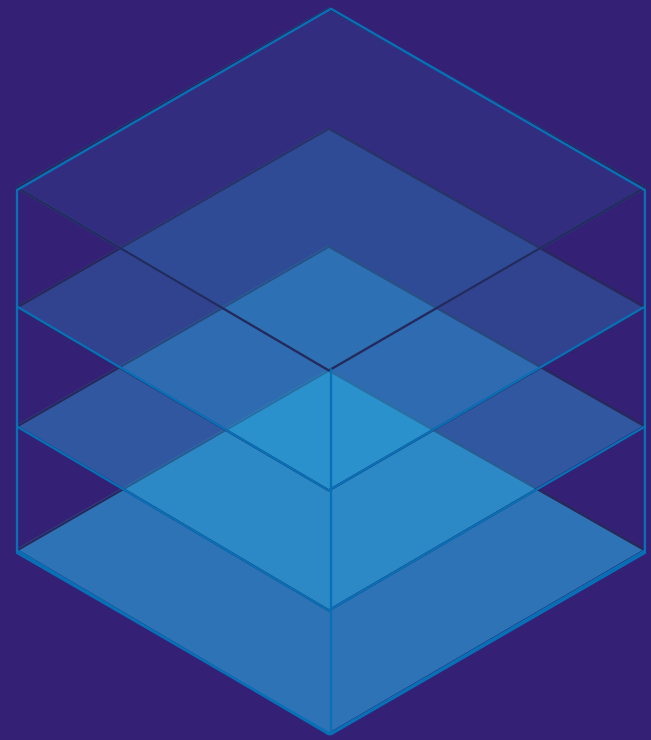


MISO

Maestría en Ingeniería de Software

Experimentación

SaludTech – Alpes
ByteBros



MISO

Maestría en Ingeniería de Software

Experimento 1 - Escalabilidad con Concurrencia (Load Balancer)

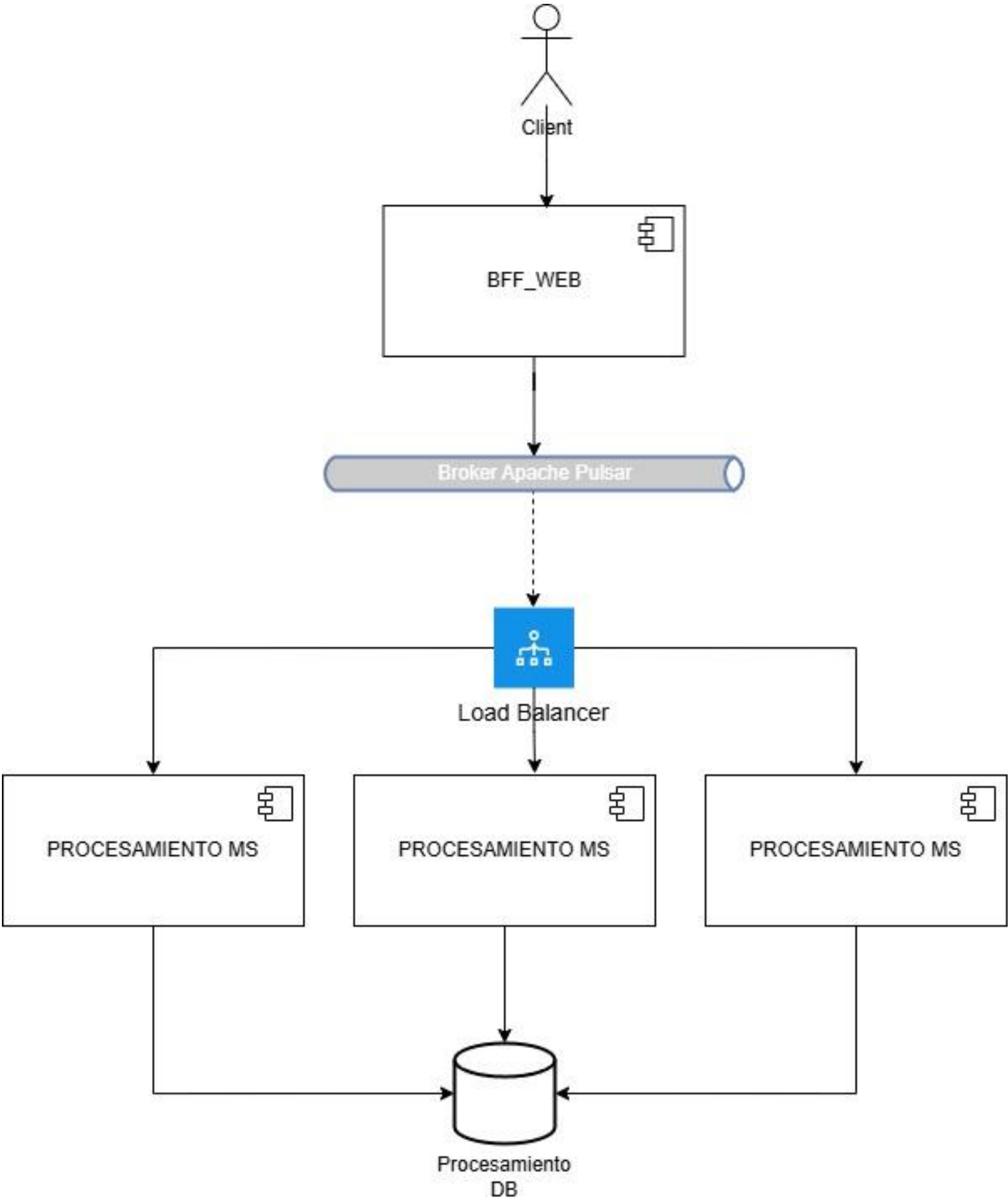
Atributo de calidad 1: Escalabilidad

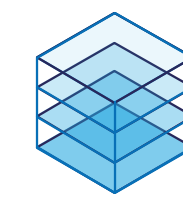
Escenario de calidad: Escalabilidad con Concurrencia (Load Balancer)			
Escenario #: 3	Evalúa la capacidad del sistema para manejar altos volúmenes de solicitudes concurrentes, distribuyéndolas de manera eficiente entre las instancias del sistema mediante un patrón de <i>Load Balancer</i> . El sistema debe ser capaz de procesar paquetes de imágenes, distribuyendo la carga entre diversas instancias de procesamiento, manteniendo la baja latencia, alta disponibilidad y tolerancia a fallos, especialmente al manejar eventos de procesamiento generados por el sistema.		
Fuente	El sistema que gestiona las solicitudes de procesamiento de las imágenes.		
Estímulo	El sistema recibe un paquete de imágenes que debe ser procesado. Este paquete es enviado como un evento para iniciar el procesamiento, disparando la asignación de tareas a las instancias disponibles.		
Ambiente	Operación normal con múltiples instancias activas, bajo una carga creciente de solicitudes. El sistema está preparado para gestionar picos de demanda y puede redirigir tareas entre instancias según la necesidad. En el caso de que una instancia falle, otra la sustituye sin interrumpir el flujo.		
Artefacto	El microservicio de procesamiento de imágenes		
Respuesta	El sistema distribuye las solicitudes de procesamiento de paquetes de imágenes entre varias instancias, asignando un paquete a cada instancia disponible. Si llegan más paquetes mientras una instancia está procesando, la carga es redistribuida entre las instancias restantes. El sistema debe asegurar que no haya cuellos de botella y que el procesamiento continúe sin interrupciones.		
Medida de la respuesta	El sistema debe procesar al menos el 99.9% de las solicitudes de imágenes sin superar un tiempo de respuesta de 2 segundos por paquete bajo carga máxima. El balanceador de carga debe ser capaz de distribuir la carga de manera eficiente sin que ninguna instancia supere el 75% de su capacidad.		
Decisiones Arquitecturales	Punto de sensibilidad	Tradeoff	Riesgo
Patrón de Load Balancer	Cuello de Botella en el Balanceo de Carga	Complejidad de Implementación vs. Escalabilidad	Falla en el Balanceador de Carga
Escalabilidad Dinámica	Sincronización de Instancias	Costos Operacionales vs. Redundancia	Sobrecarga de la Cola de Mensajes
Mensajería Asíncrona			
Justificación	El uso de un balanceador de carga es esencial para distribuir de manera eficiente las solicitudes entre las instancias, evitando cuellos de botella y mejorando la disponibilidad del sistema. La mensajería asíncrona permite distribuir tareas de forma flexible y permite que las instancias procesen solicitudes de manera independiente, manteniendo el rendimiento incluso en picos de carga. La escalabilidad dinámica permite ajustar el número de instancias automáticamente según la demanda, lo que optimiza los costos operacionales. Estas decisiones arquitecturales aseguran que el sistema maneje eficientemente altos volúmenes de solicitudes, mejorando la disponibilidad, reduciendo la latencia y respondiendo adecuadamente ante picos de demanda		

