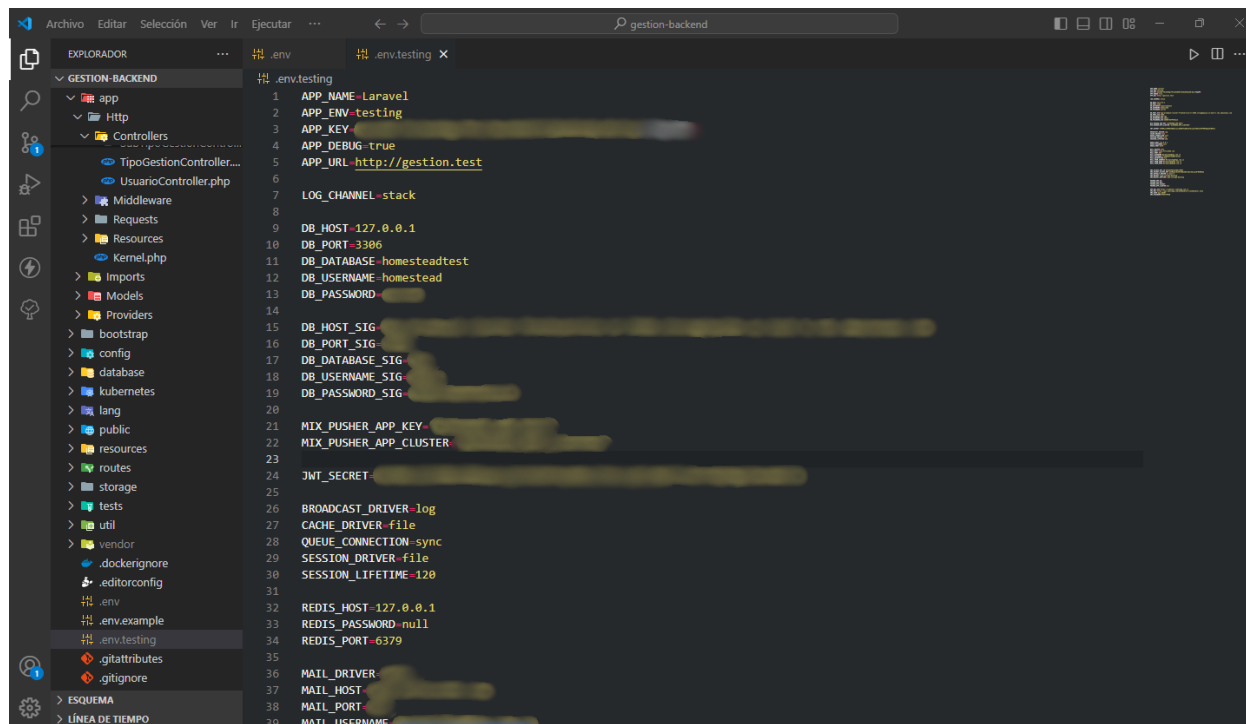


Nota: Elaboración propia

Adicionalmente se creó un nuevo archivo de variables de entorno, donde se configuró la conexión a la base local, creada en la configuración del entorno, esto debido a que, al ejecutar las pruebas, se eliminan todas las tablas de la base de datos y se generan nuevamente gracias a las migraciones, adicionalmente en el archivo se declara que es un ambiente de pruebas.

Figura 88

Variables de entorno para ejecución de pruebas.



Nota: Elaboración propia

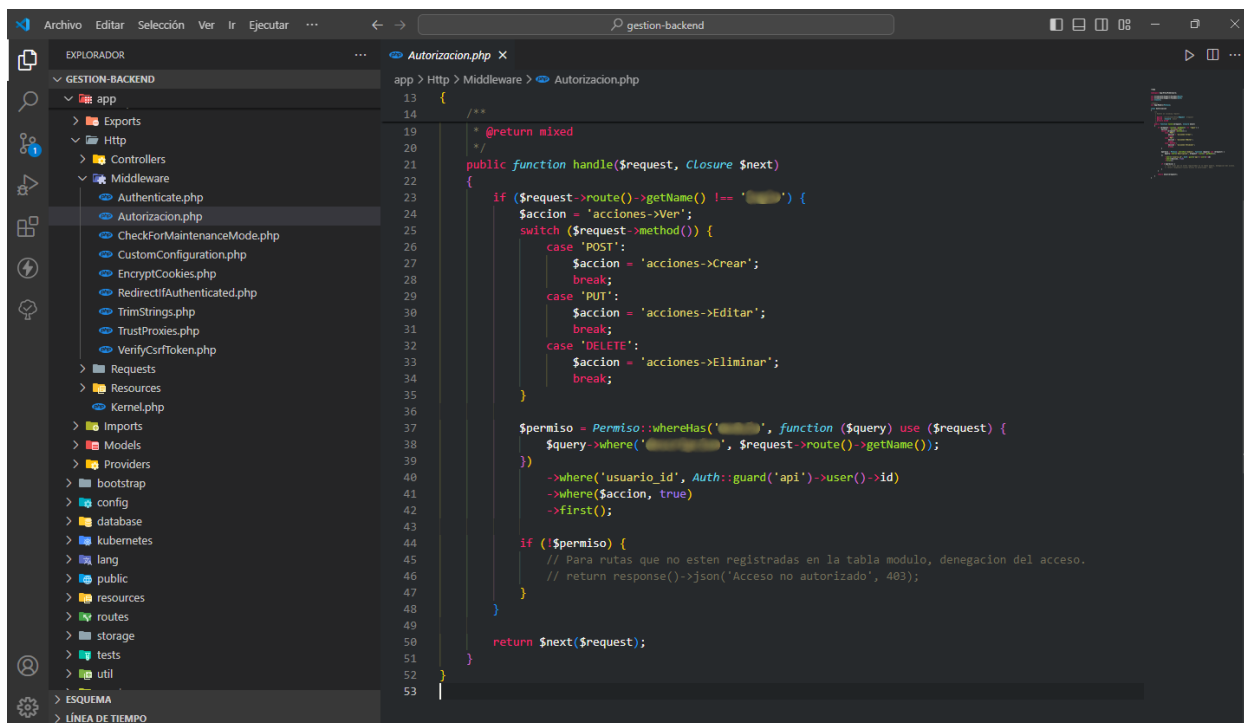
Configuración adicional para pruebas de integración. Dado que estas pruebas tienen como objetivo evaluar un conjunto de funcionalidades, al intentar llevar a cabo las pruebas utilizando los endpoints, se identificaron ciertas configuraciones en el sistema que afectan su funcionamiento.

