

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



“Proyecto de Unidad 3 - Gestión de Viajes”

Integrantes:

Nº	Código Universitario	Apellidos y Nombres
1.-	2017057528	Ccalli Chata, Joel Robert
2.-	2020067145	Anahua Coaquira, Mayner Gonzalo
3.-	2020069046	Salinas Condori, Erick Javier
4.-	2020066924	Zevallos Purca, Justin Zinedine

CURSO : “Base de Datos II”

SECCION : “A”

DOCENTE : Ing. “Patrick Jose Cuadros Quiroga”

Tacna - Perú

2023

**Desarrollo de un Sistema de Gestión de Datos en la
Nube para Mejorar la Eficiencia Operativa y Muestreo
de datos del Sistema
FD05 - Informe Final**

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	JZ, MA,ES	PC	PC	09/12/2023	Versión Original
2.0	JZ, MA,ES	PC	PC	13/12/2023	Revisión Final

INDICE GENERAL

1. Antecedentes	
2. Planteamiento del Problema	4
a. Problema	
b. Alcance	
3. Objetivos	6
4. Marco Teórico	
5. Desarrollo de la Solución	9
a. Análisis de Factibilidad (técnico, económica, operativa, social, legal, ambiental)	
b. Tecnología de Desarrollo	
c. Metodología de implementación	
6. Cronograma	11
7. Presupuesto	12
8. Conclusiones	13
Recomendaciones	14
Bibliografía	15

1. Antecedentes

Se han encontrado varios trabajos de investigación en los cuales se ha aplicado el control de asistencia o tareas relacionadas, a continuación, se presentan algunos:

a.- Sistemas de Reservas Corporativas:

Muchas empresas utilizan sistemas de reservas corporativas personalizadas o plataformas como Concur para gestionar y rastrear los viajes de negocios de los empleados.

b.- Plataformas de Gestión de Viajes en Línea:

Hay varias plataformas en línea que permiten a los usuarios reservar y gestionar sus propios viajes. Ejemplos incluyen Booking.com, Expedia for Business y Airbnb for Work.

c.- Sistemas de Información Geográfica (GIS):

Para la planificación de rutas y la gestión logística, los sistemas GIS como Google Maps Platform y Mapbox pueden ser integrados en soluciones de gestión de viajes.

d.- Soluciones de Informes y Gastos:

Herramientas como SAP Concur ofrecen funcionalidades de informes y seguimiento de gastos que son esenciales para la gestión eficiente de viajes de negocios.

e.- Plataformas de Gestión de Eventos:

Empresas que organizan eventos a menudo utilizan plataformas como Cvent o Eventbrite para gestionar la logística de viajes asociada a esos eventos.

2. Planteamiento del Problema

a. *Problema*

En el contexto de desarrollo de aplicaciones modernas, la gestión eficiente de viajes requiere una solución tecnológica que integre diversas funcionalidades de forma cohesiva y automatizada. El proyecto en cuestión aborda la necesidad de crear una API robusta para la gestión de viajes que no solo administre las operaciones propias del negocio, sino que también se enlace de manera eficiente con otros servicios esenciales como la gestión de usuarios, configuración, notificaciones y bitácora. Cada uno de estos servicios ha sido desarrollado por grupos diferentes, lo que introduce un desafío de integración para que operen en un ecosistema unificado.

El despliegue de estos servicios en la nube introduce complejidades adicionales, como la necesidad de automatización en los procesos de CI/CD, la monitorización y gestión

de métricas para asegurar una alta disponibilidad y rendimiento, así como la escalabilidad para manejar fluctuaciones en la demanda. Además, la persistencia y consistencia de datos entre servicios distribuidos, y la gestión de configuraciones en diferentes entornos son aspectos críticos que requieren una solución efectiva.

El problema central es, por lo tanto, cómo diseñar e implementar una arquitectura de microservicios que permita la integración y automatización óptimas de todos estos servicios individuales, manteniendo la calidad, seguridad y eficiencia del sistema global al ser desplegado en la nube.

b. *Alcance*

El proyecto tiene como objetivo el diseño e implementación de una arquitectura de microservicios para una aplicación de gestión de viajes en la nube. El alcance específico del proyecto incluye:

Desarrollo de una API de Gestión de Viajes: Crear y documentar una API robusta que proporcione funcionalidades para la reserva y gestión de viajes, utilizando principios RESTful y especificaciones OpenAPI.

Integración de Servicios Complementarios: Incorporar servicios adicionales como gestión de usuarios, configuración, notificaciones y bitácora, cada uno con su propia base de datos y lógica de negocio.

Automatización de Despliegue: Configurar y validar un pipeline de CI/CD para el despliegue automático de los microservicios en un entorno de nube, utilizando Docker Compose para la orquestación.

Monitoreo y Métricas: Establecer un sistema de monitoreo utilizando herramientas como Prometheus y Grafana para recoger, almacenar y visualizar métricas en tiempo real.

Escalabilidad y Rendimiento: Asegurar que cada microservicio sea escalable de forma independiente y que el sistema en conjunto mantenga un alto rendimiento bajo diferentes condiciones de carga.

Seguridad: Implementar medidas de seguridad para proteger la aplicación, incluyendo la autenticación y autorización de usuarios, así como la seguridad en la comunicación entre servicios.

Documentación: Proporcionar documentación técnica detallada para cada uno de los servicios, así como guías de usuario y administración del sistema.

Pruebas: Diseñar y ejecutar un plan de pruebas completas que incluya pruebas unitarias, de integración, de carga y de seguridad para garantizar la robustez del sistema.

Entrega: El sistema debe estar completamente operativo y desplegado en la nube con todos los servicios funcionando de manera integrada para la fecha [insertar fecha de finalización].

3. Objetivos

a. *Objetivo general*

- i. Desarrollar e implementar una solución integral y automatizada de microservicios en la nube para la gestión de viajes, que permita una interacción eficaz y segura entre diversos servicios especializados, asegurando una experiencia de usuario óptima, alta disponibilidad, escalabilidad y un rendimiento consistente del sistema.

b. *Objetivos Específicos*

- Diseñar la Arquitectura de Microservicios: Crear un diseño detallado de la arquitectura que defina cómo los diferentes servicios interactuarán y se integrarán entre sí.
- Implementar la API de Gestión de Viajes: Codificar y desplegar una API robusta que maneje las operaciones de reserva, consulta y cancelación de viajes.
- Desarrollar Servicios de Soporte: Construir servicios complementarios que incluyan gestión de usuarios, configuración y notificaciones, asegurando su correcta integración con la API principal de viajes.

- **Automatizar el Proceso de Despliegue:** Establecer un pipeline de CI/CD que permita el despliegue automático y la integración continua de los microservicios en la plataforma de nube elegida.
- **Configurar Monitoreo y Métricas:** Integrar un sistema de monitoreo con Prometheus y Grafana para recolectar, almacenar y visualizar métricas clave del rendimiento del sistema.
- **Validar la Escalabilidad del Sistema:** Realizar pruebas de carga para asegurar que cada microservicio pueda escalar adecuadamente en respuesta a las demandas del sistema.
- **Garantizar la Seguridad del Sistema:** Implementar mecanismos de autenticación, autorización y cifrado para proteger la información y las operaciones del sistema.
- **Producir Documentación Completa:** Elaborar documentación técnica para desarrolladores y manuales para usuarios finales y administradores del sistema.
- **Realizar Pruebas Exhaustivas:** Llevar a cabo pruebas unitarias, de integración y de sistema para asegurar la fiabilidad y estabilidad del sistema completo.
- **Asegurar la Calidad del Servicio:** Establecer protocolos de calidad de servicio (QoS) para mantener la experiencia del usuario y la satisfacción del cliente.

4. Marco Teórico

Grafana: es una plataforma de código abierto utilizada para la visualización y análisis de datos en tiempo real. Se especializa en la creación de paneles interactivos y dashboards que permiten monitorear y analizar métricas provenientes de diversas fuentes de datos. Grafana es ampliamente utilizado en entornos de monitoreo y observabilidad para presentar datos de manera visualmente atractiva y comprensible.

Prometheus: es un sistema de monitorización y alerta de código abierto diseñado para supervisar sistemas y servicios en entornos informáticos modernos. Fue creado originalmente por SoundCloud y más tarde donado a la Cloud Native Computing Foundation (CNCF). Prometheus se ha convertido en una herramienta popular en el

ecosistema de la nube y es especialmente conocido por su capacidad para el monitoreo de sistemas distribuidos y contenedores.

Modelo de Datos Multidimensional: Prometheus utiliza un modelo de datos multidimensional para representar las series temporales de métricas. Cada métrica puede estar etiquetada con pares clave-valor, lo que permite una gran flexibilidad en la consulta y análisis de datos.

Recopilación de Métricas mediante Pull: Prometheus sigue un modelo de recopilación de datos mediante "pull", lo que significa que **los servidores Prometheus consultan activamente las métricas a intervalos regulares** desde los objetivos que están supervisando. Esto facilita la monitorización de sistemas que pueden ser efímeros o dinámicamente escalables, como los contenedores.

Lenguaje de Consulta PromQL: Prometheus proporciona un lenguaje de consulta llamado PromQL, que permite realizar consultas complejas sobre las métricas almacenadas. Esto facilita la generación de alertas y la creación de gráficos personalizados.

Almacenamiento Local de Datos: Prometheus almacena los datos localmente en su propio formato de almacenamiento de series temporales. Esto facilita el análisis retrospectivo y la visualización de datos históricos.

Alertas Integradas: Prometheus tiene un sistema de alertas integrado que permite definir reglas de alerta basadas en las métricas recopiladas. Puede enviar notificaciones cuando se detectan condiciones anómalas o fuera de los límites establecidos.

Integración con Otros Componentes: Es común usar Prometheus junto con Grafana para la visualización de datos y Alertmanager para la gestión de alertas. Además, es parte del ecosistema de herramientas de la CNCF y se integra fácilmente con otras tecnologías de la nube.

5. Desarrollo de la Solución

1. *Estudio de Factibilidad*

En las siguientes secciones, detallaremos cada una de las áreas de factibilidad para proporcionar una evaluación completa de la viabilidad del proyecto propuesto.

1.1 Factibilidad Técnica

En esta sección, detallaremos las especificaciones de los equipos donde se desplegará la aplicación, la empresa dispone de 3 equipos con característica de gama media.

RECURSOS TECNICOS		
Tipo de Recurso	Nombre del recurso	Descripción
Hardware	PC (Personal Computer)	Escritorio
		Intel Core i7 - 1.8GHz
		8 GB de RAM recomendados (4 GB mínimo).
		240 GB de espacio en disco duro
Software	Windows 10 64 bits	Sistema operativo

Recursos de Despliegue		
Tipo de recurso	Nombre	Descripción
Prometheus	Metricas mediante Pull	consultan activamente las métricas a intervalos regulares desde los objetivos que están supervisando
	Modelo Multidimensional	Prometheus utiliza un modelo de datos multidimensional para representar las series temporales de métricas

1.2 Factibilidad Económica

La factibilidad económica es un componente crucial en la evaluación de cualquier proyecto, ya que permite analizar y determinar la viabilidad financiera de llevar a cabo una iniciativa específica. En esta etapa, se lleva a cabo un exhaustivo análisis de los gastos asociados con el desarrollo y la implementación del proyecto en cuestión.