Atributo de calidad 1: Escalabilidad

Escalabilidad con Concurrency (Load Balancer)

Diagrama de arquitectura





Hipótesis de diseño asociada al experimento	
Punto de sensibilidad	El balanceador de carga debe distribuir eficientemente las solicitudes entre las instancias disponibles, asegurando que ninguna supere el 75% de su capacidad y evitando cuellos de botella en el procesamiento. Además, debe redistribuir la carga si una instancia falla sin interrumpir el flujo de solicitudes.
Historia de arquitectura asociada	Como sistema de procesamiento de imágenes, cuando se reciben paquetes de imágenes para su procesamiento, dado que las instancias pueden enfrentar picos de demanda, quiero que el balanceador distribuya la carga dinámicamente, de manera eficiente y tolerante a fallos, para garantizar tiempos de respuesta óptimos y evitar sobrecargas en instancias individuales. Esto debe suceder en tiempo real y debe procesar al menos el 99.9% de las solicitudes de imágenes sin superar un tiempo de respuesta de 2 segundos por paquete.
Nivel de incertidumbre	El nivel de incertidumbre es medio, ya que el éxito del experimento depende en un 100% de la capacidad del balanceador de carga para gestionar eficientemente los picos de demanda, evitando latencias elevadas y cuellos de botella. Se reconoce que pueden ocurrir retrasos en la sincronización de instancias o en la redistribución de la carga en caso de fallas.

Análisis de los resultados obtenidos

- 1- Indique si la hipótesis de diseño pudo ser confirmada o no
- 2- En caso de que la hipótesis se haya confirmado, explique las decisiones de arquitectura que favorecieron el resultado
- 3- En caso de que los resultados del experimento no hayan sido favorables, explique por qué y cuáles cambios realizaría en el diseño

1. El sistema es capaz de distribuir las solicitudes entre las instancias disponibles, sin que ninguna supere el 75% de su capacidad.
2. La hipótesis se cumplió gracias a la aplicación del patrón del load balancer, la escalabilidad dinámica, la arquitectura utilizando mensajería asíncrona por medio de apache pulsar entre el bff y el ms de procesamiento, además de la táctica de Monitor la cual le permitieron al sistema identificar con rapidez las instancias que presentaban errores y retirarlas de la operación del sistema. Debido al desacoplamiento del microservicio de procesamiento pudo escalar de manera independiente, escuchando los eventos del bff para que cuando se disparara un comando, ejecutara el evento.

Se obtuvieron los siguientes resultados:

Buen rendimiento: Tiempo de respuesta promedio de 228 ms, lo cual es rápido (menos de 2 segundos).

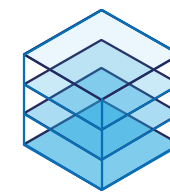
Baja latencia mínima (90 ms), lo que indica que el servidor puede manejar respuestas rápidas en algunos casos.

Sin errores (0.00%), lo que significa que todas las solicitudes fueron exitosas.

Tiempo máximo de 1,350 ms: Aunque el promedio es bajo, algunas solicitudes tardaron más de 1 segundo. Puede ser una señal de sobrecarga en momentos puntuales.

Desviación estándar de 179.42 ms, lo que indica cierta variabilidad en los tiempos de respuesta.

Ninguna instancia supera el 75% de uso, lo máximo que alcanza una instancia es el 49% de uso de CPU.



Evidencias

1- Presente evidencias de los resultados obtenidos en el experimento.

Se configura el JMeter:

Propiedades del Hilo:

Número de Hilos: 1000

→ Simula 1000 usuarios virtuales concurrentes en la prueba.

Periodo de Subida (Ramp-Up Period) en segundos: 60

→ Los 1000 hilos se iniciarán en 60 segundos, es decir, JMeter agregará aproximadamente 16-17 usuarios por segundo hasta alcanzar los 1000.

Contador del Bucle: 4

→ Cada hilo ejecutará 4 iteraciones antes de finalizar. Si estuviera en "Sin fin", se ejecutaría de manera indefinida.

Posible impacto:

Esta configuración genera una carga muy alta en un tiempo relativamente corto (60s). Si el sistema no está bien preparado, podría colapsar debido al número de solicitudes simultáneas.

Después se configura el JMeter con la dirección del balanceador de carga:

<http://34.54.118.91:80>

Utilizando el endpoint del BFF: '/procesar/procesar-imagen', methods=['POST']

Header:

Content-Type = json/application

Body

```
{
  "id_imagen": "46769a88-00000-4468o-8a14-1579887d01",
  "modalidad": "Rayos X",
  "patologia": "Neumonía",
  "region_anatomica": "Cráneo",
  "formato_imagen": "DICOM",
  "fuente_de_datos": "Hospital del norte",
  "antecedentes": "Diabético",
  "id_paciente" : "46769a88-181f-4468-8i34-15f750381d01",
  "fecha_ingesta": "2024-02-01 09:10:00"
}
```

Grupo de Hilos

Nombre: Pruebas Saludtech

Comentarios

Acción a tomar después de un error de Muestreador

☒ Continuar ☐ Comenzar siguiente iteración ☐ Parar Hilo ☐ Parar Test ☐ Parar test ahora

Propiedades de Hilo

Número de Hilos: 1000

Periodo de Subida (en segundos): 60

Contador del bucle: ☐ Sin fin ☒ 4

☒ Same user on each iteration

☐ Retrasar la creación de Hilos hasta que se necesiten

☐ Planificador



Evidencia Jmeter

Petición HTTP

Nombre:

Comentarios:

Basic Advanced

Servidor Web

Protocolo: Nombre de Servidor o IP: Puerto:

Petición HTTP

Ruta: Codificación del contenido:

☐ Enviar automáticamente ☒ Seguir Redirecciones ☒ Usar Facilita ☐ Usar multipart/form-data para HTTP POST ☐ Opciones compatibles con navegadores

Parámetros Body Data Files Upload

```

1 {
2   "id_usuario": "46251428-8888-4080-8410-13798374017",
3   "email_usuario": "jorge@37",
4   "password_usuario": "12345678",
5   "region_anterior": "Cuba",
6   "nombre_usuario": "Jorge",
7   "nombre_de_usuario": "jorge1541@okl.com",
8   "password_usuario": "12345678",
9   "id_usuario": "46251428-8888-4080-8410-13798374017"
10 }

```




Evidencias

1- Presente evidencias de los resultados obtenidos en el experimento.

Reporte resumen

Nombre:	Reporte resumen
---------	-----------------

Comentarios

-Escribir todos los datos a Archivo-

Nombre de archivo

Navegar...