Una de ellas es un middleware que controla el acceso a las rutas según ciertos permisos. Estas rutas son accesibles únicamente si están registradas en una tabla de la base de datos; de lo contrario, se denegará el acceso. Sin embargo, para llevar a cabo las pruebas de integración, es necesario desactivar este middleware. Esto se debe a que se realiza constantemente una

actualización de la base de datos, lo que puede provocar que los datos sobre los módulos no estén disponibles en la tabla y, por lo tanto, las rutas no sean accesibles.

Figura 89

Middleware de permisos por modulo.



Nota: Elaboración propia

También cuenta con un middleware diseñado para proteger la conexión al bucket S3 de AWS. Para las pruebas que requieran el uso del bucket, es necesario configurar las credenciales realizando la consulta a través de la clase “DB” al lugar donde se encuentran dichas credenciales.

Figura 90

Obtención de credencias para bucket S3 en AWS.

```
$configuraciones = DB::connection('')->table('')->whereIn('',['',''])->get();

$s3Credentials = $configuraciones->where('','')->first();
$s3Credentials = json_decode($s3Credentials->json);

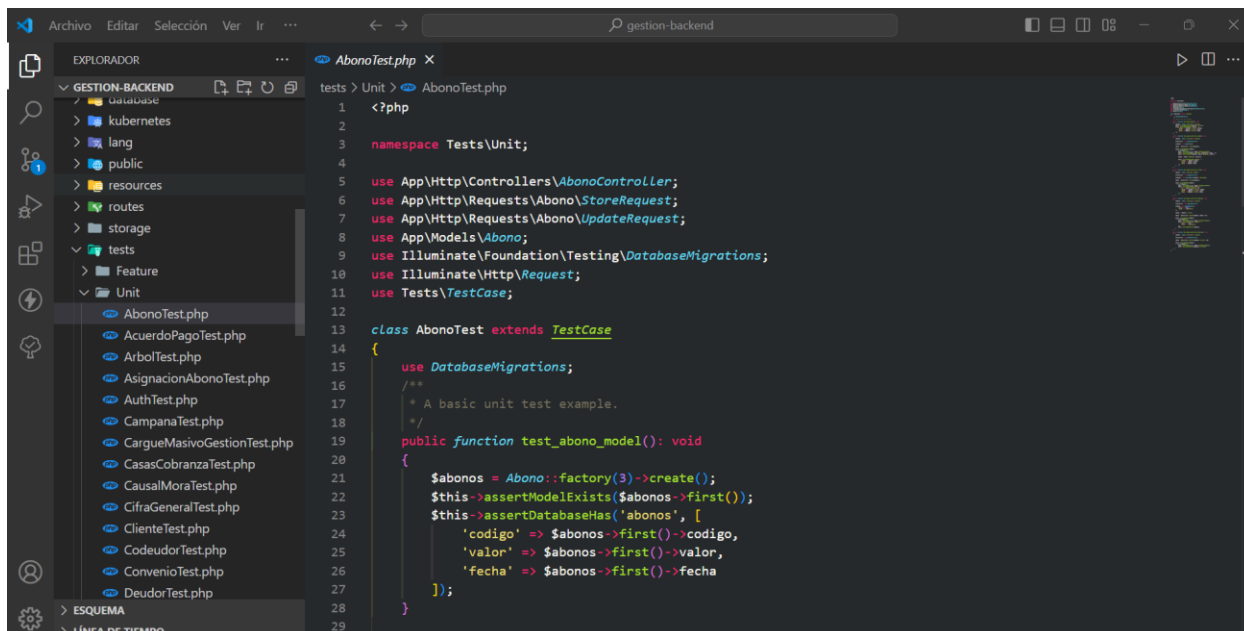
Config::set('filesystems.disks.s3.key', $s3Credentials->gestion->key);
Config::set('filesystems.disks.s3.secret', $s3Credentials->gestion->secret);
Config::set('filesystems.disks.s3.region', $s3Credentials->gestion->region);
Config::set('filesystems.disks.s3.bucket', $s3Credentials->gestion->bucket);
```

Nota: Elaboración propia

Pruebas unitarias. Estas pruebas son generadas a través de comandos en el directorio “test/Unit”, y es obligatorio el remplazo el uso de la clase “PHPUnit\Framework\TestCase” por “Test\TestCase” y hacer uso de la clase “DatabaseMigrations”.

Figura 91

Archivo de pruebas unitarias para Abono.