1.2.1 Costos Generales

Costos Generales			
Hardware	Cantidad	Precio / Unidad	Precio Total
Impresora	1	600	600
Router	1	80	80
Equipos de Oficina (alquiler)	3	150	450
Software	Cantidad	Precio / Unidad	Precio Total
Licencias	1	50	50
Otros	Cantidad	Precio / Unidad	Precio Total
Cartuchos de Impresora	2	59	118
Marcadores	1	9	9
Papeles	2	30	60
Total			S/. 1367

1.2.2 Costos operativos durante el desarrollo

Elemento	Meses	Precio.	Coste (soles)
Luz	3	60	S/. 180
Internet	3	60	S/. 180
Agua	3	35	S/. 105
TOTAL			S/. 465

1.2.3 Costos de ambiente

Hardware - Software	Descripción	Cantidad	Costo Total
Licencias Windows	Windows 10 Profesional	3	S/57.00
Antivirus	Malware bytes	3	S/. 27.00
		Total	S/. 84.00

1.2.4 Costos de personal

Perfil requerido / Puesto	Costo por hora	Horas	Cantidad Personal	Total Mes	Subtotal
Director de Proyecto	S/5	80 horas	1	3	S/.1200
Analista y diseñador	S/5	80 horas	1	3	S/.1200
Desarrollador	S/5	80 horas	1	3	S/.1200
Total					S/3600

1.2.5 Costos totales de desarrollo de sistema

Tipos de Costo	Subtotal (S/)
Costos Generales	S/. 1367
Costos Operativos durante el desarrollo	S/. 465
Costos del ambiente	S/. 84.00
Costo del Personal	S/. 3600
TOTAL	S/. 5516

1.3 Factibilidad Operativa

La factibilidad operativa de este proyecto requiere la disponibilidad de un equipo de recursos humanos con las habilidades y experiencia necesarias para el desarrollo, implementación y mantenimiento continuo del sistema. Además, será fundamental brindar capacitación adecuada al personal de la óptica para garantizar un uso efectivo de Dominio de Aplicativo Web.

En términos de recursos tecnológicos, la infraestructura tecnológica, que incluye computadoras, bases de datos y herramientas de seguridad informática, debe ser suficiente y confiable para mantener un funcionamiento constante y seguro del sistema. Esto garantiza que el proyecto pueda operar de manera eficiente.

El proceso operativo, desde el registro de datos financieros hasta la generación de informes, debe estar claramente definido y documentado. Un flujo de trabajo eficiente es esencial para garantizar una operación sin problemas para que siga los procedimientos establecidos.

La evaluación de riesgos es esencial para identificar posibles obstáculos operativos a lo largo del proyecto, y se deben desarrollar estrategias de contingencia para abordar estos riesgos y garantizar la continuidad operativa.

Además, se debe considerar la sostenibilidad y el mantenimiento a largo plazo del sistema. Esto implica la gestión de actualizaciones, correcciones de errores y adaptación del sistema a las cambiantes necesidades. La viabilidad de mantener el sistema de manera sostenible en términos de recursos humanos y financieros es un aspecto crítico de la factibilidad operativa.

1.4 Factibilidad Legal

La factibilidad legal es muy importante en el desarrollo y mantención del software, es por esto por lo que la ley N° 822 (LEY SOBRE EL DERECHO DE AUTOR) y la ley N° 29733 (Ley de protección de datos personales) son los pilares fundamentales en la construcción del software y para la protección de este.

Ley 29733 :

- Esta ley se aplica a todas las entidades públicas y privadas que tratan datos personales en el territorio peruano.
- La ley define términos clave relacionados con la protección de datos, como datos personales, tratamiento de datos, titular de datos, entre otros.
- La ley establece principios fundamentales que deben regir el tratamiento de datos personales, incluyendo el principio de consentimiento, finalidad, calidad, proporcionalidad, seguridad y confidencialidad.
- La ley establece sanciones para las entidades que no cumplan con sus disposiciones, que pueden incluir multas y otras medidas.

Estas leyes tienen por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor, intimidad y privacidad personal y familiar.

1.5 Factibilidad Social

La factibilidad social del proyecto del Dominio de Aplicativo Web implica considerar cómo beneficiará a la comunidad local y a usuarios del sistema. El proyecto tiene el potencial de mejorar la estabilidad económica. También puede aumentar la satisfacción de los clientes al ofrecer servicios más eficientes. Sin embargo, se deben abordar posibles resistencias al cambio entre el personal para garantizar una transición exitosa.

1.6 Factibilidad Ambiental

El equipo de trabajo se regirá por normas ISO relevantes para garantizar la calidad y la sostenibilidad del proyecto. Entre las normas ISO que podrían aplicarse están:

ISO 14001: Esta norma se enfoca en sistemas de gestión ambiental. Ayudaría al equipo a establecer un marco para identificar y controlar los aspectos ambientales, reduciendo el impacto del proyecto en el medio ambiente.

ISO 50001: Si bien esta norma se relaciona principalmente con sistemas de gestión de la energía, puede ser relevante para fomentar un uso eficiente de la energía durante el proceso de desarrollo y operación.

En conjunto, el sistema de gestor de finanzas se desarrollará y operará con un compromiso continuo hacia la sostenibilidad ambiental, buscando lograr resultados positivos tanto para los usuarios como para el entorno en el que se implementa.

a. *Tecnología de Desarrollo*

El proyecto se está realizando con las siguientes herramientas: Prometheus ,
Postman, Grafana y entre otros servicios de despliegue de desarrollo.

b. *Metodología de implementación*

Para este proyecto se está usando la metodología RUP.

6.-Cronograma

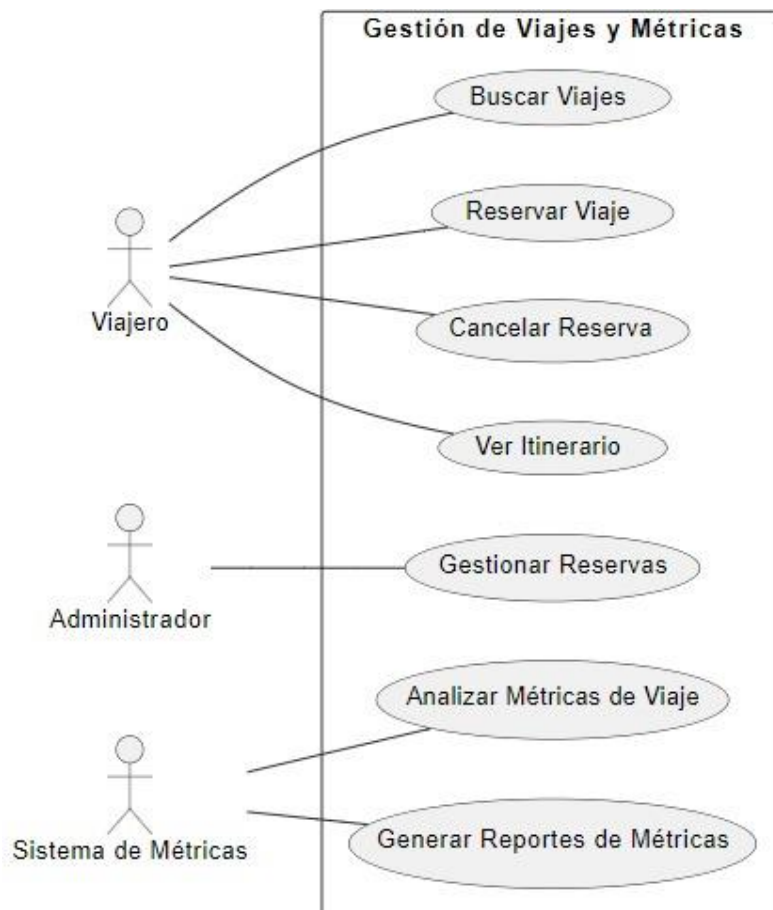
[illegible]

7. Presupuesto

A continuación, se muestra los costos totales del desarrollo del sistema que sería nuestro presupuesto

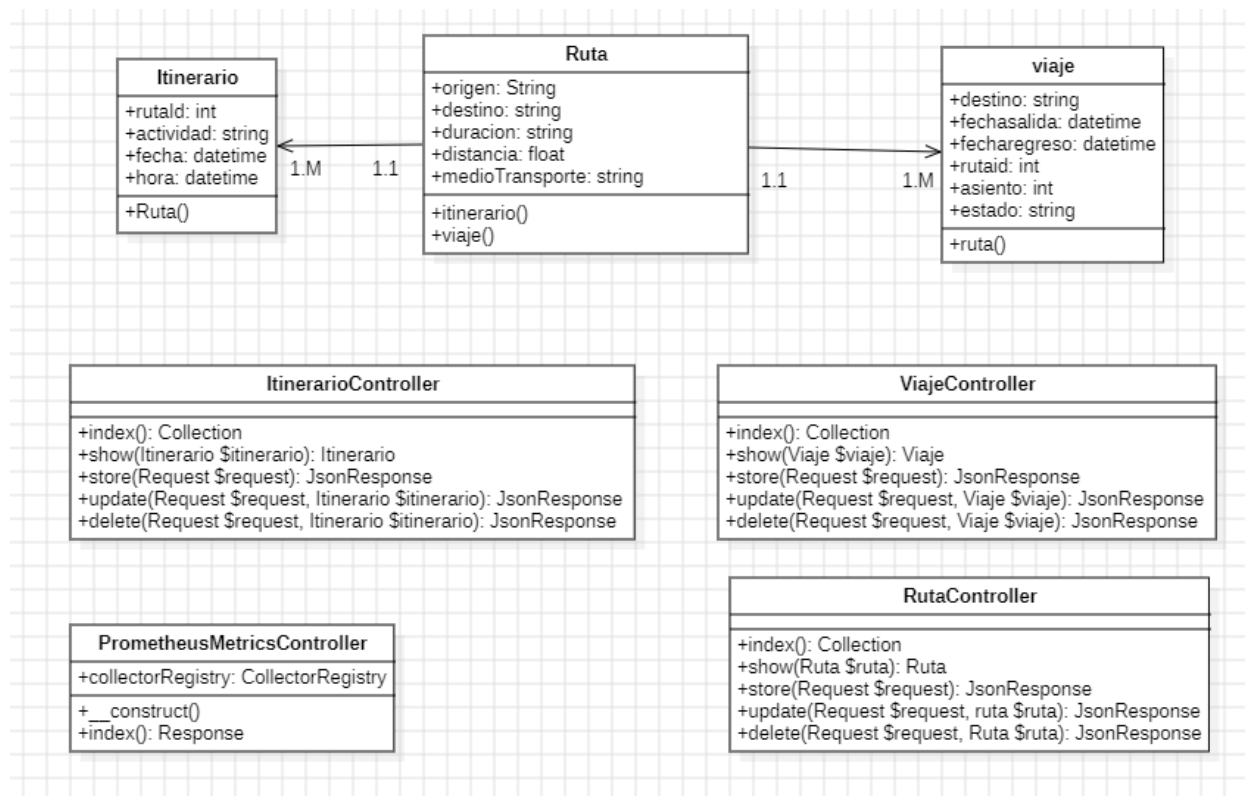
Tipos de Costo	Subtotal (S/)
Costos Generales	S/. 1367
Costos Operativos durante el desarrollo	S/. 465
Costos del ambiente	S/. 84.00
Costo del Personal	S/. 3600
TOTAL	S/. 5516