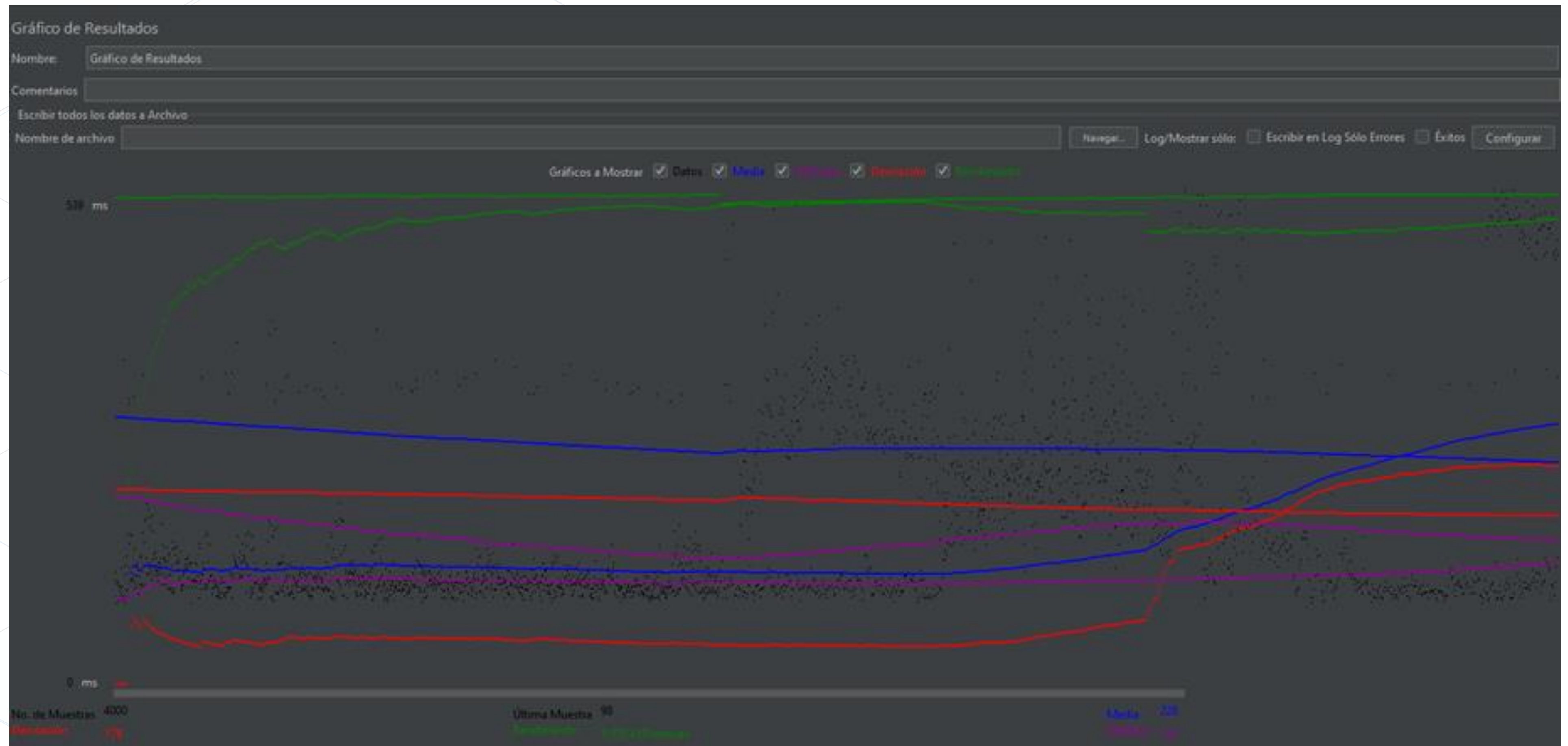
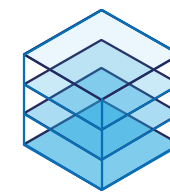
Log/Mostrar sólo:

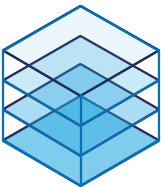
Escribir en Log Sólo Errores

Éxitos

Configurar

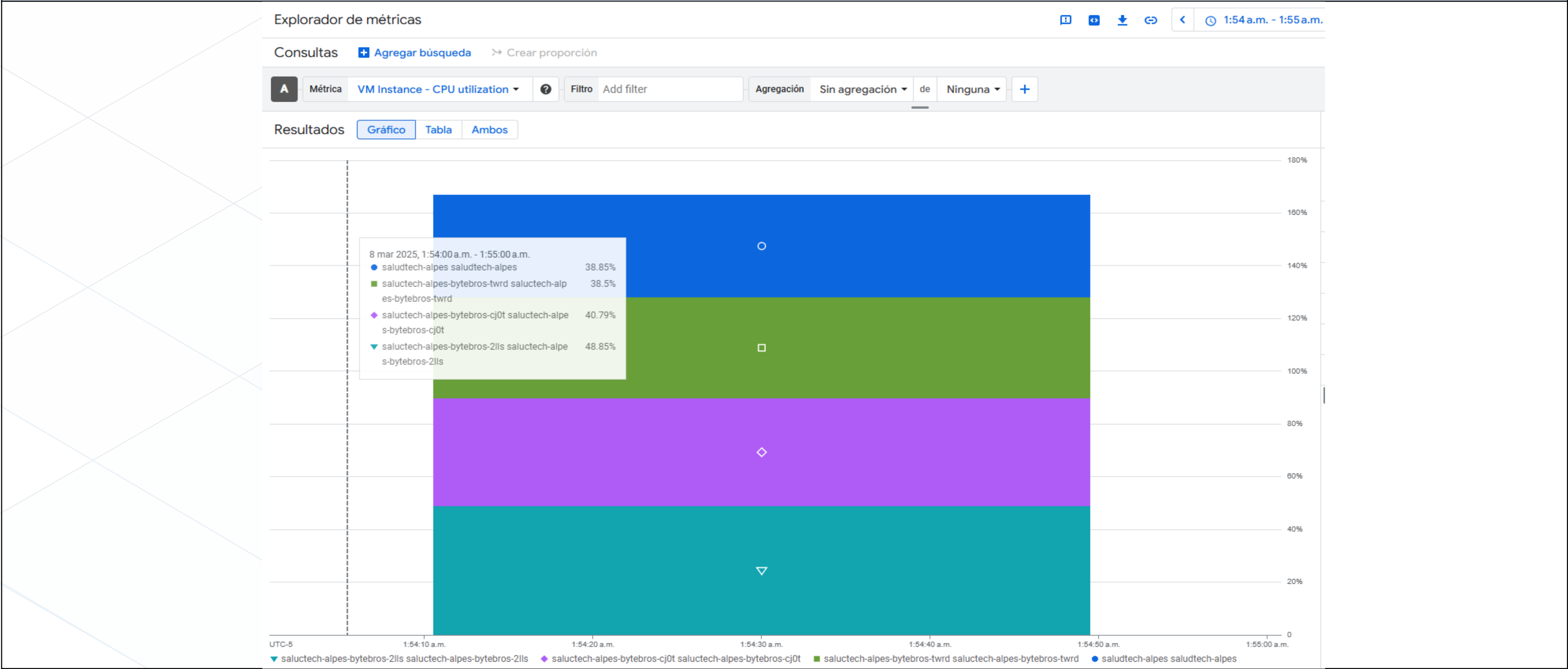
[illegible]





Evidencias

1- Presente evidencias de los resultados obtenidos en el experimento.





Evidencias

1- Presente evidencias de los resultados obtenidos en el experimento.

Explorador de métricas

1:54 a.m. - 1:55 a.m.

Consultas

+ Agregar búsqueda

>> Crear proporción

A

Métrica

VM Instance - CPU utilization

?

Filtro

Add filter

Agregación

Sin agregación

de

Ninguna

+

Resultados

Gráfico

Tabla

Ambos

Filtro

Ingresar el nombre o el valor de la propiedad

?

|||

<input type="checkbox"/>	Nombre (de instance_id) ↑	instance_name	Value
<input type="checkbox"/>	▼ saluctech-alpes-bytebros-2lls saluctech-alpes-bytebros-2lls	saluctech-alpes-bytebros-2lls	39.494 %
<input type="checkbox"/>	◆ saluctech-alpes-bytebros-cj0t saluctech-alpes-bytebros-cj0t	saluctech-alpes-bytebros-cj0t	62.653 %
<input type="checkbox"/>	■ saluctech-alpes-bytebros-twrd saluctech-alpes-bytebros-twrd	saluctech-alpes-bytebros-twrd	38.214 %
<input type="checkbox"/>	● saludtech-alpes saludtech-alpes	saludtech-alpes	38.773 %



Evidencia Balanceador de carga GCP

←

Detalles del balanceador de cargas

EDITAR

BORRAR

saludtech-alpes-lb

Balanceador de cargas de aplicaciones externo global

Rendimiento web más rápido y mayor protección web con Cloud CDN y Cloud Armor. [Más información](#)

IGNORAR

DETALLES

MONITORING

ALMACENAMIENTO EN CACHE

Frontend

Protocolo ↑	IP:Puerto	Certificado	Mapa de certificados	Política de SSL	Nivel de red ?	Tiempo de espera keepalive de HTTP ?
HTTP	34.54.118.91:80	-			Premium	610 segundos

Normas de enrutamiento

Hosts ↑	Rutas	Backend
Todos los que no coincidan (predeterminado)	Todos los que no coincidan (predeterminado)	microservicios-backend

Backend

Servicios de backend

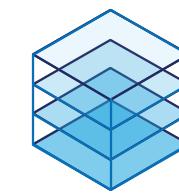
1. microservicios-backend

Protocolo de extremo	HTTP
Puerto con nombre	http
Tiempo de espera	30 segundos
Política de selección de direcciones IP ?	Solo IPv4
Verificación de estado	microservicios-saludtech
Cloud CDN	Habilitada VER DETALLES
Registros	Inhabilitada
Política de seguridad del backend	Ninguna

✓

MOSTRAR OPCIONES AVANZADAS

Evidencia grupo de contenedores en GCP



- Se creó el grupo de contenedores en GCP, un grupo de mínimo 1, máximo 3, como regla que al alcanzar el 75% de capacidad de CPU, se redirigiera a otra instancia. Para este grupo de instancias, se usaba una plantilla de instancia con las instrucciones para correr el proyecto.