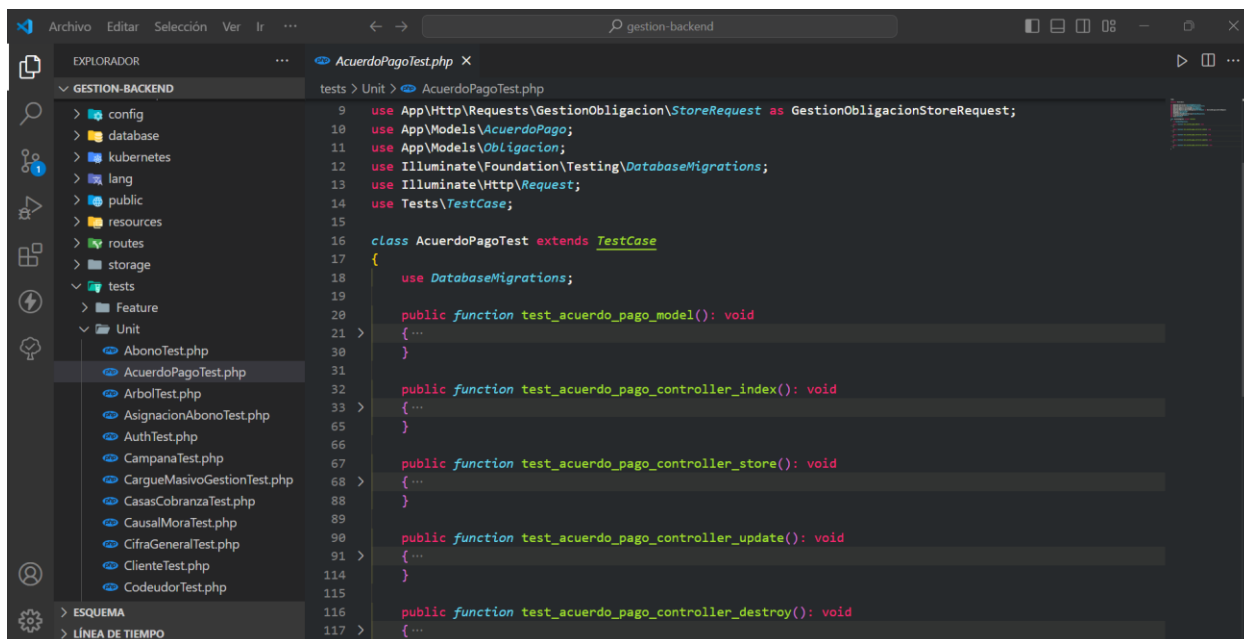


Nota: Elaboración propia

Cada uno de los archivos de pruebas unitarias contienen los métodos para realizar las pruebas para cada uno de los métodos de los controladores presentes en el aplicativo y en algunos casos al modelo.

Figura 92

Archivo de pruebas unitarias para acuerdo de pago.

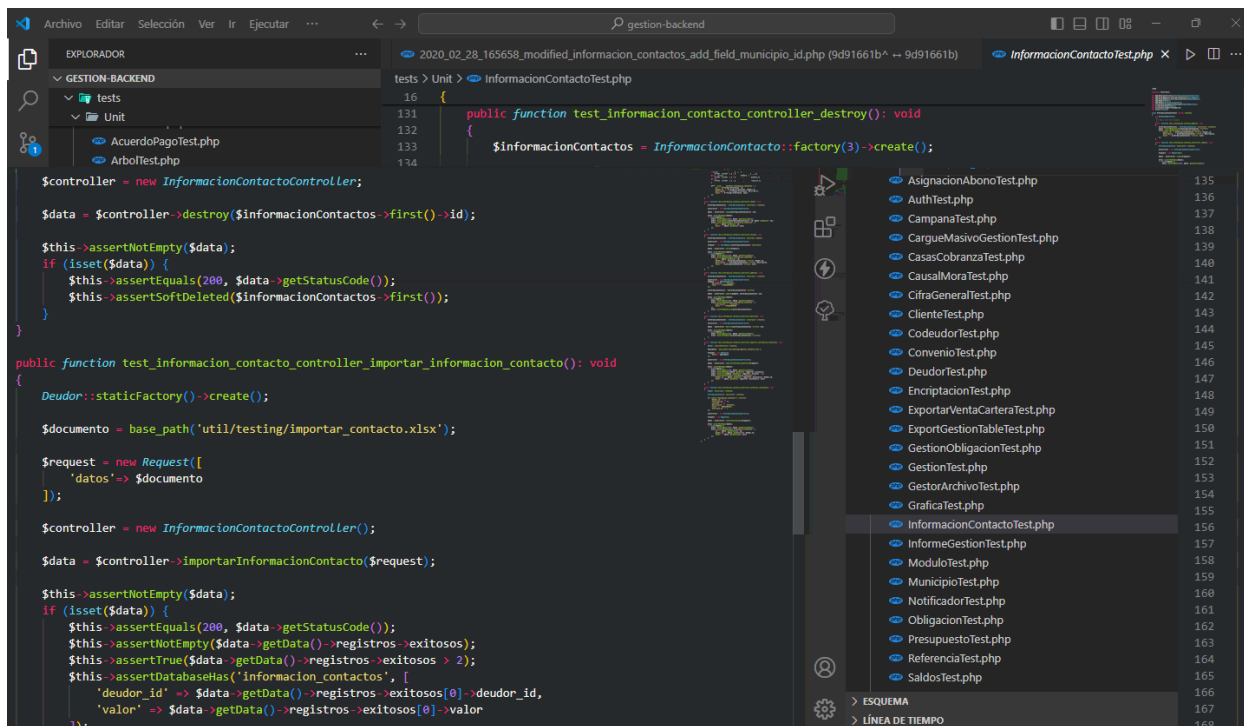


Nota: Elaboración propia

Dependiendo de lo que se desea testear, en algunos casos es invocado el modelo, el controlador u ambos, y se definen las afirmaciones con los resultados esperados comparándolos con la respuesta obtenida, con el fin de verificar el correcto funcionamiento o detectar posibles fallas.

Figura 93

Archivo de pruebas unitarias para cargue masivo.