Diagrama de Casos de Uso

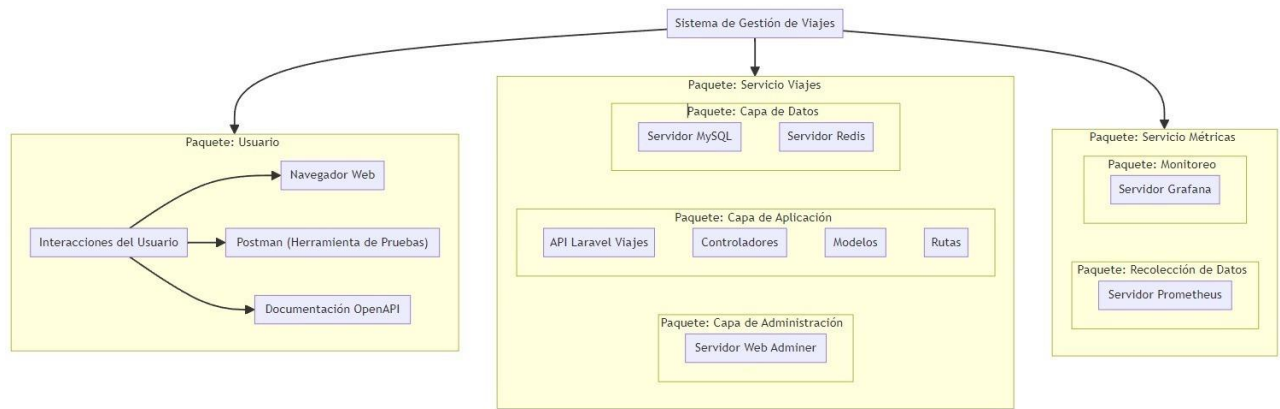


6.2. Diagrama de Clases

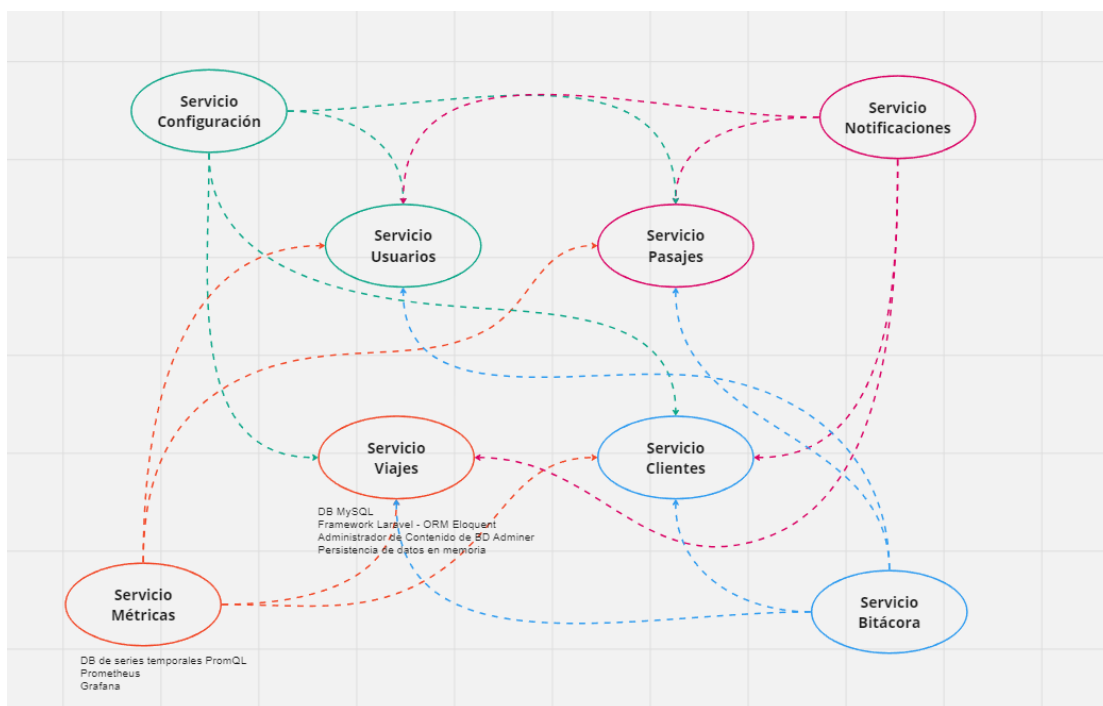
Gestión de Viajes:

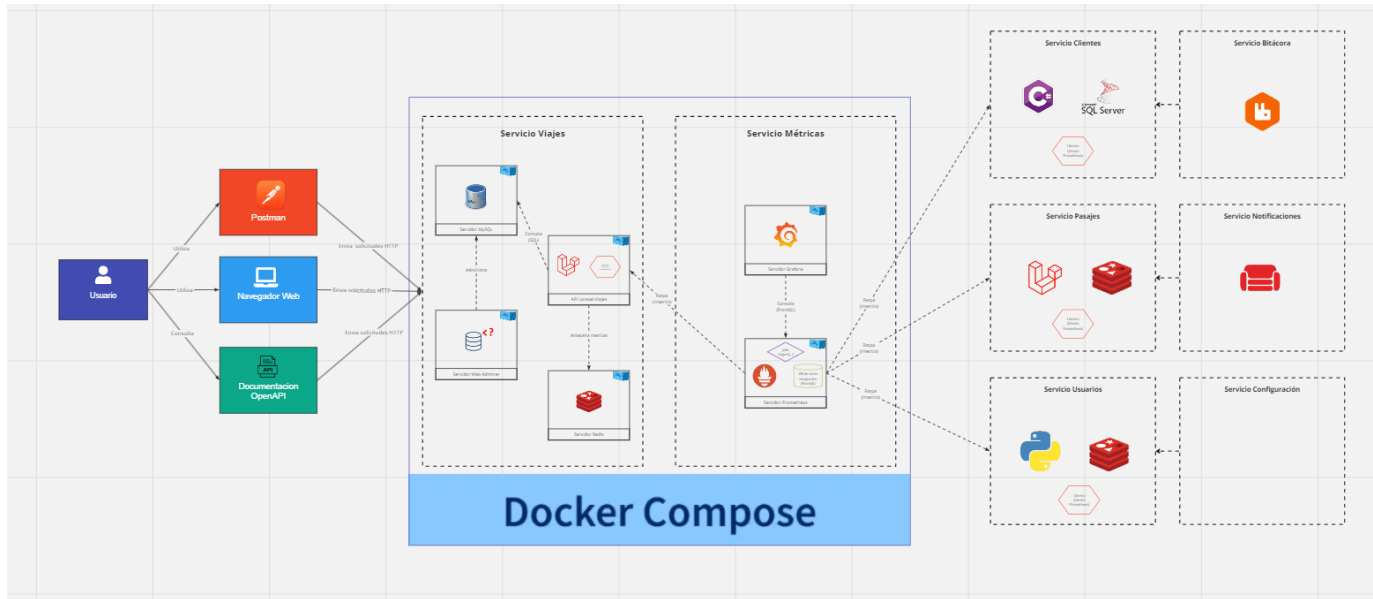


6.3. Diagrama de Componentes



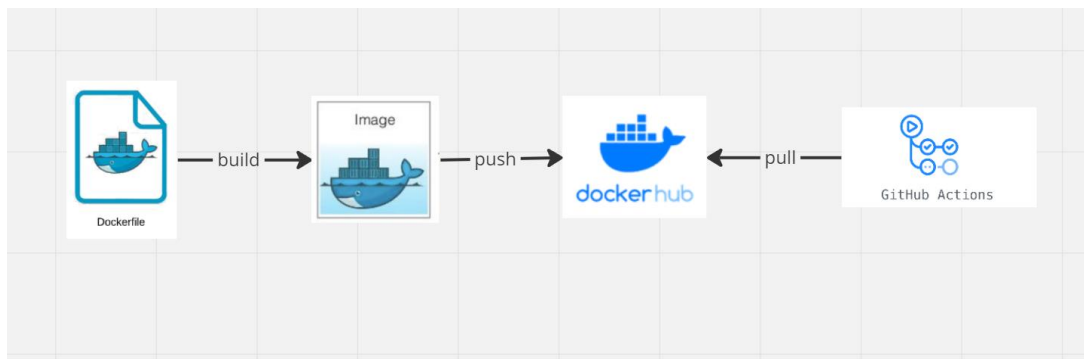
6.4. Arquitectura.





https://miro.com/app/board/uXjVMkhxs94=/?share_link_id=526151654638

6.5. Despliegue automatizado a un registro de contenedores “DockerHub” con “GitHub Actions”



api-viajes-metricas-u3Public

PinUnwatch1Fork0Star0

main1 Branch0 TagsGo to fileAdd fileCode

MArnerACdocker-push-v2.0ca566af · 9 hours ago3 Commits

.github/workflows	docker-push-v1.2	2 days ago
gestion_viajes	docker-push-v2.0	9 hours ago
README.md	docker-push-v1.1	2 days ago
docker-push.txt	docker-push-v1	2 days ago

README

api-viajes-metricas-u3

VIAJES - METRICAS

About

No description, website, or topics provided.

ReadmeActivity0 stars1 watching0 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages

PHP77.5%Blade21.8%Other0.7%

MArnerAC / api-viajes-metricas-u3

CodeIssuesPull requestsActionsProjectsWikiSecurityInsightsSettings

Actions

New workflow

All workflows

Showing runs from all workflows

4 workflow runs

	Event	Status	Branch	Actor
docker-push-v2.0 deploy #4: Commit ca566af pushed by MArnerAC	push	Succeeded	main	MArnerAC
docker-push-v1.2 deploy #5: Commit 861317c pushed by MArnerAC	push	Succeeded	main	MArnerAC
docker-push-v1.1 deploy #2: Commit 52b372d pushed by MArnerAC	push	Failed	main	MArnerAC
docker-push-v1 deploy #1: Commit 563032f pushed by MArnerAC	push	Failed	main	MArnerAC