[←](#) Edición de saludtech-alpes-bytebros [CONVERTIR A CON ESTADO](#)

saludtech-alpes-pl-22

Cantidad de instancias
Según la configuración del ajuste de escala automático

Ajuste de escala automático

Usa el ajuste de escala automático para agregar y quitar instancias de forma automática en el grupo durante los períodos de cargas altas y bajas. [Más información](#)

Modo de ajuste de escala automático
Activado: agrega y quita instancias del grupo

Número mínimo de instancias *
1

Número máximo de instancias *
3

Autoscaling signals

Usa los indicadores para determinar cuándo escalar el grupo. [Más información](#)

▼ Uso de CPU: 75%
El ajuste de escala automático predictivo está off

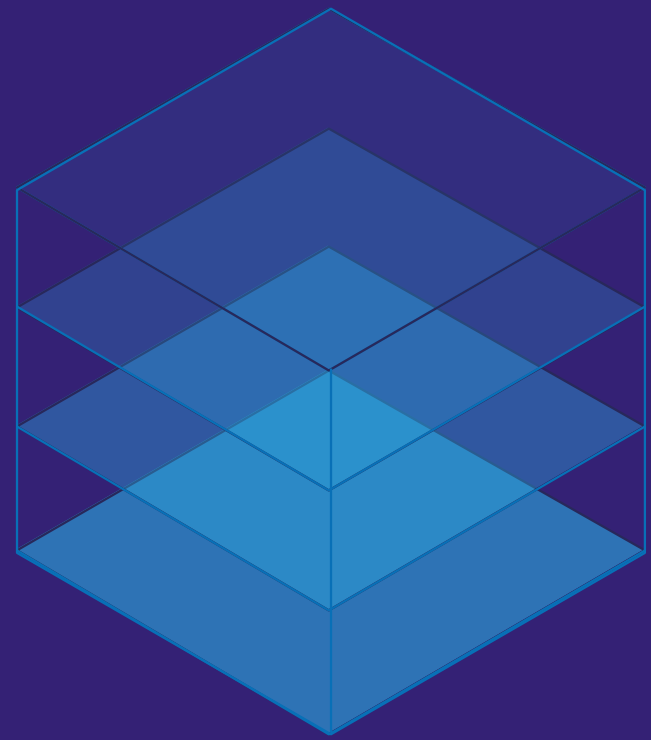
AGREGAR UN INDICADOR

Autoscaling schedules

Período de inicialización

Especifica cuánto le toma a la app inicializarse desde el inicio hasta que está lista para la entrega. [Más información](#)

Período de inicialización *
60 segundos



MISO

Maestría en Ingeniería de Software

Experimento 2 - Seguridad Integridad y seguridad de los datos

Atributo de calidad: Seguridad

Escenario de calidad: Integridad y seguridad de los datos			
Escenario #: 6	El sistema debe garantizar la integridad y seguridad de los datos en las transacciones entre los microservicios de suscripciones, servicio de datos y notificaciones, asegurando que no haya inconsistencias o fallos en el flujo de información.		
Fuente	Clientes suscrito a un plan de servicio		
Estímulo	Solicitud de suscripción y procesamiento de datos asociado a la entrega del servicio.		
Ambiente	Producción con múltiples transacciones concurrentes.		
Artefacto	Microservicios de suscripciones, servicio de datos y notificaciones.		
Respuesta	Garantizar la seguridad de las transacciones mediante compensaciones en caso de fallos, evitando inconsistencias y pérdida de datos.		
Medida de la respuesta	Se deben registrar el 100% de las transacciones sin inconsistencias en las bases de datos o fallos en el flujo de información.		
Decisiones Arquitecturales	Punto de sensibilidad	Tradeoff	Riesgo
Uso del patrón Saga para la gestión de transacciones distribuidas	Complejidad en la implementación y monitoreo de las compensaciones.	Mayor latencia al dividir una transacción en múltiples pasos, mecanismos adicionales para la gestión de estado.	Inconsistencia temporal en la ejecución de la Saga, fallos en la compensación, sobrecarga del sistema de mensajería.
Uso de Apache Pulsar para la persistencia de estados	Complejidad en el orden de los eventos.	Mayor complejidad en la implementación.	Sobrecarga en el sistema por acumulación de eventos, generando cuellos de botella.
Justificación	Para garantizar la integridad y seguridad de las transacciones en los microservicios de suscripciones, servicio de datos y notificaciones, se requiere un mecanismo de gestión de transacciones distribuidas como el Patrón Saga. Además, la implementación de Apache Pulsar permite un registro inmutable de eventos con capacidad de reprocesamiento en caso de generación de fallos.		

Integridad y seguridad de datos

El diagrama de arquitectura superior muestra la interacción entre varios componentes: BFF, SAGA, SUSCRIPCIONES, NOTIFICACIONES y SERVICIO DE DATOS. Se utilizan números 1, 2 y 3 para indicar el orden de las operaciones. Las acciones incluyen 'Pulsar', 'Crear suscripción', 'Crear infraestructura' y 'Crear notificación'. El diagrama inferior, titulado 'Diagrama de arquitectura', detalla el flujo de una Saga. Comienza con 'Inicio Saga' que llega al 'Saga Execution Coordinator'. Este coordina tres transacciones: 'Crear suscripción (Transacción 1)', 'Crear Infraestructura (Transacción 2)' y 'Crear Notificación (Transacción 3)'. Cada transacción envía señales de 'Inicio' y 'Fin' al coordinador. El coordinador registra estos eventos en un 'Log de Saga' y finalmente envía 'Fin Saga' al coordinador al completarse la última transacción.