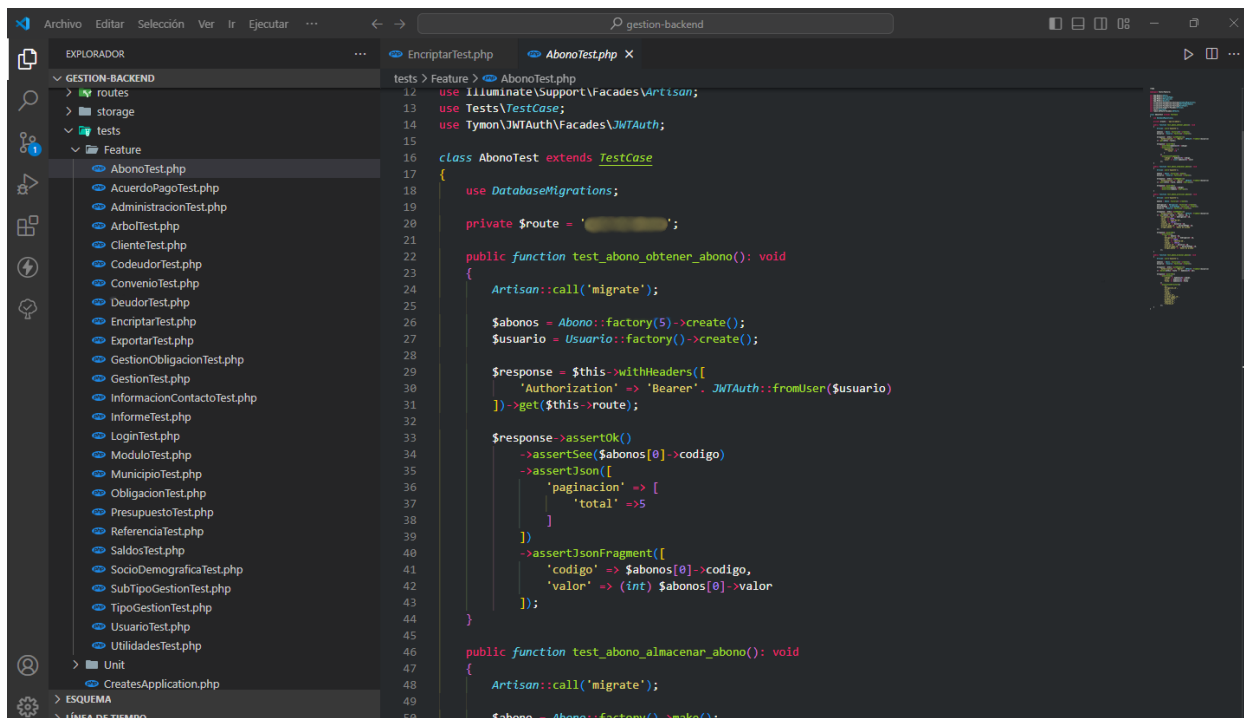


Nota: Elaboración propia

Pruebas de integración. Estas pruebas son generadas a través de comandos en el directorio “test/Feature”, y es obligatorio hacer uso de la clase “DatabaseMigrations” y en todos los métodos definidos usar “Artisan::call(‘migrate’)”.

Figura 94

Archivo de pruebas de integración para Abono.

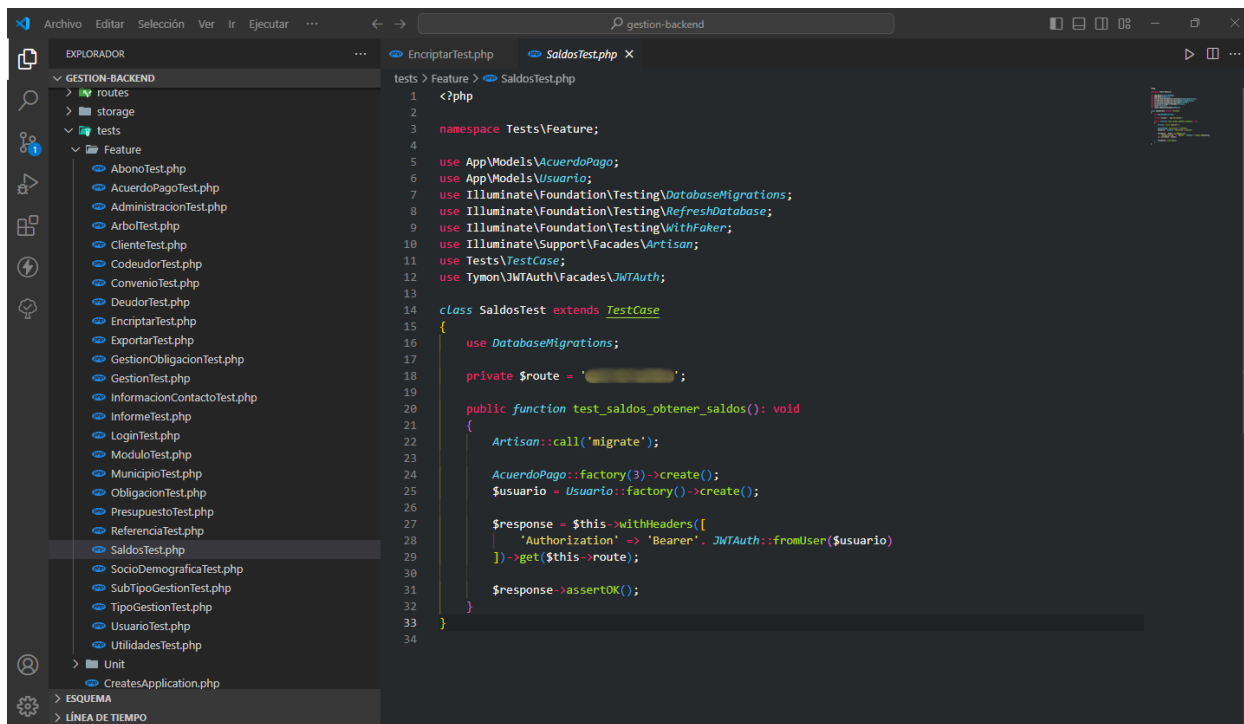


Nota: Elaboración propia

Cada uno de los archivos destinados a pruebas de integración contienen los métodos, el tipo de petición que se realizará y la ruta para realizar las pruebas a cada uno de los endpoints expuestos por el aplicativo, la cual es la encargada de direccionar la información a un método específico el cual contiene las instrucciones de lo que se desea realizar. Adicionalmente la mayoría de estas rutas es necesario estar autenticado, para ello en la cabecera de la petición se establece la autenticación requerida.

Figura 95

Archivo de pruebas de integración para Saldos.



Nota: Elaboración propia

Ejecución de pruebas. Inicialmente al tratar de ejecutar las pruebas definidas, se reporta un problema con las migraciones ya que no tienen coherencia con los modelos que están definidos debido a cambios en las tablas de la base de datos sin hacer uso de dichas migraciones. Para solucionar esto se modifican las migraciones en las tablas las cuales se identifica el conflicto.