← deploy

docker-push-v1.2 #3

Summary

Jobs

Deploy

Run details

Usage

Workflow file

Deploy

succeeded 2 days ago in 1m 15s

Set up job

Run actions/checkout@v2

Login

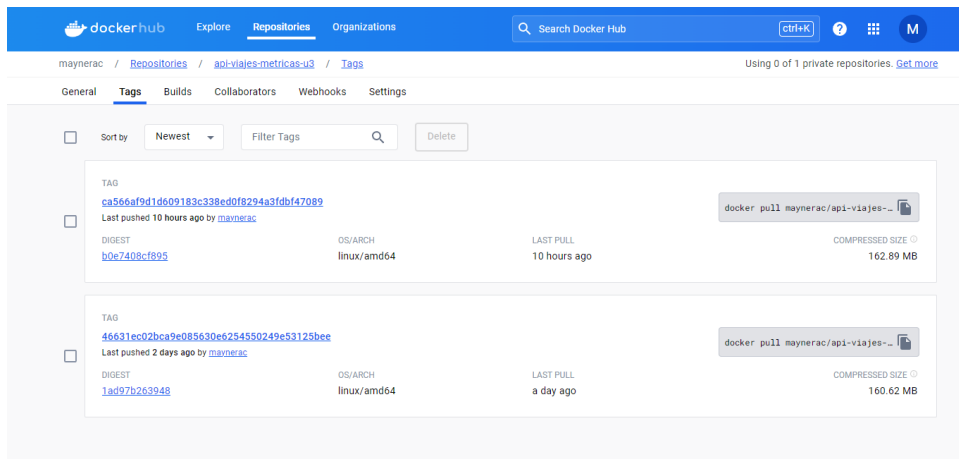
Build and push to Docker Hub

Post Build and push to Docker Hub

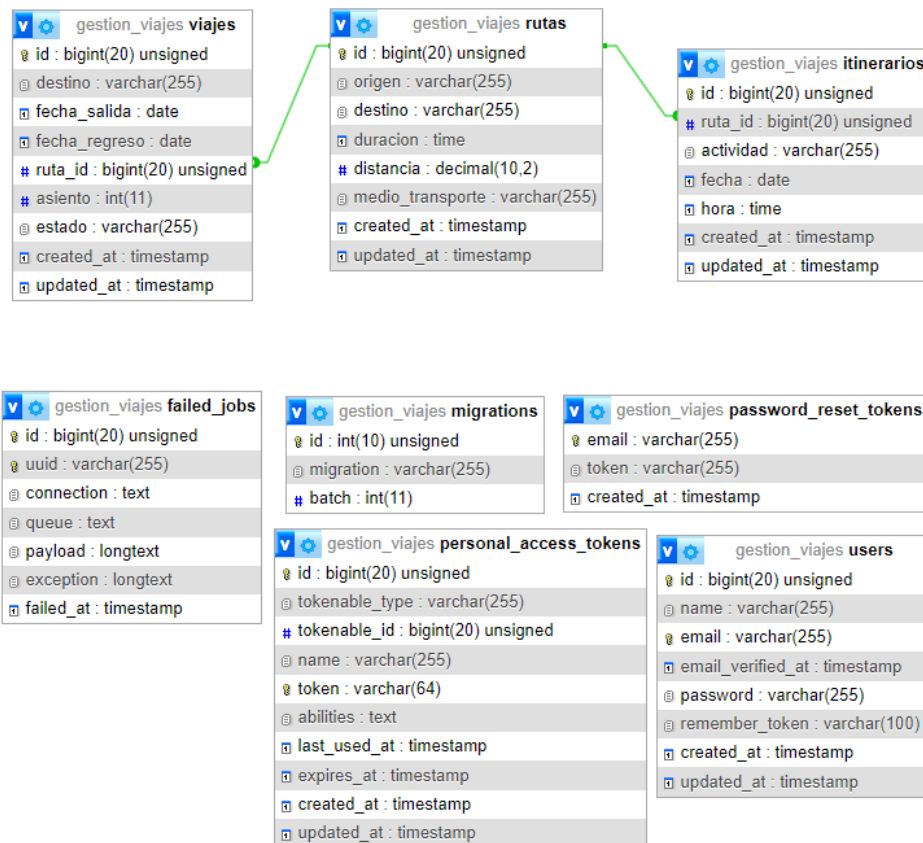
Post Login

Post Run actions/checkout@v2

Complete job



6.6. Diagrama de Base de Datos



8. Conclusiones

- Eficiencia Operativa y Ahorro de Recursos:** La implementación de un sistema web de gestión de viajes se traduce directamente en una mayor eficiencia operativa al reducir la carga administrativa manual, minimizar errores y agilizar los flujos de trabajo. Esto no

solo ahorra tiempo para los usuarios y administradores, sino que también resulta en un uso más eficiente de los recursos de la empresa.

- **Mejora Sostenida de la Experiencia del Usuario:** La combinación de una planificación simplificada, información en tiempo real y opciones de personalización en un sistema bien diseñado proporciona una mejora constante en la experiencia del usuario. Esta mejora continua no solo satisface las necesidades actuales de los usuarios, sino que también fomenta una mayor adopción y compromiso con el sistema a lo largo del tiempo.
- **Decisiones Estratégicas Informadas:** La recopilación y análisis de datos estratégicos permiten a los usuarios y administradores tomar decisiones informadas. La plataforma brinda visibilidad sobre itinerarios y gastos de viaje, facilitando la identificación de patrones, la optimización de costos y la mejora continua en la eficiencia de la gestión de viajes, contribuyendo así a la toma de decisiones estratégicas.

9. Recomendaciones

Implementación Gradual y Formación Efectiva: Para maximizar la eficiencia operativa, se recomienda una implementación gradual del sistema web de gestión de viajes, asegurándose de que los usuarios y administradores reciban una formación efectiva. Esto garantizará una transición suave y una rápida adaptación a las nuevas herramientas y procesos, reduciendo la resistencia al cambio.

Personalización Intuitiva y Continua: Para mantener una mejora sostenida en la experiencia del usuario, es esencial ofrecer opciones de personalización intuitivas y continuar actualizando el sistema según las necesidades y comentarios de los usuarios. La retroalimentación constante asegurará que el sistema siga siendo relevante y satisfaga las expectativas cambiantes de los usuarios.

Acceso Seguro y Protección de Datos: Dado que la plataforma involucra datos sensibles, se recomienda implementar medidas de seguridad robustas para garantizar el acceso seguro y la protección de la información. Esto incluye prácticas de cifrado, autenticación multifactor y auditorías regulares para mantener la integridad de los datos y la confianza de los usuarios.

10. Bibliografía

WebGrafía:

<https://keepcoding.io/blog/que-son-metricas-de-prometheus/>

WebGrafía:

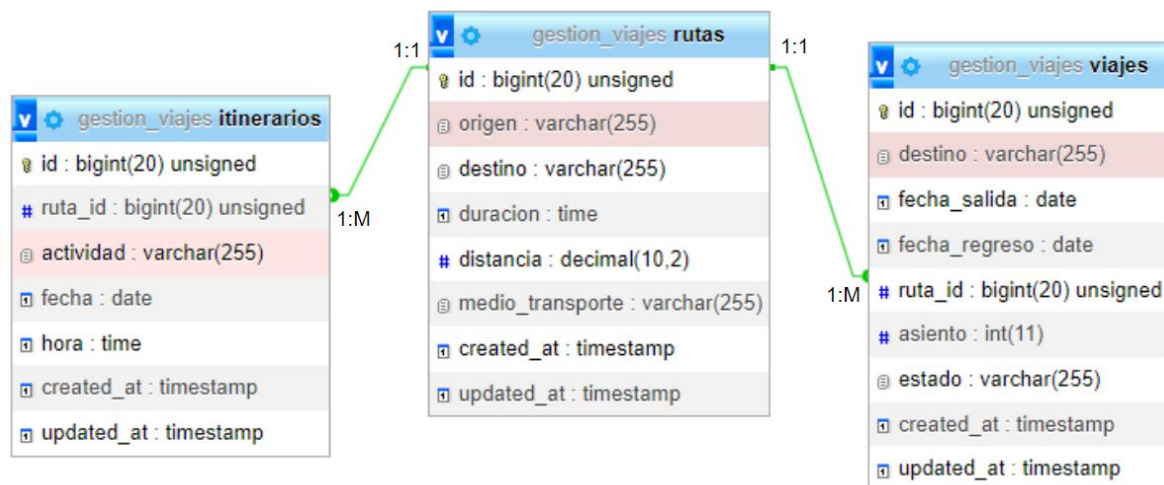
<https://docs.aws.amazon.com/es-es/AmazonCloudWatch/latest/monitoring/ContainerInsights-Prometheus-viewmetrics.html>

WebGrafía:

<https://docs.netapp.com/es-es/storagegrid-enable/tools-apps-guides/federate-prometheus.html>

ANEXOS:

1.-Diccionario de datos:



Cuadro de Entidades o Tablas de la Base de Datos

Tabla	Nombre de la Tabla	Descripción de la tabla	Nomenclatura
1	Tb_viajes	viajes	Viaje_id
2	Tb_rutas	rutas	ruta_id

3	Tb_Itinerarios	Itinerarios	Itinerario_id
---	----------------	-------------	---------------

[DICCIONARIO DE DATOS]

09 de Diciembre de 2023

7. TABLA DE ATRIBUTOS DE LAS ENTIDADES DE LA BASE DE DATOS

Tabla	Llave	Atributo	Tipo de Dato	nullable	Descripcion
tb_viajes	PK	id_viajes	Bigint(20)	NOT NULL	Id de la tabla viajes
	FK	Ruta_id	Bigint(20)	NULL	Id de la tabla ruta
		destino	varchar(255)	NULL	creacion de destino
		Fecha_regreso	date()	NULL	Fecha del Ingreso
		Fecha_Salida	date()	NULL	Fecha de la salida
		asiento	INT(11)	NULL	Numero de asiento
		estado	VARCHAR(255)	NULL	Estado de viaje
		Created_at	TIMESTAMP	NULL	Creación de at
		Updated_at	TIMESTAMP	NULL	Actualizacion de at
tb_rutas	PK	id_rutas	Bigint(20)	NOT NULL	Id de la tabla rutas
		duracion	time()	NULL	Cantidad de tiempo
		distancia	decimal(10,2)	NULL	Cantidad en decimal












		destino	VARCHAR(255)	NULL	Destino de ruta
		origen	VARCHAR(255)	NULL	Origen de ruta
		medio_transporte	VARCHAR(255)	NULL	Medio de ruta
		create_at	TIMESTAMP	NULL	Creacion de at. ruta
		updated_at	TIMESTAMP	NULL	Actualizacion de at. ruta
tb_itinerarios	PK	id_itinerarios	Bigint(20)	NOT NULL	Id de la tabla de itinerarios
		actividad	varchar(255)	NULL	Nombre de la actividad
		fecha	date()	NULL	Fecha de creacion
		hora	time()	NULL	Hora de entrada
	FK	Ruta_id	Bigint(10)	NULL	Id de la tabla ruta
		created_at	TIMESTAMP	NULL	Creacion de at it.
		Updated_at	TIMESTAMP	NULL	Actualizacion de at.

gestion_viajes	320,0 KiB
failed_jobs	32,0 KiB
itinerarios	80,0 KiB
migrations	16,0 KiB
password_reset_tokens	16,0 KiB
personal_access_tokens	48,0 KiB
rutas	16,0 KiB
users	32,0 KiB
viajes	80,0 KiB

#	Nombre	Tipo de datos	Longitud/Co...
1	id	BIGINT	20
2	destino	VARCHAR	255
3	fecha_salida	DATE	
4	fecha_regreso	DATE	
5	ruta_id	BIGINT	20
6	asiento	INT	11
7	estado	VARCHAR	255
8	created_at	TIMESTAMP	
9	updated_at	TIMESTAMP	

gestion_viajes	320,0 KiB
failed_jobs	32,0 KiB
itinerarios	80,0 KiB
migrations	16,0 KiB
password_reset_tokens	16,0 KiB
personal_access_tokens	48,0 KiB
rutas	16,0 KiB
users	32,0 KiB
viajes	80,0 KiB

#	Nombre	Tipo de datos	Longitud/Co...
1	id	BIGINT	20
2	origen	VARCHAR	255
3	destino	VARCHAR	255
4	duracion	TIME	
5	distancia	DECIMAL	10,2
6	medio_transp...	VARCHAR	255
7	created_at	TIMESTAMP	
8	updated_at	TIMESTAMP	

▼  gestion_viajes	320,0 KiB	#	Nombre	Tipo de datos	Longitud/Co...
 failed_jobs	32,0 KiB	 1	id	BIGINT	20
 itinerarios	80,0 KiB	 2	ruta_id	BIGINT	20
 migrations	16,0 KiB	3	actividad	VARCHAR	255
 password_reset_tokens	16,0 KiB	4	fecha	DATE	
 personal_access_tokens	48,0 KiB	5	hora	TIME	
 rutas	16,0 KiB	6	created_at	TIMESTAMP	
 users	32,0 KiB	7	updated_at	TIMESTAMP	
 viajes	80,0 KiB				

2.-Estandar de Programación:

PHP:

PHP (Hypertext Preprocessor):

PHP, acrónimo recursivo de "Hypertext Preprocessor", es un lenguaje de programación de código abierto y ampliamente utilizado, especialmente diseñado para el desarrollo de aplicaciones web dinámicas. Originalmente creado por Rasmus Lerdorf en 1994, PHP ha experimentado varias actualizaciones y mejoras a lo largo de los años y se ha convertido en uno de los lenguajes más populares para el desarrollo del lado del servidor en la web.

.-Convenciones de Nombres:

- Utiliza camelCase para variables y funciones.
- Usa PascalCase para nombres de clases e interfaces.
- Prefija nombres de propiedades privadas con un guion bajo (e.g., **\$ _miPropiedadPrivada**).

Indentación y Espaciado:

- Indenta con cuatro espacios.
- Utiliza espacios alrededor de los operadores (excepto en operadores unarios).

Comentarios:

- Usa comentarios explicativos y concisos.
- Documenta las funciones y métodos usando PHPDoc.

Manejo de Errores:

- Utiliza excepciones para manejar errores.
- Implementa bloques try-catch donde sea necesario.

Seguridad:

- Evita el uso de funciones obsoletas o inseguras.
- Sanitiza y valida siempre las entradas del usuario.