Diagrama de arquitectura

Log de Saga

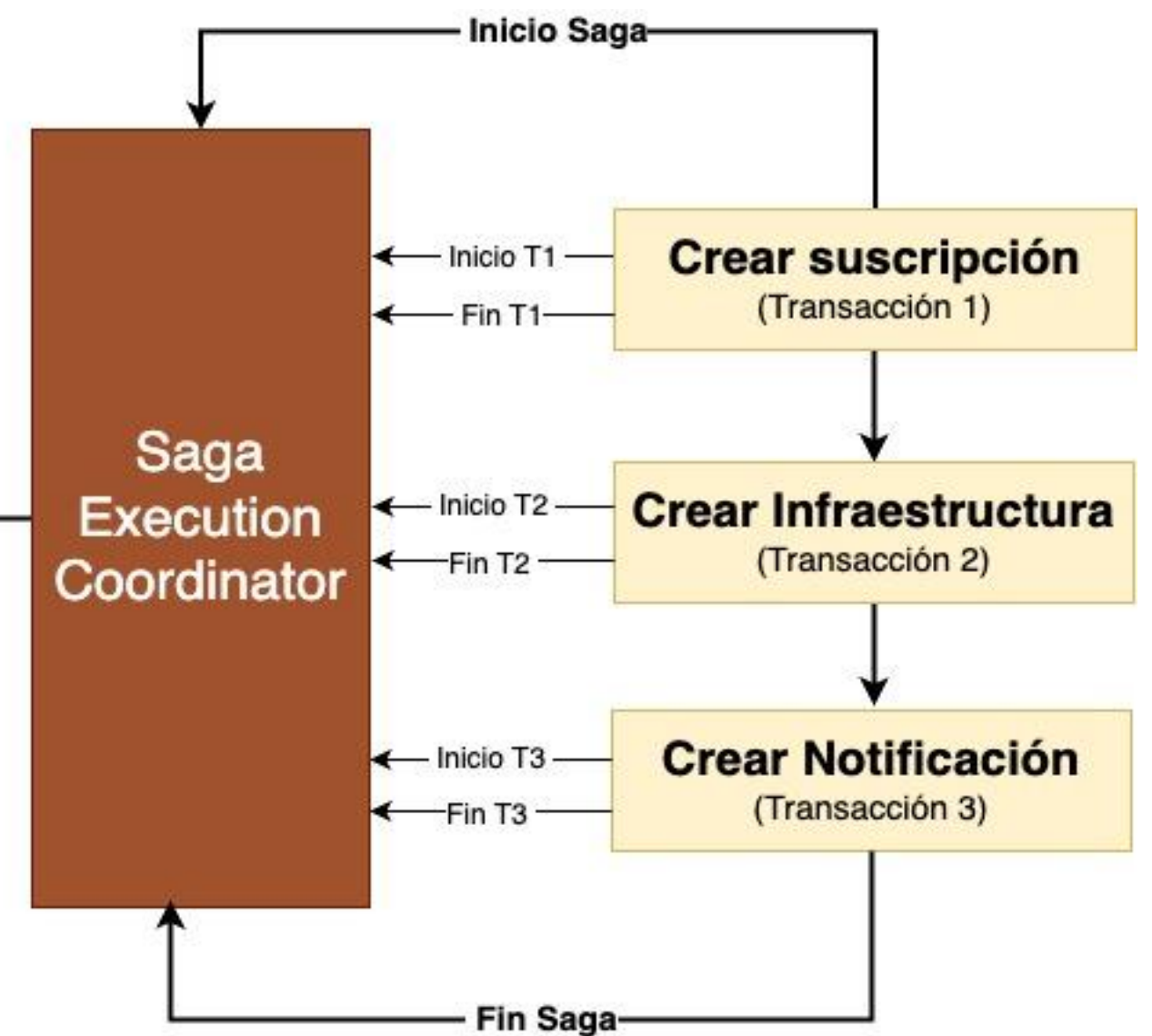
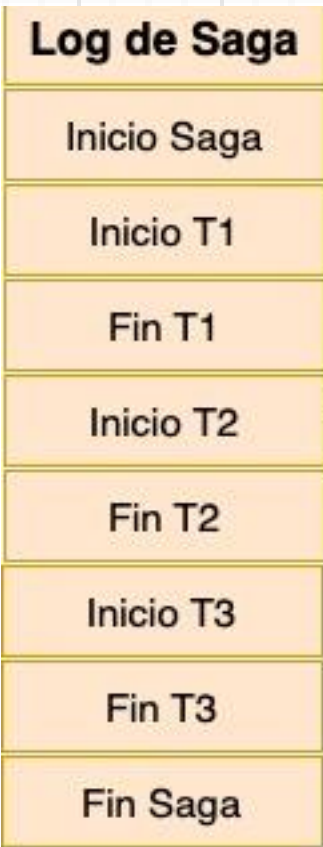
Inicio Saga
Inicio T1
Fin T1
Inicio T2
Fin T2
Inicio T3
Fin T3
Fin Saga

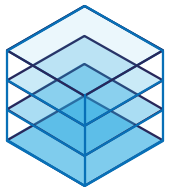
Saga Execution Coordinator

Crear suscripción (Transacción 1)

Crear Infraestructura (Transacción 2)

Crear Notificación (Transacción 3)





Hipótesis de diseño asociada al experimento	
Punto de sensibilidad	La coordinación de las transacciones distribuidas y la sincronización de los eventos entre microservicios . Específicamente, la capacidad de manejar fallos en la transmisión o procesamiento de datos, y la correcta implementación de las compensaciones en caso de errores. Si no se implementa correctamente el patrón Saga y Apache Pulsar para la persistencia de eventos, puede ocurrir desincronización de los estados entre microservicios, generando inconsistencias y afectando la seguridad de los datos.
Historia de arquitectura asociada	<p>El sistema de microservicios evolucionó para incorporar un mecanismo robusto para gestionar las transacciones distribuidas y mantener la integridad de los datos en entornos de alta concurrencia. Inicialmente, los microservicios de suscripciones, servicio de datos y notificaciones operaban de forma independiente, lo que generaba posibles inconsistencias durante la transmisión de datos.</p> <p>Para resolver estos problemas, se adoptó el Patrón Saga para coordinar transacciones distribuidas entre los microservicios. Esto permitió que cada microservicio manejara su propia transacción localmente, con la posibilidad de revertir las operaciones en caso de errores en los otros microservicios, implementando compensaciones.</p> <p>A medida que el sistema escalaba y las transacciones se multiplicaban, se incorporó Apache Pulsar para gestionar la persistencia de los estados de las transacciones y asegurar que los eventos estuvieran correctamente registrados y pudieran ser reprocesados en caso de fallos.</p> <p>Se implemento un Saga log donde se puede hacer trazabilidad del proceso de ejecución entre los microservicios.</p>
Nivel de incertidumbre	<p>El nivel de incertidumbre en este escenario se considera moderado a alto debido a la complejidad de implementar el patrón Saga en un sistema distribuido, lo que requiere una coordinación precisa entre los microservicios. Los fallos en la compensación o desincronización de los eventos podrían comprometer la integridad de los datos. Además, la sincronización de los eventos es crítica para mantener la consistencia del sistema, y si el orden de procesamiento no se gestiona correctamente, podría ocurrir desincronización, lo que afectaría la integridad de la información.</p> <p>El sistema también enfrenta incertidumbres relacionadas con su escalabilidad, ya que un aumento en la carga de trabajo o en el número de transacciones concurrentes podría generar problemas en el rendimiento o en el flujo de información. A pesar de que Apache Pulsar ofrece soluciones para la persistencia y reprocesamiento de eventos, el manejo de grandes volúmenes de datos podría generar cuellos de botella o pérdidas de información si no se configuran adecuadamente las políticas de retención y los límites de capacidad.</p>

Análisis de los resultados obtenidos

- 1- Indique si la hipótesis de diseño pudo ser confirmada o no
- 2- En caso de que la hipótesis se haya confirmado, explique las decisiones de arquitectura que favorecieron el resultado
- 3- En caso de que los resultados del experimento no hayan sido favorables, explique por qué y cuáles cambios realizaría en el diseño

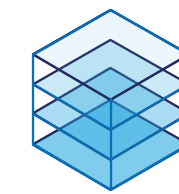
Sí, la hipótesis de diseño pudo ser confirmada. El uso del **Patrón Saga** con orquestación para gestionar las transacciones distribuidas y la implementación de **Apache Pulsar** para la persistencia de eventos garantizó la integridad y seguridad de los datos durante las transacciones entre los microservicios. A pesar de la alta concurrencia las transacciones se completaron correctamente sin inconsistencias ni pérdida de datos. Además, el sistema pudo manejar de manera eficiente los fallos y realizar compensaciones cuando fue necesario, sin comprometer la coherencia de los datos entre los microservicios.

La implementación del **Patrón Saga** jugó un papel fundamental en la gestión de las transacciones distribuidas, permitiendo que cada microservicio manejara sus propias transacciones locales, con la capacidad de revertir las operaciones en caso de fallo. Esta decisión arquitectural permitió manejar los errores de manera eficiente y coordinar las transacciones entre microservicios sin causar inconsistencias. Además, el uso de **Saga Log** para registrar los estados de las transacciones proporcionó una solución robusta que facilitó la auditoría y el reprocesamiento de eventos en caso de que se produjeran fallos.

La persistencia de eventos y la capacidad de reprocesarlos garantizó que el sistema pudiera recuperar los datos correctamente y evitar pérdidas.

Se obtuvieron los siguientes resultados:

De 100 procesos ejecutados 55 generaron compensación, el 100% de las transacciones quedaron registradas en el Saga Log y el 100% de las transacciones que generaron fallidas quedaron correctamente compensadas en cada microservicio.



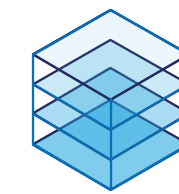
Evidencias

1- Presente evidencias de los resultados obtenidos en el experimento.

Para realizar las pruebas en los microservicios se introdujo el siguiente código para introducir error, por cada microservicio tenemos un 25% de probabilidad de generación de error con el fin de validar la compensación.

```
# Introducción de una falla aleatoria
estado = ['normal', 'error', 'normal', 'normal']
# 25%
if random.choice(estado) is 'error':
    raise 'Error generado aleatoriamente'
```

A continuación se encuentra la información almacenada en el Saga Log donde se muestra el estado, el paso ejecutado en cada uno de los eventos, una identificación única que permite hacer seguimiento dentro de toda la SAGA, esta identificación se almacena en cada microservicio para la gestión de la compensación y así determinar si se ejecuto.



Evidencias

1- Presente evidencias de los resultados obtenidos en el experimento.