Figura 96

Ultima migración registrada para la tabla acuerdo de pagos.

```

1 <?php
2
3 use Illuminate\Support\Facades\Schema;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Database\Migrations\Migration;
6
7 class AddColumnUsuarioIdToAcuerdoPagosTable extends Migration
8 {
9     public $set_schema_table = 'acuerdo_pagos';
10
11     /**
12      * Run the migrations.
13      *
14      * @return void
15      */
16     public function up()
17     {
18         Schema::table($this->set_schema_table, function (Blueprint $table) {
19             $table->double('valor_compromiso')->nullable()->after('valor_cuota_inicial');
20             $table->string('tipo')->nullable()->after('valor_compromiso');
21             $table->string('condonacion_capital')->nullable()->after('tipo');
22             $table->double('porcentaje_condonacion_capital', 8, 2)->nullable()->after('condonacion_capital');
23             $table->dateTime('migrado_app')->nullable()->after('porcentaje_condonacion_capital');
24             $table->unsignedBigInteger('usuario_id')->nullable()->after('migrado_app');
25             sleep(2);
26             $table->foreign('usuario_id')->references('id')->on('usuarios');
27         });
28     }
29
30     /**
31      * Reverse the migrations.
32      *
33      * @return void
34      */

```

Nota: Elaboración propia

Figura 97

Ultima migración registrada para la tabla información de contactos.

```

1 <?php
2
3 use Illuminate\Support\Facades\Schema;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Database\Migrations\Migration;
6
7 class ModifiedInformacionContactosAddFieldMunicipioId extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::table('informacion_contactos', function (Blueprint $table) {
17             $table->bigInteger('municipio_id')->unsigned()->nullable()->after('deleted_at');
18             $table->foreign('municipio_id')->references('id')->on('municipios');
19         });
20     }
21
22     /**
23      * Reverse the migrations.
24      *
25      * @return void
26      */
27     public function down()
28     {
29         Schema::table('informacion_contactos', function (Blueprint $table) {
30             $table->dropForeign(['municipio_id']);
31             $table->dropColumn('municipio_id');
32         });
33     }
34 }

```

Nota: Elaboración propia

En algunos de los controladores, era necesario estar autenticado para obtener un dato en específico, sin embargo, a través de las pruebas unitarias no era posible enviar información sobre autenticación. Por dicha razón se procedió refactorizar partes del Código donde se presentaba este problema, identificando que si se esta ejecutando un ambiente de pruebas reemplace el dato requerido de la autenticación.

Figura 98

Reemplazo de obtención de datos desde un ambiente de pruebas.

```

77 public function index(Request $request)
78 {
79     $modulos = Modulo::with(['children'])
80     ->where(function ($query) use ($request) {
81         $query->where('antecesor', $request->antecesor ?: null);
82         $query->where('show', 1);
83     })
84     ->join('...', '...', '...')
85+    ->where('...', app()->runningUnitTests() ? auth('api')->user()->id : 1)
86     ->orderBy('orden', 'asc')
87     ->get([
88         '...',
89         '...',
90         '...',
91         '...',
92         '...',
93         '...',
94         '...',
95         '...',
96         '...',
97         '...',
98         '...'
99     ]);
100
101     return response()->json($modulos);
102 }

```

Nota: Elaboración propia

Después de realizada las correcciones, se verifico el correcto funcionamiento de las pruebas definidas. En total se generaron 60 archivos destinados para pruebas, 34 de ellos para pruebas unitarias y 26 para pruebas de integración.

Figura 99

Resumen de pruebas.

```

Tests:      232 passed (1027 assertions)
Duration: 4373.14s

vagrant@homestead:~/code/gestion-backend$

```

Nota: Elaboración propia.

3.4 Documentar a detalle el proceso de migración y pruebas.

3.4.1 Documentación general.

Para cumplir con esta actividad se realizó documentación de acuerdo a las actividades anteriormente expuestas. Adicionalmente se documentó los endpoints resultantes del aplicativo.

Tabla 8

Listado de documentos.

Documento.	Descripción.
Análisis de dependencias.	Proporciona información acerca de las dependencias en el aplicativo, donde se divide por la versión de laravel y la dependencia en uso.
Análisis de funcionalidades.	Proporciona una base clara para comprender las características y capacidades del aplicativo.
Colección de rutas.	Contiene un mapeo de todas las rutas existentes del aplicativo, compatible con postman.

Creación de pruebas de integración.	Proporciona información detallada de la configuración, creación y ejecución de pruebas de integración.
Creación de pruebas unitarias.	Proporciona información detallada de la configuración, creación y ejecución de pruebas unitarias.
Fábricas en gestión.	Proporciona información detallada de todas las fábricas creadas para el aplicativo, además de ofrecer orientación para la creación y configuración.
Migración gestión - 5.8 a 6.	Ofrece un análisis exhaustivo de todos los aspectos considerados y los cambios implementados durante el proceso de migración de la versión de Laravel 5.8 a 6.
Migración Gestión - 6.x a 7.	Ofrece un análisis exhaustivo de todos los aspectos considerados y los cambios implementados durante el proceso de migración de la versión de Laravel 6.x a 7.
Migración Gestión - 7.x a 8.	Ofrece un análisis exhaustivo de todos los aspectos considerados y los cambios implementados durante el proceso de migración de la versión de Laravel 7.x a 8.
Migración Gestión - 8.x a 9.	Ofrece un análisis exhaustivo de todos los aspectos considerados y los cambios implementados durante el proceso de migración de la versión de Laravel 8.x a 9.
Migración Gestión - 9.x a 10.	Ofrece un análisis exhaustivo de todos los aspectos considerados y los cambios implementados durante el proceso de migración de la versión de Laravel 9.x a 10.
Pruebas de integración en el aplicativo gestión.	Proporciona un desglose completo de todas las pruebas realizadas, especificando el endpoint que consumen y su método HTTP, además de detallar todas las afirmaciones realizadas sobre las respuestas recibidas e información de configuración para su creación y ejecución.

Pruebas unitarias en el aplicativo gestión.

Proporciona un desglose completo de todas las pruebas unitarias realizadas, especificando si es al modelo o a un método en alguno de los controladores, además de detallar todas las afirmaciones realizadas sobre las respuestas recibidas e información de configuración para su creación y ejecución.

Validación de rutas - Aplicativo gestión

Ofrece una descripción detallada de todas las rutas existentes y preexistentes, incluyendo su método HTTP correspondiente, el endpoint asociado, los datos del cuerpo, las validaciones implementadas, las respuestas generadas y los problemas en ciertas situaciones.

Nota: Elaboración Propia.

Figura 100

Documentos generados.