Laravel:

Laravel es un moderno y poderoso marco de desarrollo de código abierto para aplicaciones web basadas en PHP. Creado por Taylor Otwell y lanzado por primera vez en 2011, Laravel ha ganado rápidamente popularidad en la comunidad de desarrollo web debido a su elegante sintaxis, su conjunto completo de características y su enfoque en la elegancia y la simplicidad del código.

Eloquent:

- Sigue las convenciones de nomenclatura de Eloquent.
- Utiliza relaciones Eloquent para simplificar las consultas de la base de datos.

Rutas:

- Utiliza rutas nombradas para referenciarlas en lugar de rutas codificadas.
- Agrupa rutas relacionadas utilizando el método **group**.

Controladores:

- Divide lógica relacionada en controladores.
- Limita la cantidad de métodos en un controlador y sigue el principio de responsabilidad única.

Middleware:

- Utiliza middleware para la autorización, autenticación y otras tareas de filtrado.
- Define middleware personalizado según sea necesario.

Vistas:

- Divide las vistas en secciones (layouts, partials) para una mejor modularidad.
- Utiliza el sistema de plantillas Blade de manera efectiva.

Migraciones y Semillas:

- Utiliza migraciones para gestionar la estructura de la base de datos.
- Utiliza semillas para poblar la base de datos con datos iniciales.

Testing:

- Escribe pruebas unitarias y de integración.
- Utiliza las herramientas de prueba proporcionadas por Laravel, como PHPUnit.

Configuración:

- Almacena configuraciones en el archivo **.env**.
- Divide configuraciones complejas en archivos dedicados en el directorio **config**.

Seguridad:

- Utiliza el middleware proporcionado por Laravel para proteger contra ataques CSRF y XSS.
- Implementa políticas de autorización para controlar el acceso a recursos.

Documentación:

- Documenta el código de manera clara y concisa.
- Usa anotaciones de PHPDoc en clases y métodos.

Capturas del Servicio y Trabajo Desplegado Implementado en la nube:

Dominio: <http://salinas.sytes.net>




















Librerías:

```
composer require promphp/prometheus_client_php
composer require predis/predis
composer require rakutentech/laravel-request-docs --dev
```

Despliegue del Servicio de la API de Gestion de Viajes

Ejecucion de los contenedores: docker-compose up

```
PS C:\xampp\htdocs\Base de Datos II\BD2_API_viajesPHP_U2\Despliegue_YAML_U1_U2> docker-compose up -d
[+] Building 0.0s (0/0)
[+] Running 7/7
✓ Network despliegue_yaml_u1_u2_default Created
✓ Container grafana-metricas Started
✓ Container viajes-db Started
✓ Container redis-metricas Started
✓ Container prometheus-metricas Started
✓ Container viajes-api Started
✓ Container viajes-adminer Started
PS C:\xampp\htdocs\Base de Datos II\BD2_API_viajesPHP_U2\Despliegue_YAML_U1_U2> 
```

<input type="checkbox"/>		despliegue_yaml.u1.u2		Running (6/6)	0.93%
<input type="checkbox"/>		viajes-adminer 63a27bdac75a 	maynerac/adminer-viajes-u1-2.0	Running	0% 8484.8080 
<input type="checkbox"/>		prometheus-metricas f174ccff44c9 	maynerac/prometheus-metricas-u2-2.0	Running	0.03% 9090.9090 
<input type="checkbox"/>		viajes-api cb4dee156a88 	maynerac/api-viajes-u1-2.0	Running	0.03% 8000.8000 
<input type="checkbox"/>		redis-metricas b10a038e942c 	maynerac/redis-metricas-u2-2.0	Running	0.16% 6379.6379 
<input type="checkbox"/>		viajes-db cd6a35689198 	maynerac/mysql-viajes-u1-2.0	Running	0.65% 3306.3306 
<input type="checkbox"/>		grafana-metricas 66cd76897bdc 	maynerac/grafana-metricas-u2-2.0	Running	0.06% 3000.3000 

Uso del ORM Eloquent de Laravel

Ejecucion de migraciones: docker exec viajes-api php artisan migrate

```

Administrador: Windows PowerShell

PS C:\> docker exec viajes-api php artisan migrate

INFO  Preparing database.

Creating migration table ..... 65ms DONE

INFO  Running migrations.

2014_10_12_000000_create_users_table ..... 110ms DONE
2014_10_12_100000_create_password_reset_tokens_table ..... 111ms DONE
2019_08_19_000000_create_failed_jobs_table ..... 33ms DONE
2019_12_14_000001_create_personal_access_tokens_table ..... 48ms DONE
2023_10_07_052022_create_rutas_table ..... 20ms DONE
2023_10_07_052146_create_viajes_table ..... 60ms DONE
2023_10_07_052317_create_itinerarios_table ..... 63ms DONE

```

Ejecucion de seeders: docker exec viajes-api php artisan db:seed

```

Administrador: Windows PowerShell

PS C:\> docker exec viajes-api php artisan db:seed
>>

INFO  Seeding database.

Database\Seeders\UsersTableSeeder ..... RUNNING
Database\Seeders\UsersTableSeeder ..... 140.91 ms DONE

Database\Seeders\RutasTableSeeder ..... RUNNING
Database\Seeders\RutasTableSeeder ..... 99.94 ms DONE

Database\Seeders\ViajesTableSeeder ..... RUNNING
Database\Seeders\ViajesTableSeeder ..... 1,144.83 ms DONE

Database\Seeders\ItinerariosTableSeeder ..... RUNNING
Database\Seeders\ItinerariosTableSeeder ..... 1,005.89 ms DONE

PS C:\>

```

Adminer:

No es seguro | salinas.sytes.net:8484

Idioma:

Español

Adminer 4.8.1

Login

Motor de base de datos

MySQL

Servidor

viajes-db

Usuario

admin

Contraseña

.....

Base de datos

gestion_viajes

Login

☐ Guardar contraseña

No es seguro | salinas.sytes.net:8484/?server=viajes-db&username=admin&db=gestion_viajes&select=viajes

Idioma:

Español

Adminer 4.8.1

Mostrar: viajes

DB:

gestion_viajes

Comando SQL

Importar

Exportar

Crear tabla

registros failed_jobs

registros itinerarios

registros migrations

registros password_reset_tokens

registros personal_access_tokens

registros rutas

registros users

registros viajes

MySQL > viajes-db > gestion_viajes > Mostrar: viajes

Visualizar contenido

Mostrar estructura

Modificar tabla

Nuevo Registro

Mostrar

Condición

Ordenar

Limite

Longitud de texto

Acción

Mostrar

50

100

Mostrar

SELECT * FROM `viajes` LIMIT 50 (0,001 s) Modificar

	Modificar	Id	destino	fecha_salida	fecha_regreso	ruta_id	asiento	estado	created_at	updated_at
<input type="checkbox"/>	Modificar	1	North Lemuefurt	1980-09-29	1976-01-06	1	75	En curso	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	2	Jedburgh	1995-12-12	2005-09-02	1	1	En curso	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	3	Hellerchester	2002-04-23	1971-07-02	1	78	Completado	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	4	New Cordellamouth	1987-04-05	1985-04-21	1	8	Completado	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	5	Gloverview	1988-02-15	2002-05-08	1	37	En curso	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	6	Mayertmouth	2001-07-28	1973-01-20	1	82	En curso	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	7	Port Freddie	1996-06-18	1985-02-05	1	38	Completado	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	8	Keeblertown	1998-04-20	2016-03-25	1	80	Completado	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	9	New Kaya	1991-01-04	1993-11-24	1	92	Completado	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	10	Brannonborough	2016-07-03	1999-01-03	1	9	Completado	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	11	Port Heidi	2023-01-29	1971-06-11	2	15	En curso	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	12	Boyerland	1974-12-17	2004-04-09	2	94	Completado	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	13	Jacobimouth	2016-04-28	1988-07-27	2	79	Programado	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	14	Americastad	2015-06-24	1999-07-28	2	21	Completado	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	15	Terranceside	2022-12-09	1974-09-05	2	36	Programado	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	16	Bobbyshire	2006-09-10	1987-07-06	2	91	Programado	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	17	South Ladarius	1975-06-18	2001-10-09	2	58	Completado	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	18	South Nicolettefurt	2017-11-17	1992-02-25	2	98	Completado	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	19	Lowebury	1993-09-30	2009-01-28	2	1	En curso	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	20	Jennietown	2002-01-29	2003-08-28	2	9	Programado	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	21	Antwanfurt	1999-12-01	1992-05-05	3	43	Programado	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	22	Geovannyview	1972-07-24	2006-07-27	3	31	En curso	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	23	Lake Theodore	2018-06-10	1979-07-23	3	33	Programado	2023-12-07 03:41:59	2023-12-07 03:41:59
<input type="checkbox"/>	Modificar	24	Port Orphastad	2017-03-24	1998-02-10	3	90	Completado	2023-12-07 03:41:59	2023-12-07 03:41:59

Página

1 2 3 4 5 6

Resultado completo

☐ 300 registros

Modificar

Clonar

Eliminar

Exportar (300)

Request-docs

No es seguro | salinas.sytes.net:8000/request-docs

LARAVEL REQUEST DOCS

Filter APIs

OPENAPI 3.0

FEATURE REQ

API LIST

Total 25

GET

api/rutas

HEAD

api/rutas

GET

api/rutas/{ruta}

HEAD

api/rutas/{ruta}

POST

api/rutas

PUT

api/rutas/{ruta}

DELETE

api/rutas/{ruta}

GET

api/viajes

HEAD

api/viajes

GET

api/viajes/{viaje}

HEAD

api/viajes/{viaje}

POST

api/viajes

PUT

api/viajes/{viaje}

DELETE

api/viajes/{viaje}

GET

api/itinerarios

HEAD

api/itinerarios

GET

api/itinerarios/{itinerario}

HEAD

api/itinerarios/{itinerario}

POST

api/itinerarios

PUT

api/itinerarios/{itinerario}

DELETE

api/itinerarios/{itinerario}

GET

/

HEAD

/

GET

/metrics

HEAD

/metrics

api/rutas

REQUEST SCHEMA

web::Location::from

QUERY PARAMETERS

No Rules Defined

Info

Send

Response

SQL

Logs

Events

Method

GET

Controller

RouteController

Function

index

Middlewares

API

Status Codes

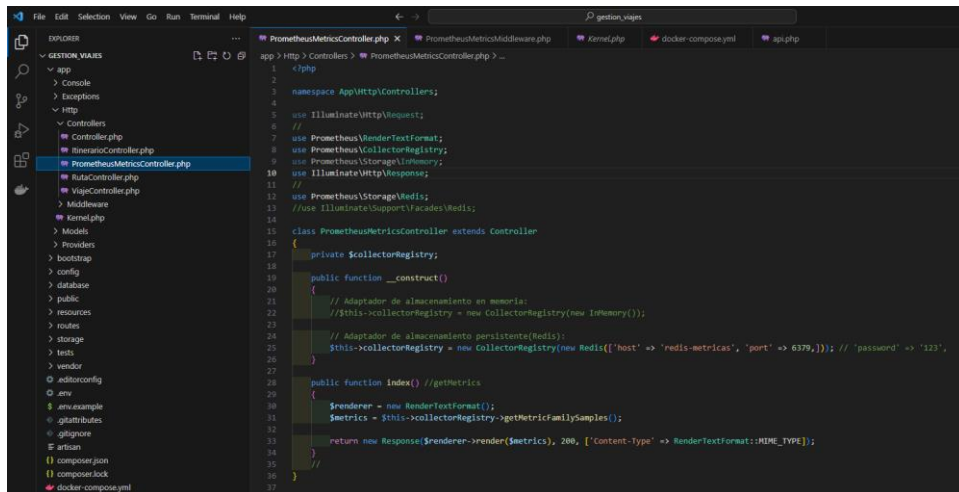
Show Response codes for this request >

curl

Show curl command >

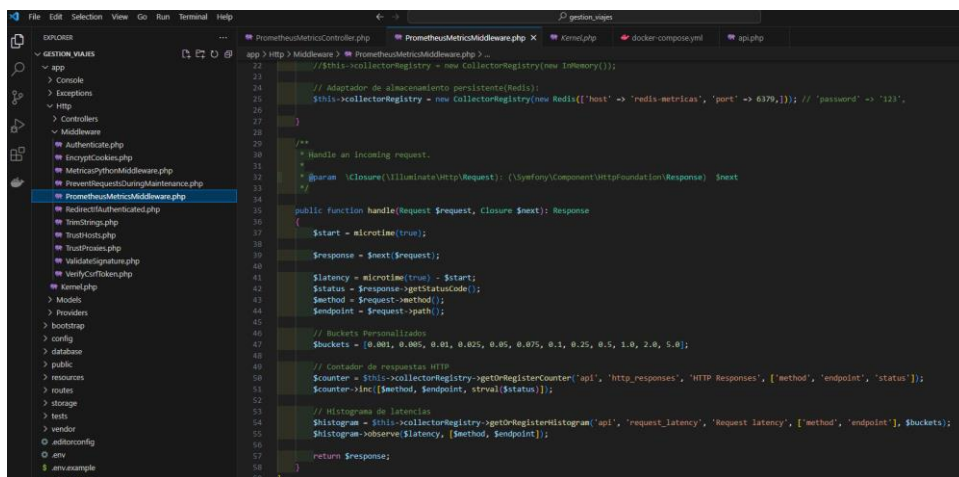
Activar Windows

Controller:



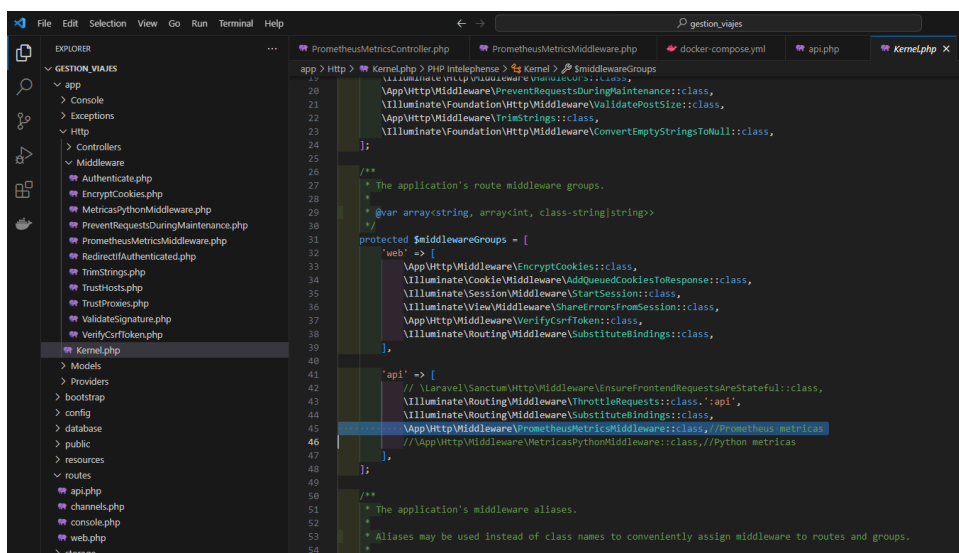
```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 //
7 use Prometheus\RenderTextFormat;
8 use Prometheus\CollectorRegistry;
9 use Prometheus\Storage\Redis;
10 use Illuminate\Http\Response;
11 //
12 use Prometheus\Storage\Redis;
13 //use Illuminate\Support\Facades\Redis;
14
15 class PrometheusMetricsController extends Controller
16 {
17     private $collectorRegistry;
18
19     public function __construct()
20     {
21         // Adaptador de almacenamiento en memoria:
22         // $this->collectorRegistry = new CollectorRegistry(new InMemory());
23
24         // Adaptador de almacenamiento persistente (Redis):
25         $this->collectorRegistry = new CollectorRegistry(new Redis(['host' => 'redis-metrics', 'port' => 6379])); // 'password' => '123';
26     }
27
28     public function index() //getMetrics
29     {
30         $renderer = new RenderTextFormat();
31         $metrics = $this->collectorRegistry->getMetricFamilySamples();
32
33         return new Response($renderer->render($metrics), 200, ['Content-Type' => RenderTextFormat::MIME_TYPE]);
34     }
35
36 }
```

Middleware:



```
1 // PrometheusMetricsMiddleware.php
2
3 // $this->collectorRegistry = new CollectorRegistry(new InMemory());
4
5 // Adaptador de almacenamiento persistente (Redis):
6 $this->collectorRegistry = new CollectorRegistry(new Redis(['host' => 'redis-metrics', 'port' => 6379])); // 'password' => '123';
7
8
9
10 /**
11  * Handle an incoming request.
12  *
13  * @param \Illuminate\Http\Request $request
14  * @param \Closure($request): \Illuminate\Http\Response $next
15  *
16  * @return \Illuminate\Http\Response
17  */
18 public function handle($request, Closure $next): Response
19 {
20     $start = microtime(true);
21
22     $response = $next($request);
23
24     $latency = microtime(true) - $start;
25     $status = $response->getStatusCode();
26     $method = $request->method();
27     $endpoint = $request->path();
28
29     // Buckets Personalizados
30     $buckets = [0.001, 0.005, 0.01, 0.025, 0.05, 0.075, 0.1, 0.25, 0.5, 1.0, 2.0, 5.0];
31
32     // Contador de respuestas HTTP
33     $counter = $this->collectorRegistry->getRegisterCounter('api', 'http_responses', 'HTTP Responses', ['method', 'endpoint', 'status']);
34     $counter->inc($method, $endpoint, strval($status));
35
36     // Histograma de latencias
37     $histogram = $this->collectorRegistry->getRegisterHistogram('api', 'request_latency', 'Request latency', ['method', 'endpoint'], $buckets);
38     $histogram->observe($latency, [$method, $endpoint]);
39
40     return $response;
41 }
```

Kernel:



```
1 // Kernel.php
2
3 // $this->collectorRegistry = new CollectorRegistry(new InMemory());
4
5 // Adaptador de almacenamiento persistente (Redis):
6 $this->collectorRegistry = new CollectorRegistry(new Redis(['host' => 'redis-metrics', 'port' => 6379])); // 'password' => '123';
7
8
9
10 /**
11  * The application's route middleware groups.
12  *
13  * @var array<string, array<int, class-string>>
14  */
15 protected $middlewareGroups = [
16     'web' => [
17         \App\Http\Middleware\EncryptCookies::class,
18         \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,
19         \Illuminate\Session\Middleware\StartSession::class,
20         \Illuminate\View\Middleware\ShareErrorsFromSession::class,
21         \App\Http\Middleware\VerifyCsrfToken::class,
22         \Illuminate\Routing\Middleware\SubstituteBindings::class,
23     ],
24
25     'api' => [
26         // \Laravel\Sanctum\Http\Middleware\EnsureFrontendRequestsAreStateful::class,
27         \Illuminate\Routing\Middleware\ThrottleRequests::class,
28         \Illuminate\Routing\Middleware\SubstituteBindings::class,
29         \App\Http\Middleware\PrometheusMetricsMiddleware::class,
30         // \App\Http\Middleware\MetricsPythonMiddleware::class,
31     ],
32 ];
33
34 /**
35  * The application's middleware aliases.
36  *
37  * Aliases may be used instead of class names to conveniently assign middleware to routes and groups.
38  */
39 protected $middlewareAliases = [
40     'throttle' => \Illuminate\Routing\Middleware\ThrottleRequests::class,
41     'bindings' => \Illuminate\Routing\Middleware\SubstituteBindings::class,
42     'metrics' => \App\Http\Middleware\PrometheusMetricsMiddleware::class,
43     'python-metrics' => \App\Http\Middleware\MetricsPythonMiddleware::class,
44 ];
```

Routes:

```

93     });
94     */
95
96
97     //-----Rutas-----//
98
99     Route::get('rutas', [RutaController::class, 'index']);
100    Route::get('rutas/{ruta}', [RutaController::class, 'show']);
101    Route::post('rutas', [RutaController::class, 'store']);
102    Route::put('rutas/{ruta}', [RutaController::class, 'update']);
103    Route::delete('rutas/{ruta}', [RutaController::class, 'delete']);
104
105    //-----Viajes-----//
106
107    Route::get('viajes', [ViajeController::class, 'index']);
108    Route::get('viajes/{viaje}', [ViajeController::class, 'show']);
109    Route::post('viajes', [ViajeController::class, 'store']);
110    Route::put('viajes/{viaje}', [ViajeController::class, 'update']);
111    Route::delete('viajes/{viaje}', [ViajeController::class, 'delete']);
112
113    //-----Itinerarios-----//
114
115    Route::get('itinerarios', [ItinerarioController::class, 'index']);
116    Route::get('itinerarios/{itinerario}', [ItinerarioController::class, 'show']);
117    Route::post('itinerarios', [ItinerarioController::class, 'store']);
118    Route::put('itinerarios/{itinerario}', [ItinerarioController::class, 'update']);
119    Route::delete('itinerarios/{itinerario}', [ItinerarioController::class, 'delete']);
120
121    //-----//

```

Endpoint “/metrics”