Para realizar la ejecución de las pruebas se utilizo el endpoint del BFF: '/sagas/iniciar-suscripcion-comando', methods=['POST']

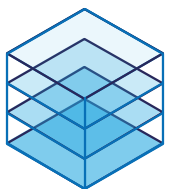
Header:

Content-Type = json/application

Body

```
{  
  "id_suscripcion": "{{${randomUUID}}}",  
  "cliente_codigo": "0001",  
  "cliente_nombres": "Pablo Saga",  
  "cliente_apellidos": "Perez Prieto",  
  "cliente_usuario": "pperez",  
  "cliente_rut": "1234567890",  
  "cliente_cedula": "987654321",  
  "cliente_email": "pperez@domain.com",  
  "plan_codigo": "pro",  
  "plan_nombre": "PRO"  
}
```

El id_suscripcion corresponde a un valor random que corresponde al valor que se almacena en el Saga Log y en cada uno de los microservicios para seguimiento.



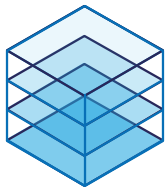
Evidencia – Saga Log

Proceso exitoso

	A-Z id_correlacion	123 index	A-Z descripcion	fecha_creacion
1	73ab98cf-4c6e-4005-ae0-da071fa63b5f	0	Inicio Saga	2025-03-10 20:44:21.223
2	73ab98cf-4c6e-4005-ae0-da071fa63b5f	1	Inicio T1	2025-03-10 20:44:21.223
3	73ab98cf-4c6e-4005-ae0-da071fa63b5f	1	Fin Existoso T1	2025-03-10 20:44:21.223
4	73ab98cf-4c6e-4005-ae0-da071fa63b5f	2	Inicio T2	2025-03-10 20:44:21.223
5	73ab98cf-4c6e-4005-ae0-da071fa63b5f	2	Fin Existoso T2	2025-03-10 20:44:21.223
6	73ab98cf-4c6e-4005-ae0-da071fa63b5f	0	Fin Saga	2025-03-10 20:44:21.223

Proceso fallido – compensación

	A-Z id_correlacion	123 index	A-Z descripcion	fecha_creacion
1	bb3172ce-1a9e-44f2-a605-641ceed62c3a	0	Inicio Saga	2025-03-10 22:02:42.443
2	bb3172ce-1a9e-44f2-a605-641ceed62c3a	1	Inicio T1	2025-03-10 22:02:42.443
3	bb3172ce-1a9e-44f2-a605-641ceed62c3a	1	Fin Existoso T1	2025-03-10 22:02:42.443
4	bb3172ce-1a9e-44f2-a605-641ceed62c3a	2	Inicio T2	2025-03-10 22:02:42.443
5	bb3172ce-1a9e-44f2-a605-641ceed62c3a	2	Fin Fallido T2	2025-03-10 22:02:42.443
6	bb3172ce-1a9e-44f2-a605-641ceed62c3a	1	Compensacion T1	2025-03-10 22:02:42.443
7	bb3172ce-1a9e-44f2-a605-641ceed62c3a	3	Fin Saga	2025-03-10 22:02:42.443



Script

Script-2

Script-3

select saga_logs.id_correlacion, saga_logs."index", saga_logs.descripcion, saga_logs.fecha_creacion from public.saga_logs order by saga_logs.id_correlacion , "index"

saga_logs 1

Enter a SQL expression to filter results (use Ctrl+Space)

Grid

Text

Record

	A-Z id_correlacion	123 index	A-Z descripcion	fecha_creacion
1	2307f9b0-7da9-4bad-990b-a35e372decdf	0	Inicio Saga	2025-03-10 22:02:42.443
2	2307f9b0-7da9-4bad-990b-a35e372decdf	1	Inicio T1	2025-03-10 22:02:42.443
3	2307f9b0-7da9-4bad-990b-a35e372decdf	1	Fin Fallido T1	2025-03-10 22:02:42.443
4	2307f9b0-7da9-4bad-990b-a35e372decdf	3	Fin Saga	2025-03-10 22:02:42.443
5	2a69089a-aa8b-4f06-86c9-ae99449a55d8	0	Inicio Saga	2025-03-10 22:02:42.443
6	2a69089a-aa8b-4f06-86c9-ae99449a55d8	1	Fin Existoso T1	2025-03-10 22:02:42.443
7	2a69089a-aa8b-4f06-86c9-ae99449a55d8	1	Inicio T1	2025-03-10 22:02:42.443
8	2a69089a-aa8b-4f06-86c9-ae99449a55d8	2	Inicio T2	2025-03-10 22:02:42.443
9	2a69089a-aa8b-4f06-86c9-ae99449a55d8	2	Fin Existoso T2	2025-03-10 22:02:42.443
10	2a69089a-aa8b-4f06-86c9-ae99449a55d8	3	Fin Saga	2025-03-10 22:02:42.443
11	2fc1a4d3-ccab-4697-bf0a-bb96f2809c06	0	Inicio Saga	2025-03-10 22:02:42.443
12	2fc1a4d3-ccab-4697-bf0a-bb96f2809c06	1	Fin Fallido T1	2025-03-10 22:02:42.443
13	2fc1a4d3-ccab-4697-bf0a-bb96f2809c06	1	Inicio T1	2025-03-10 22:02:42.443
14	2fc1a4d3-ccab-4697-bf0a-bb96f2809c06	3	Fin Saga	2025-03-10 22:02:42.443
15	3108a6bf-dded-45fb-ba08-2625231bf755	0	Inicio Saga	2025-03-10 22:02:42.443
16	3108a6bf-dded-45fb-ba08-2625231bf755	1	Inicio T1	2025-03-10 22:02:42.443
17	3108a6bf-dded-45fb-ba08-2625231bf755	1	Fin Fallido T1	2025-03-10 22:02:42.443
18	3108a6bf-dded-45fb-ba08-2625231bf755	3	Fin Saga	2025-03-10 22:02:42.443
19	32749283-864a-4669-ba7d-f563ee5bdf01	0	Inicio Saga	2025-03-10 22:02:42.443
20	32749283-864a-4669-ba7d-f563ee5bdf01	1	Inicio T1	2025-03-10 22:02:42.443
21	32749283-864a-4669-ba7d-f563ee5bdf01	1	Fin Existoso T1	2025-03-10 22:02:42.443
22	32749283-864a-4669-ba7d-f563ee5bdf01	2	Fin Existoso T2	2025-03-10 22:02:42.443
23	32749283-864a-4669-ba7d-f563ee5bdf01	2	Inicio T2	2025-03-10 22:02:42.443
24	32749283-864a-4669-ba7d-f563ee5bdf01	3	Fin Saga	2025-03-10 22:02:42.443
25	4776cc15-d944-4ec9-ae61-a296e6be798d	0	Inicio Saga	2025-03-10 22:02:42.443
26	4776cc15-d944-4ec9-ae61-a296e6be798d	1	Inicio T1	2025-03-10 22:02:42.443
27	4776cc15-d944-4ec9-ae61-a296e6be798d	1	Fin Existoso T1	2025-03-10 22:02:42.443
28	4776cc15-d944-4ec9-ae61-a296e6be798d	2	Inicio T2	2025-03-10 22:02:42.443
29	4776cc15-d944-4ec9-ae61-a296e6be798d	2	Fin Existoso T2	2025-03-10 22:02:42.443
30	4776cc15-d944-4ec9-ae61-a296e6be798d	3	Fin Saga	2025-03-10 22:02:42.443
31	656d899a-2ade-4ef4-93aa-fae2ab9ea953	0	Inicio Saga	2025-03-10 22:02:42.443
32	656d899a-2ade-4ef4-93aa-fae2ab9ea953	1	Inicio T1	2025-03-10 22:02:42.443
33	656d899a-2ade-4ef4-93aa-fae2ab9ea953	1	Fin Existoso T1	2025-03-10 22:02:42.443
34	656d899a-2ade-4ef4-93aa-fae2ab9ea953	2	Inicio T2	2025-03-10 22:02:42.443
35	656d899a-2ade-4ef4-93aa-fae2ab9ea953	2	Fin Existoso T2	2025-03-10 22:02:42.443
36	656d899a-2ade-4ef4-93aa-fae2ab9ea953	3	Fin Saga	2025-03-10 22:02:42.443
37	6673a909-46a4-42bb-bd90-d737e5ae9d1b	0	Inicio Saga	2025-03-10 22:02:42.443
38	6673a909-46a4-42bb-bd90-d737e5ae9d1b	1	Fin Existoso T1	2025-03-10 22:02:42.443
39	6673a909-46a4-42bb-bd90-d737e5ae9d1b	1	Inicio T1	2025-03-10 22:02:42.443
40	6673a909-46a4-42bb-bd90-d737e5ae9d1b	2	Inicio T2	2025-03-10 22:02:42.443
41	6673a909-46a4-42bb-bd90-d737e5ae9d1b	2	Fin Existoso T2	2025-03-10 22:02:42.443
42	6673a909-46a4-42bb-bd90-d737e5ae9d1b	3	Fin Saga	2025-03-10 22:02:42.443
43	71f4ef74-8e97-4a3b-a722-c2200354f8ea	0	Inicio Saga	2025-03-10 22:02:42.443
44	71f4ef74-8e97-4a3b-a722-c2200354f8ea	1	Compensacion T1	2025-03-10 22:02:42.443
45	71f4ef74-8e97-4a3b-a722-c2200354f8ea	1	Inicio T1	2025-03-10 22:02:42.443
46	71f4ef74-8e97-4a3b-a722-c2200354f8ea	1	Fin Existoso T1	2025-03-10 22:02:42.443
47	71f4ef74-8e97-4a3b-a722-c2200354f8ea	2	Fin Fallido T2	2025-03-10 22:02:42.443
48	71f4ef74-8e97-4a3b-a722-c2200354f8ea	2	Inicio T2	2025-03-10 22:02:42.443
49	71f4ef74-8e97-4a3b-a722-c2200354f8ea	3	Fin Saga	2025-03-10 22:02:42.443
50	7ae3c76a-5450-4714-aa74-007dbcdaab62	0	Inicio Saga	2025-03-10 22:02:42.443
51	7ae3c76a-5450-4714-aa74-007dbcdaab62	1	Inicio T1	2025-03-10 22:02:42.443
52	7ae3c76a-5450-4714-aa74-007dbcdaab62	1	Fin Existoso T1	2025-03-10 22:02:42.443
53	7ae3c76a-5450-4714-aa74-007dbcdaab62	2	Inicio T2	2025-03-10 22:02:42.443
54	7ae3c76a-5450-4714-aa74-007dbcdaab62	2	Fin Existoso T2	2025-03-10 22:02:42.443
55	7ae3c76a-5450-4714-aa74-007dbcdaab62	3	Fin Saga	2025-03-10 22:02:42.443
56	7e0090ba-5961-41e9-be49-e56ddd0c5923	0	Inicio Saga	2025-03-10 22:02:42.443
57	7e0090ba-5961-41e9-be49-e56ddd0c5923	1	Compensacion T1	2025-03-10 22:02:42.443