Nombre	Estado	Tipo	Tamaño
 Análisis de dependencias.pdf	✓ 	Microsoft Edge PDF ...	117 KB
 Análisis de funcionalidades.pdf	✓ 	Microsoft Edge PDF ...	114 KB
 Colección de rutas.json	✓ 	Archivo de origen JS...	78 KB
 Creación de pruebas de integración.pdf	✓ 	Microsoft Edge PDF ...	1.034 KB
 Creación de pruebas unitarias.pdf	✓ 	Microsoft Edge PDF ...	880 KB
 Fábricas en gestión.pdf	✓ 	Microsoft Edge PDF ...	119 KB
 Migración Gestión - 5.8 a 6.pdf	✓ 	Microsoft Edge PDF ...	403 KB
 Migración Gestión - 6.x a 7.pdf	✓ 	Microsoft Edge PDF ...	1.349 KB
 Migración Gestión - 7.x a 8.pdf	✓ 	Microsoft Edge PDF ...	745 KB
 Migración Gestión - 8.x a 9.pdf	✓ 	Microsoft Edge PDF ...	1.328 KB
 Migración Gestión - 9.x a 10.pdf	✓ 	Microsoft Edge PDF ...	525 KB
 Pruebas de integración del aplicativo gestión.pdf	✓ 	Microsoft Edge PDF ...	357 KB
 Pruebas unitarias del aplicativo gestión.pdf	✓ 	Microsoft Edge PDF ...	410 KB
 Validación de rutas - Aplicativo gestión.pdf	✓ 	Microsoft Edge PDF ...	668 KB

FGA

Guía de Funcionamiento de Pruebas de Integración

FGA

Proceso de Creación

Area Tecnología y Operaciones

Operaciones

 Creación de pruebas de integración.pdf

 Compartir

Actividad reciente

 Tú se movió de PDF · 24 de may.

 Tú editó · 24 de may.

 Tú cambió el nombre de Creacion de Pruebas de Integracion.pdf a Creación de pruebas de integración.pdf · 24 de

Nota: Elaboración propia.

4. Diagnóstico final.

El proceso de migración del sistema desde la versión 5.8 a la versión 10 del framework Laravel, junto con el desarrollo de pruebas unitarias y de integración, ha sido un hito significativo en la evolución del sistema.

Se logró completar con éxito la migración del sistema de la versión 5.8 a la versión 10 de Laravel. Este logro representa una actualización importante que proporciona acceso a nuevas características, mejoras de rendimiento y seguridad ofrecidas por la última versión del framework.

Se desarrollaron pruebas unitarias y de integración para garantizar la estabilidad y la funcionalidad del sistema. Estas pruebas permiten identificar y corregir errores de manera proactiva, garantizando la calidad del código y la integridad del sistema.

Durante todo el proceso, se llevó a cabo una exhaustiva documentación de los procesos. Esto incluyó la creación de documentos detallados que describían cada etapa del proyecto. La documentación también abarcó todos los procedimientos específicos y cualquier otro proceso relevante. Esta documentación proporcionará orientación invaluable para futuros proyectos similares y para cualquier personal nuevo que se una al equipo.

Como resultado de este proceso, el sistema ahora se encuentra en una versión más moderna y robusta del framework Laravel, con un conjunto completo de pruebas que respaldan su funcionalidad y documentación a detalle. Esto proporciona una base sólida para futuras iteraciones y mejoras del sistema, así como una mayor confiabilidad y rendimiento para los usuarios finales.

5. Conclusiones.

Al realizar un diagnóstico del estado inicial del aplicativo web, se llevó a cabo un análisis exhaustivo de su arquitectura, código y funcionalidades. Se identificaron áreas de mejora y posibles desafíos para la migración.

Se diseñó un plan detallado que abordó los aspectos técnicos, temporales y de recursos necesarios para la migración del framework. Donde se trató de anticipar posibles conflictos y se establecieron estrategias de mitigación.

Durante la migración, se implementaron cuidadosamente los pasos del plan diseñado con éxito, priorizando la continuidad del funcionamiento del aplicativo. Enfrentando desafíos significativos durante el proceso de migración, como la compatibilidad entre versiones, la actualización de dependencias, la reestructuración del código existente y errores imprevistos. Se realizaron pruebas exhaustivas para garantizar la estabilidad, rendimiento del sistema. Además, se realizó la búsqueda e identificación de brechas de seguridad con ayuda de los analistas del área, para el cierre de dichas brechas y un sondeo final evaluando respuestas. Demostrando habilidad y determinación para superar estos obstáculos y cumplir con el objetivo.

Se elaboró una documentación completa y detallada donde se registró cada fase del proceso de migración, incluyendo los cambios realizados, problemas encontrados, soluciones aplicadas y aspectos importantes. Esta documentación servirá como referencia futura y guía para otros proyectos similares. En última instancia, la documentación de los procesos contribuyó a la transparencia y la coherencia en todo el proyecto.

6. Recomendaciones.

- **Implementar los principios SOLID:** Adoptar los principios SOLID (Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) en la arquitectura de la aplicación para mejorar su mantenibilidad, escalabilidad y flexibilidad.
- **Mejorar el uso de métodos HTTP:** Revisar y corregir el uso actual de los métodos HTTP (GET, POST, PUT, DELETE) para garantizar que se implementen de acuerdo con las mejores prácticas y estándares RESTful.
- **Utilizar códigos de estado HTTP adecuados:** Emplear códigos de estado HTTP más específicos como 201 (Creado) al crear nuevos registros o 204 (Sin contenido) al no encontrar información, en lugar de usar 200 (OK) de forma generalizada en peticiones con respuestas satisfactorias.
- **Refactorizar el código:** Reestructurar el código para eliminar controladores con funciones innecesarias y evitar respuestas afirmativas irrelevantes en el resultado de la solicitud.
- **Minimizar la información en el token:** Reducir la cantidad de datos en el token de autenticación, excluyendo información personal sensible del usuario, para proteger su privacidad y seguridad. El exceso de información puede afectar el rendimiento de la aplicación en casos de redes con baja latencia y genera mayor carga al servidor al procesar el token.