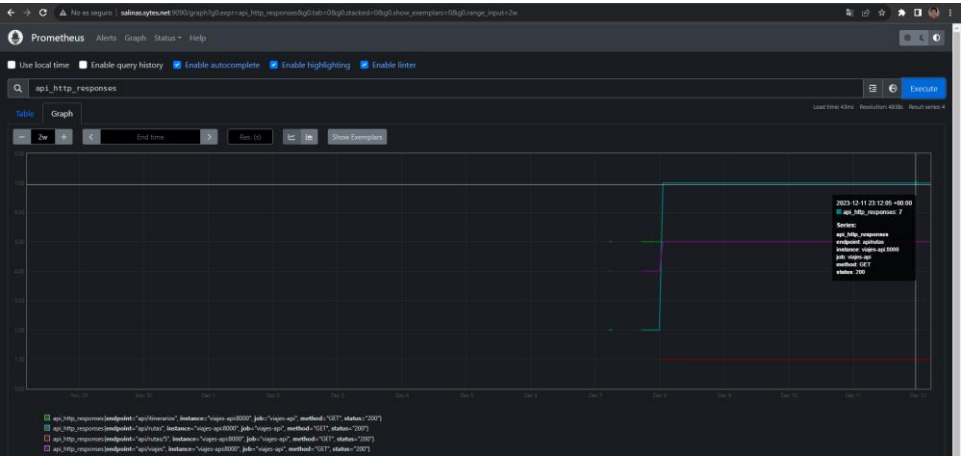
← → C No es seguro | salinas.sytes.net:8000/metrics

```

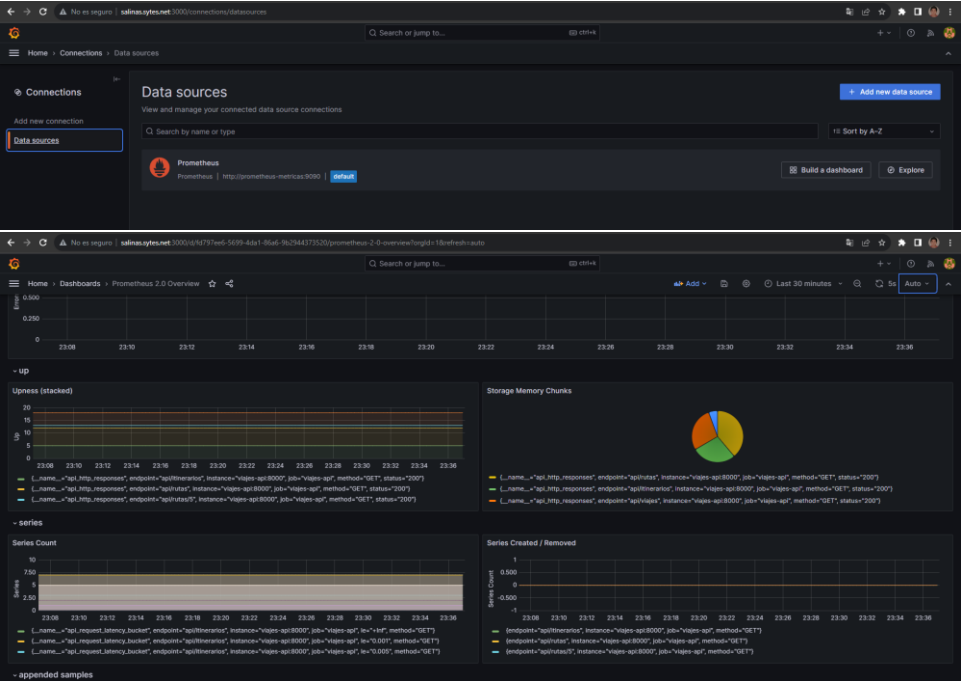
# HELP api_http_responses HTTP Responses
# TYPE api_http_responses counter
api_http_responses{method="GET",endpoint="api/itinerarios",status="200"} 5
api_http_responses{method="GET",endpoint="api/rutas/5",status="200"} 1
api_http_responses{method="GET",endpoint="api/rutas",status="200"} 7
api_http_responses{method="GET",endpoint="api/viajes",status="200"} 5
# HELP api_request_latency Request Latency
# TYPE api_request_latency histogram
api_request_latency_bucket{method="GET",endpoint="api/itinerarios",le="0.001"} 0
api_request_latency_bucket{method="GET",endpoint="api/itinerarios",le="0.005"} 0
api_request_latency_bucket{method="GET",endpoint="api/itinerarios",le="0.01"} 0
api_request_latency_bucket{method="GET",endpoint="api/itinerarios",le="0.025"} 0
api_request_latency_bucket{method="GET",endpoint="api/itinerarios",le="0.05"} 0
api_request_latency_bucket{method="GET",endpoint="api/itinerarios",le="0.075"} 0
api_request_latency_bucket{method="GET",endpoint="api/itinerarios",le="0.1"} 1
api_request_latency_bucket{method="GET",endpoint="api/itinerarios",le="0.25"} 5
api_request_latency_bucket{method="GET",endpoint="api/itinerarios",le="0.5"} 5
api_request_latency_bucket{method="GET",endpoint="api/itinerarios",le="1"} 5
api_request_latency_bucket{method="GET",endpoint="api/itinerarios",le="2"} 5
api_request_latency_bucket{method="GET",endpoint="api/itinerarios",le="5"} 5
api_request_latency_bucket{method="GET",endpoint="api/itinerarios",le="+Inf"} 5
api_request_latency_count{method="GET",endpoint="api/itinerarios"} 5
api_request_latency_sum{method="GET",endpoint="api/itinerarios"} 0.54398488984125
api_request_latency_bucket{method="GET",endpoint="api/rutas",le="0.001"} 0
api_request_latency_bucket{method="GET",endpoint="api/rutas",le="0.005"} 0
api_request_latency_bucket{method="GET",endpoint="api/rutas",le="0.01"} 0
api_request_latency_bucket{method="GET",endpoint="api/rutas",le="0.025"} 0
api_request_latency_bucket{method="GET",endpoint="api/rutas",le="0.05"} 3
api_request_latency_bucket{method="GET",endpoint="api/rutas",le="0.075"} 7
api_request_latency_bucket{method="GET",endpoint="api/rutas",le="0.1"} 7
api_request_latency_bucket{method="GET",endpoint="api/rutas",le="0.25"} 7
api_request_latency_bucket{method="GET",endpoint="api/rutas",le="0.5"} 7
api_request_latency_bucket{method="GET",endpoint="api/rutas",le="1"} 7
api_request_latency_bucket{method="GET",endpoint="api/rutas",le="2"} 7
api_request_latency_bucket{method="GET",endpoint="api/rutas",le="5"} 7
api_request_latency_bucket{method="GET",endpoint="api/rutas",le="+Inf"} 7
api_request_latency_count{method="GET",endpoint="api/rutas"} 7
api_request_latency_sum{method="GET",endpoint="api/rutas"} 0.369286775588988
api_request_latency_bucket{method="GET",endpoint="api/rutas/5",le="0.001"} 0
api_request_latency_bucket{method="GET",endpoint="api/rutas/5",le="0.005"} 0
api_request_latency_bucket{method="GET",endpoint="api/rutas/5",le="0.01"} 0
api_request_latency_bucket{method="GET",endpoint="api/rutas/5",le="0.025"} 1
api_request_latency_bucket{method="GET",endpoint="api/rutas/5",le="0.05"} 1
api_request_latency_bucket{method="GET",endpoint="api/rutas/5",le="0.075"} 1
api_request_latency_bucket{method="GET",endpoint="api/rutas/5",le="0.1"} 1
api_request_latency_bucket{method="GET",endpoint="api/rutas/5",le="0.25"} 1
api_request_latency_bucket{method="GET",endpoint="api/rutas/5",le="0.5"} 1
api_request_latency_bucket{method="GET",endpoint="api/rutas/5",le="1"} 1
api_request_latency_bucket{method="GET",endpoint="api/rutas/5",le="2"} 1
api_request_latency_bucket{method="GET",endpoint="api/rutas/5",le="5"} 1
api_request_latency_bucket{method="GET",endpoint="api/rutas/5",le="+Inf"} 1
api_request_latency_count{method="GET",endpoint="api/rutas/5"} 1
api_request_latency_sum{method="GET",endpoint="api/rutas/5"} 0.01043701171875
api_request_latency_bucket{method="GET",endpoint="api/viajes",le="0.001"} 0
api_request_latency_bucket{method="GET",endpoint="api/viajes",le="0.005"} 0
api_request_latency_bucket{method="GET",endpoint="api/viajes",le="0.01"} 0
api_request_latency_bucket{method="GET",endpoint="api/viajes",le="0.025"} 0
api_request_latency_bucket{method="GET",endpoint="api/viajes",le="0.05"} 0
api_request_latency_bucket{method="GET",endpoint="api/viajes",le="0.075"} 0
api_request_latency_bucket{method="GET",endpoint="api/viajes",le="0.1"} 2
api_request_latency_bucket{method="GET",endpoint="api/viajes",le="0.25"} 5
api_request_latency_bucket{method="GET",endpoint="api/viajes",le="0.5"} 5

```

Prometheus:







Grafana:



Capturas del Servicio de metricas para las APIs de Viajes, Clientes, Pasajes y Usuarios

Contenedores:

<input type="checkbox"/>	Name	Image	Status	CPU (%)	Port(s)
<input type="checkbox"/>	>  apiredisgest		Running (2/2)	0.26%	
<input type="checkbox"/>	>  pasajesapi		Running (4/4)	0.64%	
<input type="checkbox"/>	>  gestiondeclientes_u1		Running (2/2)	1.28%	
<input type="checkbox"/>	>  despliegue_yaml_u1_u2		Running (6/6)	1.34%	

Configuracion de Targets para Prometheus

```
! prometheus.yml x docker-compose.yml
! prometheus.yml
1  global:
2    scrape_interval: 5s
3
4  scrape_configs:
5    - job_name: "prometheus-metricas"
6      static_configs:
7        - targets: ["prometheus-metricas:9090"]
8
9    - job_name: "viajes-api"
10     static_configs:
11       - targets: ["viajes-api:8000"]
12
13    - job_name: "clientes-api"
14     static_configs:
15       - targets: ["host.docker.internal:5000"]
16
17    - job_name: "pasajes-api"
18     static_configs:
19       - targets: ["host.docker.internal:8001"]
20
21    - job_name: "usuarios-api"
22     static_configs:
23       - targets: ["host.docker.internal:5001"]
```

localhost:9090/targets?search=#pool-clientes-api

Prometheus Alerts Graph Status Help

Targets

All scrape pools * All 100 healthy Collapse All

Filter by endpoint or labels

clientes-api (1/1 up) [Details](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://host.docker.internal:5000/metrics	UP	instance="host.docker.internal:5000" job="clientes-api"	14.673s ago	5.366ms	

pasajes-api (1/1 up) [Details](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://host.docker.internal:8001/metrics	UP	instance="host.docker.internal:8001" job="pasajes-api"	16.115s ago	78.742ms	

prometheus-metricas (1/1 up) [Details](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://prometheus-metricas:9090/metrics	UP	instance="prometheus-metricas:9090" job="prometheus-metricas"	14.167s ago	5.761ms	

usuarios-api (1/1 up) [Details](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://host.docker.internal:5001/metrics	UP	instance="host.docker.internal:5001" job="usuarios-api"	16.147s ago	5.927ms	

viajes-api (1/1 up) [Details](#)

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://viajes-api:8000/metrics	UP	instance="viajes-api:8000" job="viajes-api"	17.796s ago	73.085ms	