Refresh

Save

Cancel

Export data

200

111

111 row(s) fetched - 0.002s, on 2025-03-10 at 17:39:17

Evidencia – Base de datos microservicios

Saga Log

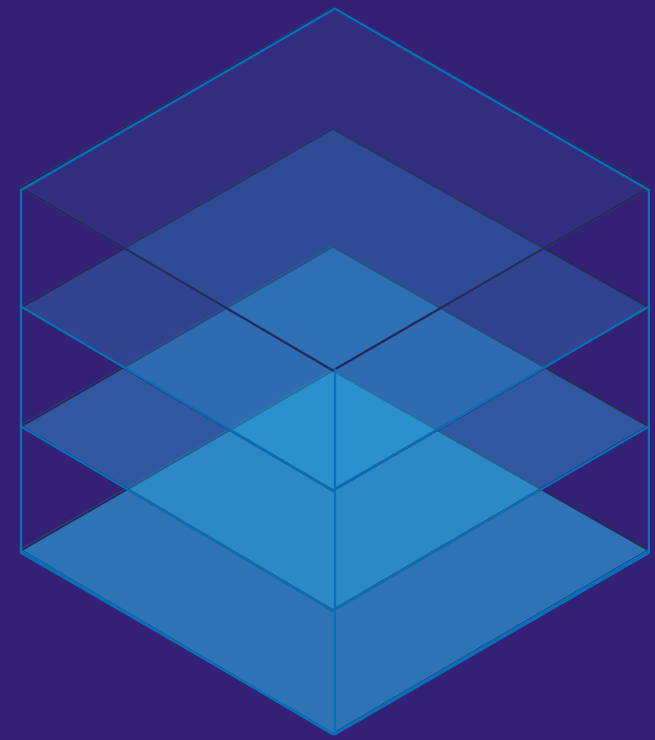
id correlacion	index	log	hora
2307f9b0-7da9-4bad-990b-a35e372decdf	0	Inicio Saga	02:42.4
2307f9b0-7da9-4bad-990b-a35e372decdf	1	Inicio T1	02:42.4
2307f9b0-7da9-4bad-990b-a35e372decdf	1	Fin Fallido T1	02:42.4
2307f9b0-7da9-4bad-990b-a35e372decdf	3	Fin Saga	02:42.4
2a69089a-aa8b-4f06-86c9-ae99449a55d8	0	Inicio Saga	02:42.4
2a69089a-aa8b-4f06-86c9-ae99449a55d8	1	Fin Existoso T1	02:42.4
2a69089a-aa8b-4f06-86c9-ae99449a55d8	1	Inicio T1	02:42.4
2a69089a-aa8b-4f06-86c9-ae99449a55d8	2	Inicio T2	02:42.4
2a69089a-aa8b-4f06-86c9-ae99449a55d8	2	Fin Existoso T2	02:42.4
2a69089a-aa8b-4f06-86c9-ae99449a55d8	3	Fin Saga	02:42.4
2fc1a4d3-ccab-4697-bf0a-bb96f2809c06	0	Inicio Saga	02:42.4
2fc1a4d3-ccab-4697-bf0a-bb96f2809c06	1	Fin Fallido T1	02:42.4
2fc1a4d3-ccab-4697-bf0a-bb96f2809c06	1	Inicio T1	02:42.4
2fc1a4d3-ccab-4697-bf0a-bb96f2809c06	3	Fin Saga	02:42.4
3108a6bf-dded-45fb-ba08-2625231bf755	0	Inicio Saga	02:42.4
3108a6bf-dded-45fb-ba08-2625231bf755	1	Inicio T1	02:42.4
3108a6bf-dded-45fb-ba08-2625231bf755	1	Fin Fallido T1	02:42.4
3108a6bf-dded-45fb-ba08-2625231bf755	3	Fin Saga	02:42.4
32749283-864a-4669-ba7d-f563ee5bdf01	0	Inicio Saga	02:42.4
32749283-864a-4669-ba7d-f563ee5bdf01	1	Fin Existoso T1	02:42.4
32749283-864a-4669-ba7d-f563ee5bdf01	2	Fin Existoso T2	02:42.4
32749283-864a-4669-ba7d-f563ee5bdf01	2	Inicio T2	02:42.4
32749283-864a-4669-ba7d-f563ee5bdf01	3	Fin Saga	02:42.4
4776cc15-d944-4ec9-ae61-a296e6be798d	0	Inicio Saga	02:42.4
4776cc15-d944-4ec9-ae61-a296e6be798d	1	Inicio T1	02:42.4
4776cc15-d944-4ec9-ae61-a296e6be798d	1	Fin Existoso T1	02:42.4
4776cc15-d944-4ec9-ae61-a296e6be798d	2	Inicio T2	02:42.4
4776cc15-d944-4ec9-ae61-a296e6be798d	2	Fin Existoso T2	02:42.4
4776cc15-d944-4ec9-ae61-a296e6be798d	3	Fin Saga	02:42.4
656d899a-2ade-4ef4-93aa-fae2ab9ea953	0	Inicio Saga	02:42.4
656d899a-2ade-4ef4-93aa-fae2ab9ea953	1	Inicio T1	02:42.4
656d899a-2ade-4ef4-93aa-fae2ab9ea953	1	Fin Existoso T1	02:42.4
656d899a-2ade-4ef4-93aa-fae2ab9ea953	2	Inicio T2	02:42.4
656d899a-2ade-4ef4-93aa-fae2ab9ea953	2	Fin Existoso T2	02:42.4
656d899a-2ade-4ef4-93aa-fae2ab9ea953	3	Fin Saga	02:42.4
6673a909-46a4-42bb-bd90-d737e5ae9d1b	0	Inicio Saga	02:42.4
6673a909-46a4-42bb-bd90-d737e5ae9d1b	1	Fin Existoso T1	02:42.4
6673a909-46a4-42bb-bd90-d737e5ae9d1b	1	Inicio T1	02:42.4
6673a909-46a4-42bb-bd90-d737e5ae9d1b	2	Inicio T2	02:42.4
6673a909-46a4-42bb-bd90-d737e5ae9d1b	2	Fin Existoso T2	02:42.4
6673a909-46a4-42bb-bd90-d737e5ae9d1b	3	Fin Saga	02:42.4