- **Implementar validaciones de datos adecuadas:** Reforzar las validaciones de datos en las solicitudes para garantizar la integridad y confiabilidad de la información procesada.
- **Actualizar las migraciones con prontitud:** Implementar las migraciones del sistema lo antes posible después de cada nueva versión estable del framework para minimizar el riesgo de vulnerabilidades de seguridad.
- **Implementar CI/CD:** Esta práctica automatiza las pruebas de integración y la entrega de cambios de código, lo que mejora la calidad del software al detectar errores tempranamente, acelera la entrega al permitir despliegues automáticos y reduce los riesgos.
- **Evitar modificaciones manuales a la base de datos:** Realizar cambios en la base de datos exclusivamente a través de las migraciones proporcionadas por el framework, asegurando la consistencia y evitando errores durante las pruebas.
- **Actualizar la documentación al realizar cambios:** Mantener la documentación actualizada con cada modificación realizada en el flujo de la aplicación, garantizando que los usuarios y desarrolladores tengan acceso a información precisa y relevante.
- **Fortalecer la seguridad del sistema:** Abordar las vulnerabilidades que expone el sistema, implementando medidas de seguridad robustas para proteger la información confidencial del mismo.
- **Mantenerse a la vanguardia:** Mantenerse actualizado sobre las últimas tendencias y mejores prácticas en desarrollo de software para optimizar continuamente la aplicación.

Referencias.

1. *Assertions—PHPUnit 11.1 Manual*. (s/f). Recuperado el 13 de junio de 2024, de <https://docs.phpunit.de/en/11.1/assertions.html>
- Administración de dependencias | Software supply chain security | Google Cloud*. (s/f). Recuperado el 12 de junio de 2024, de https://cloud.google.com/software-supply-chain-security/docs/dependencies?hl=es-419#whats_next
- AWS | Almacenamiento de datos seguro en la nube (S3)*. (s/f). Recuperado el 13 de junio de 2024, de <https://aws.amazon.com/es/s3/>
- Beaubouef, T. (2003). Why computer science students need language. *Boletín ACM SIGCSE*, 35(4), 51–54. <https://doi.org/10.1145/960492.960525>
- Chauhan, N., Kaur, N., & Saini, K. S. (2023). *Performance Analysis of Rules Generated Hybrid Optimization Algorithm for Resource Allocation and Migration in the Cloud Environment*. 2nd Edition of IEEE Delhi Section Owned Conference, DELCON 2023 - Proceedings. Scopus. <https://doi.org/10.1109/DELCON57910.2023.10127413>
- Conócenos*. (s/f). FGA Fondo de Garantías. Recuperado el 14 de junio de 2024, de <https://fga.com.co/conocenos/>
- D'Ambrosio, N., Melluso, E., Perrone, G., & Romano, S. P. (2023). A Software-Defined Approach for Mitigating Insider and External Threats via Moving Target Defense. 2023 *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 213–219. <https://doi.org/10.1109/NFV-SDN59219.2023.10329613>
- Decreto 1242 de 2021—Gestor Normativo—Función Pública*. (s/f). Recuperado el 13 de junio de 2024, de <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=172306>

- Decreto 1377 de 2013—Gestor Normativo—Función Pública.* (s/f). Recuperado el 13 de junio de 2024, de <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=53646>
- Dehghani, M. H., Kolahdouz-Rahimi, S., Tisi, M., & Tamzalit, D. (2022). Facilitating the migration to the microservice architecture via model-driven reverse engineering and reinforcement learning. *Software and Systems Modeling*, 21(3), 1115–1133. Scopus. <https://doi.org/10.1007/s10270-022-00977-3>
- Dharmaraj, S., & Kavitha, P. (2023). Cloud Network Communication Performance Improvement Using a Stochastic Bandwidth Allocation and Swarm Optimization Algorithm. *Lecture Notes in Networks and Systems*, 467, 113–125. Scopus. https://doi.org/10.1007/978-981-19-2538-2_11
- Documentation for Visual Studio Code.* (s/f). Recuperado el 13 de junio de 2024, de <https://code.visualstudio.com/docs>
- FGA Fondo de Garantías | Mitigamos el riesgo de crédito.* (s/f). Recuperado el 6 de febrero de 2024, de <https://fga.com.co/>
- Framework: Qué es, para qué sirve y algunos ejemplos.* (s/f). UNIR FP. Recuperado el 13 de junio de 2024, de <https://unirfp.unir.net/revista/ingenieria-y-tecnologia/framework/>
- Galster, M., Mitrovic, A., Malinen, S., & Holland, J. (2022). What Soft Skills Does the Software Industry *Really* Want? An Exploratory Study of Software Positions in New Zealand. *Proceedings of the 16th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*, 272–282. <https://doi.org/10.1145/3544902.3546247>
- Gandikota, P. S. S. K., Valluri, D., Mundru, S. B., Yanala, G. K., & Sushaini, S. (2023). Web Application Security through Comprehensive Vulnerability Assessment. *Procedia Computer Science*, 230, 168–182. <https://doi.org/10.1016/j.procs.2023.12.072>

- Gholami, M. F., Daneshgar, F., Low, G., & Beydoun, G. (2016). Cloud migration process—A survey, evaluation framework, and open challenges. *Journal of Systems and Software*, 120, 31–69. <https://doi.org/10.1016/j.jss.2016.06.068>
- Git*. (s/f). Recuperado el 13 de junio de 2024, de <https://git-scm.com/>
- Introduction | Vagrant | HashiCorp Developer*. (s/f). Introduction | Vagrant | HashiCorp Developer. Recuperado el 13 de junio de 2024, de <https://developer.hashicorp.com/vagrant/intro>
- Introduction—Composer*. (s/f). Recuperado el 13 de junio de 2024, de <https://getcomposer.org/doc/00-intro.md>
- Jamshidi, P., Ahmad, A., & Pahl, C. (2014). Cloud Migration Research: A Systematic Review. *IEEE Transactions on Cloud Computing*, 1, 142–157. <https://doi.org/10.1109/TCC.2013.10>
- Kiesel, J., Hubricht, F., Stein, B., & Potthast, M. (2020). A dataset for content error detection in web archives. *Proceedings of the 18th Joint Conference on Digital Libraries*, 349–350. <https://doi.org/10.1109/JCDL.2019.00065>
- Koteska, B., & Mishev, A. (2014). Change Management and Version Control of Scientific Applications. *International Journal of Computer Science & Information Technology (IJCSIT)*, 6, 153–161. <https://doi.org/10.5121/ijcsit.2014.6211>
- Lalitha Devi, K., Deepa Thilak, K., Kalaiselvi, K., & Arthi, K. (2023). Workload Prediction for Resource Scaling and Migration in the Cloud. *Lecture Notes in Networks and Systems*, 615 LNNS, 177–186. Scopus. https://doi.org/10.1007/978-981-19-9304-6_18
- Laravel—The PHP Framework For Web Artisans*. (s/f). Recuperado el 12 de junio de 2024, de <https://laravel.com/>