Endpoints “/metrics” en las APIs de Viajes, Clientes, Pasajes y Usuarios

```
← → ↻ ⓘ localhost:8000/metrics

# HELP api_http_responses HTTP Responses
# TYPE api_http_responses counter
api_http_responses(method="GET",endpoint="api/itinerarios",status="200") 4
api_http_responses(method="GET",endpoint="api/rutas",status="200") 7
api_http_responses(method="GET",endpoint="api/viajes",status="200") 5
# HELP api_request_latency Request latency
# TYPE api_request_latency histogram
api_request_latency_bucket(method="GET",endpoint="api/itinerarios",le="0.001") 0
api_request_latency_bucket(method="GET",endpoint="api/itinerarios",le="0.005") 0
api_request_latency_bucket(method="GET",endpoint="api/itinerarios",le="0.01") 0
api_request_latency_bucket(method="GET",endpoint="api/itinerarios",le="0.025") 0
api_request_latency_bucket(method="GET",endpoint="api/itinerarios",le="0.05") 4
api_request_latency_bucket(method="GET",endpoint="api/itinerarios",le="0.075") 4
api_request_latency_bucket(method="GET",endpoint="api/itinerarios",le="0.1") 4
api_request_latency_bucket(method="GET",endpoint="api/itinerarios",le="0.25") 4
api_request_latency_bucket(method="GET",endpoint="api/itinerarios",le="0.5") 4
api_request_latency_bucket(method="GET",endpoint="api/itinerarios",le="1") 4
api_request_latency_bucket(method="GET",endpoint="api/itinerarios",le="2") 4
api_request_latency_bucket(method="GET",endpoint="api/itinerarios",le="5") 4
api_request_latency_bucket(method="GET",endpoint="api/itinerarios",le="+Inf") 4
api_request_latency_count(method="GET",endpoint="api/itinerarios") 4
api_request_latency_sum(method="GET",endpoint="api/itinerarios") 0.159240007400512
api_request_latency_bucket(method="GET",endpoint="api/rutas",le="0.001") 0
api_request_latency_bucket(method="GET",endpoint="api/rutas",le="0.005") 0
api_request_latency_bucket(method="GET",endpoint="api/rutas",le="0.01") 0
api_request_latency_bucket(method="GET",endpoint="api/rutas",le="0.025") 7
api_request_latency_bucket(method="GET",endpoint="api/rutas",le="0.05") 7
api_request_latency_bucket(method="GET",endpoint="api/rutas",le="0.075") 7
api_request_latency_bucket(method="GET",endpoint="api/rutas",le="0.1") 7
api_request_latency_bucket(method="GET",endpoint="api/rutas",le="0.25") 7
api_request_latency_bucket(method="GET",endpoint="api/rutas",le="0.5") 7
api_request_latency_bucket(method="GET",endpoint="api/rutas",le="1") 7
api_request_latency_bucket(method="GET",endpoint="api/rutas",le="2") 7
api_request_latency_bucket(method="GET",endpoint="api/rutas",le="5") 7
api_request_latency_bucket(method="GET",endpoint="api/rutas",le="+Inf") 7
api_request_latency_count(method="GET",endpoint="api/rutas") 7
api_request_latency_sum(method="GET",endpoint="api/rutas") 0.117568254470824
api_request_latency_bucket(method="GET",endpoint="api/viajes",le="0.001") 0
api_request_latency_bucket(method="GET",endpoint="api/viajes",le="0.005") 0
api_request_latency_bucket(method="GET",endpoint="api/viajes",le="0.01") 0
api_request_latency_bucket(method="GET",endpoint="api/viajes",le="0.025") 0
api_request_latency_bucket(method="GET",endpoint="api/viajes",le="0.05") 5
api_request_latency_bucket(method="GET",endpoint="api/viajes",le="0.075") 5
api_request_latency_bucket(method="GET",endpoint="api/viajes",le="0.1") 5
api_request_latency_bucket(method="GET",endpoint="api/viajes",le="0.25") 5
api_request_latency_bucket(method="GET",endpoint="api/viajes",le="0.5") 5
api_request_latency_bucket(method="GET",endpoint="api/viajes",le="1") 5
api_request_latency_bucket(method="GET",endpoint="api/viajes",le="2") 5
api_request_latency_bucket(method="GET",endpoint="api/viajes",le="5") 5
api_request_latency_bucket(method="GET",endpoint="api/viajes",le="+Inf") 5
api_request_latency_count(method="GET",endpoint="api/viajes") 5
api_request_latency_sum(method="GET",endpoint="api/viajes") 0.211045026779174
# HELP php_info Information about the PHP environment.
# TYPE php_info gauge
php_info(version="8.2.13") 1
```

```
← → ↻ ⓘ localhost:9090/metrics

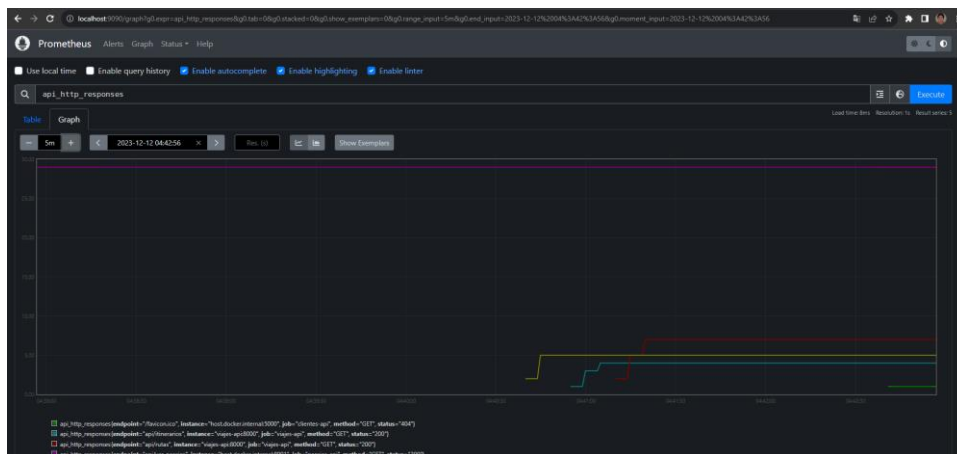
# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 2.2371e-05
go_gc_duration_seconds{quantile="0.25"} 7.785e-05
go_gc_duration_seconds{quantile="0.5"} 0.000181383
go_gc_duration_seconds{quantile="0.75"} 0.000202043
go_gc_duration_seconds{quantile="1"} 0.001233786
go_gc_duration_seconds_sum 0.000251837
go_gc_duration_seconds_count 22
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 41
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info(version="go1.21.4") 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 2.7298152e+07
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 2.421118e+08
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.40780e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 1.6948e+06
# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 1.04e+06
# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 2.729815e+07
# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
# TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 2.288448e+07
# HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use.
# TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 2.21074e+07
# HELP go_memstats_heap_objects Number of allocated objects.
# TYPE go_memstats_heap_objects gauge
go_memstats_heap_objects 154200
# HELP go_memstats_heap_released_bytes Number of heap bytes released to OS.
# TYPE go_memstats_heap_released_bytes gauge
go_memstats_heap_released_bytes 1.16736e+07
# HELP go_memstats_heap_sys_bytes Number of heap bytes obtained from system.
# TYPE go_memstats_heap_sys_bytes gauge
go_memstats_heap_sys_bytes 5.605572e+07
# HELP go_memstats_last_gc_time_seconds Number of seconds since 1970 of last garbage collection.
# TYPE go_memstats_last_gc_time_seconds gauge
go_memstats_last_gc_time_seconds 1.7823506358155083e+09
# HELP go_memstats_lookups_total Total number of pointer lookups.
# TYPE go_memstats_lookups_total counter
go_memstats_lookups_total 0
# HELP go_memstats_mallocs_total Total number of mallocs.
# TYPE go_memstats_mallocs_total counter
go_memstats_mallocs_total 1.4258e+06
# HELP go_memstats_mcache_inuse_bytes Number of bytes in use by mcache structures.
# TYPE go_memstats_mcache_inuse_bytes gauge
go_memstats_mcache_inuse_bytes 34400
# HELP go_memstats_mcache_sys_bytes Number of bytes used for mcache structures obtained from system.
# TYPE go_memstats_mcache_sys_bytes gauge
go_memstats_mcache_sys_bytes 22400
# HELP go_memstats_mspan_inuse_bytes Number of bytes in use by mspan structures.
# TYPE go_memstats_mspan_inuse_bytes gauge
go_memstats_mspan_inuse_bytes 41600
# HELP go_memstats_mspan_sys_bytes Number of bytes used for mspan structures obtained from system.
# TYPE go_memstats_mspan_sys_bytes gauge
go_memstats_mspan_sys_bytes 36176
# HELP go_memstats_next_gc_bytes Number of heap bytes when next garbage collection will take place.
# TYPE go_memstats_next_gc_bytes gauge
go_memstats_next_gc_bytes 3.470906e+07
# HELP go_memstats_other_sys_bytes Number of bytes used for other system allocations.
# TYPE go_memstats_other_sys_bytes gauge
```



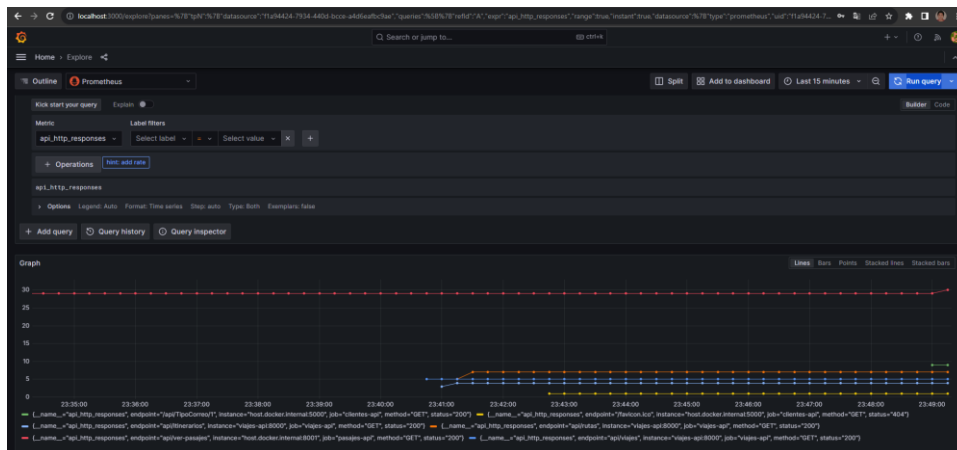
```
← → ↺ ⓘ localhost:5001/metrics

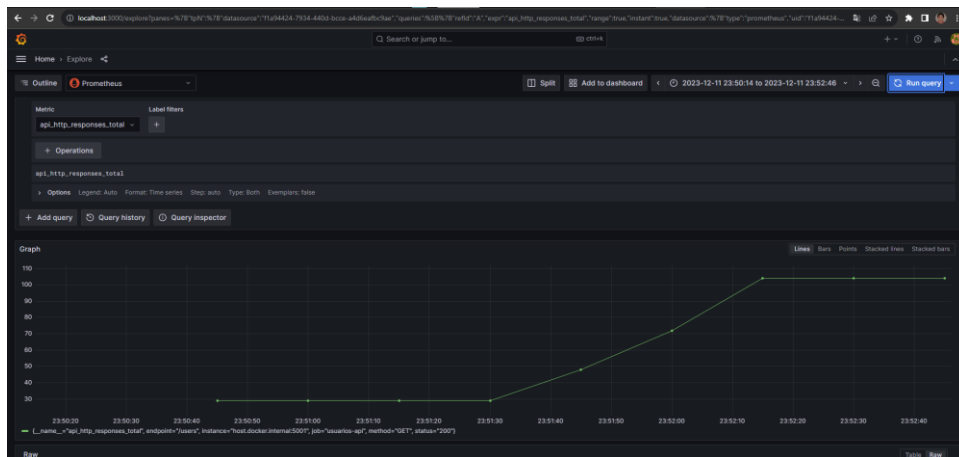
# HELP python_gc_objects_collected_total Objects collected during gc
# TYPE python_gc_objects_collected_total counter
python_gc_objects_collected_total{generation="0"} 17936.0
python_gc_objects_collected_total{generation="1"} 2226.0
python_gc_objects_collected_total{generation="2"} 35.0
# HELP python_gc_objects_uncollectable_total Uncollectable objects found during GC
# TYPE python_gc_objects_uncollectable_total counter
python_gc_objects_uncollectable_total{generation="0"} 0.0
python_gc_objects_uncollectable_total{generation="1"} 0.0
python_gc_objects_uncollectable_total{generation="2"} 0.0
# HELP python_gc_collections_total Number of times this generation was collected
# TYPE python_gc_collections_total counter
python_gc_collections_total{generation="0"} 182.0
python_gc_collections_total{generation="1"} 16.0
python_gc_collections_total{generation="2"} 1.0
# HELP python_info Python platform information
# TYPE python_info gauge
python_info{implementation="CPython",major="3",minor="8",patchlevel="18",version="3.8.18"} 1.0
# HELP process_virtual_memory_bytes Virtual memory size in bytes.
# TYPE process_virtual_memory_bytes gauge
process_virtual_memory_bytes 1.38843392e+08
# HELP process_resident_memory_bytes Resident memory size in bytes.
# TYPE process_resident_memory_bytes gauge
process_resident_memory_bytes 5.6598528e+07
# HELP process_start_time_seconds Start time of the process since unix epoch in seconds.
# TYPE process_start_time_seconds gauge
process_start_time_seconds 1.78235349131e+09
# HELP process_cpu_seconds_total Total user and system CPU time spent in seconds.
# TYPE process_cpu_seconds_total counter
process_cpu_seconds_total 2.1
# HELP process_open_fds Number of open file descriptors.
# TYPE process_open_fds gauge
process_open_fds 14.0
# HELP process_max_fds Maximum number of open file descriptors.
# TYPE process_max_fds gauge
process_max_fds 1.048576e+06
# HELP api_http_responses_total HTTP Responses
# TYPE api_http_responses_total counter
# HELP api_request_latency Request latency
# TYPE api_request_latency histogram
```

Métricas en Prometheus de todas las APIs:



Métricas Grafana de todas las APIs:





GitHub Actions Pipeline automatizar actualizacion de imagen en DockerHub