- Ley 23 de 1982—Gestor Normativo—Función Pública.* (s/f). Recuperado el 13 de junio de 2024, de <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=3431>
- Ley 527 de 1999—Gestor Normativo—Función Pública.* (s/f). Recuperado el 13 de junio de 2024, de <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=4276>
- Ley 1266 de 2008—Gestor Normativo—Función Pública.* (s/f). Recuperado el 13 de junio de 2024, de <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=34488&0>
- Ley 1273 de 2009—Gestor Normativo—Función Pública.* (s/f). Recuperado el 13 de junio de 2024, de <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=34492>
- Ley 1341 de 2009—Gestor Normativo—Función Pública.* (s/f). Recuperado el 13 de junio de 2024, de <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=36913>
- Ley 1581 de 2012—Gestor Normativo—Función Pública.* (s/f). Recuperado el 13 de junio de 2024, de <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981>
- Ley 2157 de 2021—Gestor Normativo—Función Pública.* (s/f). Recuperado el 13 de junio de 2024, de <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=173246>
- Leyes desde 1992—Vigencia expresa y control de constitucionalidad [CONSTITUCION_POLITICA_1991].* (s/f). Recuperado el 13 de junio de 2024, de http://www.secretariasenado.gov.co/senado/basedoc/constitucion_politica_1991.html
- Mahmood, M. T., & Ashour, O. I. A. (2020). Web Application Based on MVC Laravel Architecture for Online Shops. *Proceedings of the 6th International Conference on Engineering & MIS 2020*, 1–7. <https://doi.org/10.1145/3410352.3410834>
- Matalonga, S., Amalfitano, D., Doreste, A., Fasolino, A. R., & Travassos, G. H. (2022). Alternatives for testing of context-aware software systems in non-academic settings:

- Results from a Rapid Review. *Information and Software Technology*, 149, 106937.
<https://doi.org/10.1016/j.infsof.2022.106937>
- McCall, D., & Kölling, M. (2019). A New Look at Novice Programmer Errors. *ACM Transactions on Computing Education*, 19(4), 38:1-38:30.
<https://doi.org/10.1145/3335814>
- Niranjan, Y., Narayana, M. V., & Vijaya Sudha, M. (2023). Virtual Machine Migration Framework with Configuration Change Management. *Lecture Notes in Networks and Systems*, 565 LNNS, 633–644. Scopus. https://doi.org/10.1007/978-981-19-7455-7_49
- Node.js—Introduction to Node.js*. (s/f). Recuperado el 13 de junio de 2024, de <https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>
- Nvm-sh/nvm*. (2024). [Shell]. nvm.sh. <https://github.com/nvm-sh/nvm> (Obra original publicada en 2010)
- Oracle VM VirtualBox*. (s/f). Recuperado el 13 de junio de 2024, de <https://www.virtualbox.org/>
- Organigrama FGA*. (s/f). Recuperado el 6 de febrero de 2024, de [https://fgamed.sharepoint.com/sites/FGA-Intranet/SitePages/Organigrama\(1\).aspx](https://fgamed.sharepoint.com/sites/FGA-Intranet/SitePages/Organigrama(1).aspx)
- Overview | TablePlus Documentation*. (2021, septiembre 12). <https://docs.tableplus.com>
- Packagist*. (s/f). Recuperado el 13 de junio de 2024, de <https://packagist.org/>
- PHP: ¿Qué es PHP? - Manual*. (s/f). Recuperado el 13 de junio de 2024, de <https://www.php.net/manual/es/intro-what-is.php>
- Que es el código abierto - Explicación del código abierto—AWS*. (s/f). Recuperado el 12 de junio de 2024, de <https://aws.amazon.com/es/what-is/open-source/>
- Que es el Software Libre - Proyecto GNU - Free Software Foundation*. (s/f). Recuperado el 12 de junio de 2024, de <https://www.gnu.org/philosophy/free-sw.es.html>

Que es la migración de aplicaciones | IBM. (s/f). Recuperado el 13 de junio de 2024, de <https://www.ibm.com/es-es/topics/application-migration>

Que es MySQL (s/f). Recuperado el 13 de junio de 2024, de <https://www.oracle.com/co/mysql/what-is-mysql/>

Que es un proveedor de servicios en la nube | Google Cloud. (s/f). Google Cloud. Recuperado el 13 de junio de 2024, de <https://cloud.google.com/learn/what-is-a-cloud-service-provider?hl=es-419>

Que es una base de datos | Oracle Colombia. (s/f). Recuperado el 13 de junio de 2024, de <https://www.oracle.com/co/database/what-is-database/>

Que son las pruebas de software | IBM. (s/f). Recuperado el 12 de junio de 2024, de <https://www.ibm.com/es-es/topics/software-testing>

Que son los repositorios - Explicación sobre los repositorios - AWS. (s/f). Amazon Web Services, Inc. Recuperado el 13 de junio de 2024, de <https://aws.amazon.com/es/what-is/repo/>

Salahirad, A., Gay, G., & Mohammadi, E. (2023). Mapping the structure and evolution of software testing research over the past three decades. *Journal of Systems and Software*, 195, 111518. <https://doi.org/10.1016/j.jss.2022.111518>

Schmiedmayer, P., Bauer, A., & Bruegge, B. (2023). *Reducing the Impact of Breaking Changes to Web Service Clients During Web API Evolution*. 1–11. Scopus. <https://doi.org/10.1109/MOBILSoft59058.2023.00008>

Sobhani, R., Seifzadeh, H., & Gandomani, T. J. (2018). A review of migration processes to open source software. *International Journal of Open Source Software and Processes*, 9(1), 20–31. Scopus. <https://doi.org/10.4018/IJOSSP.2018010102>

The most-comprehensive AI-powered DevSecOps platform | GitLab. (s/f). Recuperado el 13 de junio de 2024, de <https://about.gitlab.com/>

Volynsky, E., Mehmed, M., & Krusche, S. (2022). *Architect: A Framework for the Migration to Microservices*. 71–76. Scopus. <https://doi.org/10.1109/iCCECE55162.2022.9875096>

What is Postman? Postman API Platform. (s/f). Recuperado el 13 de junio de 2024, de <https://www.postman.com/product/what-is-postman/>