Suscripciones

Id suscripcion	id cliente	id plan	creacion	moficiaci	estad
9565f927-000a-4c94-9f22-c0fc380aa98d	c4d7bad0-ec1e-4e4c-9d01-e9b5a2d6e46b	c7f6cee9-9a19-478a-87d6-b6131020f42e	13:18.1	13:18.1	Activo
6673a909-46a4-42bb-bd90-d737e5ae9d1b	91f40f2f-d636-4afc-92f7-a11498d813fd	7f0870be-0ca4-464e-a943-39f8d92f0702	13:18.1	13:18.1	Activo
656d899a-2ade-4ef4-93aa-fae2ab9ea953	5c92adc4-2075-464d-a28f-7955c4ee82c4	7f463874-e25b-4c1d-9f7f-2a56d6e39118	13:18.1	13:18.1	Activo
b4d657dd-721d-4804-a236-43d3f5881af5	e4a0e693-b22d-4f16-9f2f-d1ebdf9cc2c2	69914fd6-5f90-4684-bf44-7589c7952474	13:18.1	13:18.1	Activo
ecc01964-5111-4d0b-b99d-d23f832b98d7	ad482c4c-a083-4931-9955-190b39a3d66d	1b679421-f0da-4715-9e17-abd44abdde0e	13:18.1	13:18.1	Activo
4776cc15-d944-4ec9-ae61-a296e6be798d	891433dc-9cb2-4137-aac1-39c2510052ba	a0ddb7f5-b083-4d1f-b3c0-bd72a1162c8f	13:18.1	13:18.1	Activo
32749283-864a-4669-ba7d-f563ee5bdf01	c5f81d9a-e59a-47cf-878c-bb02e1c2df26	2412046b-3059-416a-ab4e-959446b24cad	13:18.1	13:18.1	Activo
f39fd1b6-2a13-41fb-87c5-c23c20ea9152	f4f3beab-442f-44b5-b37f-5660fd01f8f5	88f7b2cb-5305-4ec8-97a6-5138db3b0d4a	13:18.1	13:18.1	Activo
a86e17be-b6c1-4655-b3c2-b5841fd366ae	bba835d0-1587-4451-9371-686987074455	8a253d71-49d3-4745-a040-627f4ca57607	13:18.1	13:18.1	Activo
2a69089a-aa8b-4f06-86c9-ae99449a55d8	d88ecbce-0381-4ab5-893f-7ce87f4c2360	ec9a5cbe-8ac4-4652-9adb-8e84db28158d	13:18.1	13:18.1	Activo
7ae3c76a-5450-4714-aa74-007dbcdaab62	90fa6ff4-0fa5-4a77-97c8-7b661fe6cc01	900b37ee-32db-4bb6-a994-73f7e69f8fdb	13:18.1	13:18.1	Activo

Servicio de datos

Id servicio	Id suscripcion
fbb4e961-cad5-466c-a447-95f690f2f0ad	9565f927-000a-4c94-9f22-c0fc380aa98d
c693fb99-1336-46ed-866f-3328f3524a89	6673a909-46a4-42bb-bd90-d737e5ae9d1b
43c4ceff-f84e-4c56-8e84-d0390ddb918c	656d899a-2ade-4ef4-93aa-fae2ab9ea953
de52b698-b9fe-4347-9f0f-8008bfc3a156	b4d657dd-721d-4804-a236-43d3f5881af5
08e3a214-153f-4566-a1a9-21b7620f897c	ecc01964-5111-4d0b-b99d-d23f832b98d7
8d8dd438-51df-4836-9b5a-a4f9cc5c287f	4776cc15-d944-4ec9-ae61-a296e6be798d
6a849437-a758-41f3-b722-d60f59de02f8	32749283-864a-4669-ba7d-f563ee5bdf01
6575f99e-fc5d-451c-8d33-f5d682ee04a8	f39fd1b6-2a13-41fb-87c5-c23c20ea9152
aaf40d9b-241a-4ba3-82a6-496cd3eab48b	a86e17be-b6c1-4655-b3c2-b5841fd366ae
07a9bf8a-5a13-4ebb-8501-e206a0fca3ee	2a69089a-aa8b-4f06-86c9-ae99449a55d8
b363ef6c-b4fc-483b-8912-dd37743c433d	7ae3c76a-5450-4714-aa74-007dbcdaab62



MISO

Maestría en Ingeniería de Software

Experimento 3 - Modificabilidad Modificaciones y extensibilidad en el proceso de suscripción

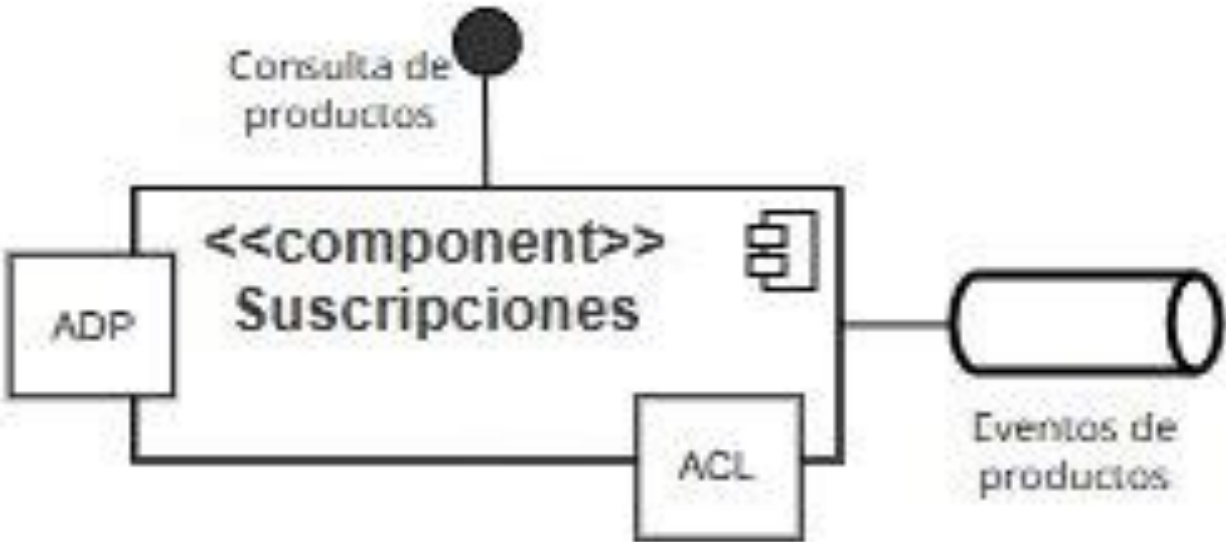
Atributo de calidad 3: Modificabilidad

Escenario de calidad: Modificaciones y extensibilidad en el proceso de suscripción			
Escenario #: 8	El sistema debe permitir modificaciones y extensibilidad en el proceso de suscripción, asegurando bajo acoplamiento y facilitando la integración de nuevos productos y cambios sin afectar la arquitectura existente.		
Fuente	Nuevos suscriptores (consumidores)		
Estímulo	Requerimiento de modificar o extender la lógica del proceso de suscripción para incluir nuevos productos o mejorar funcionalidades sin afectar componentes existentes.		
Ambiente	Desarrollo con evolución y mejoras continuas.		
Artefacto	Microservicios de suscripción		
Respuesta	Facilitar la evolución del sistema mediante patrones de reducir acoplamiento, utilizando adaptadores, mediadores y eventos.		
Medida de la respuesta	La integración de un nuevo producto en el proceso de suscripciones generó 0% de cambios en los microservicios existentes, manteniendo la estabilidad del sistema y asegurando la extensibilidad de la arquitectura.		
Decisiones Arquitecturales	Punto de sensibilidad	Tradeoff	Riesgo
Táctica de reducir el acoplamiento con el uso de adaptadores, mediadores y eventos en la lógica del microservicio de suscripción.	Reducción del acoplamiento y facilitar la evolución del sistema.	Latencia por la comunicación asíncrona, menor control en los flujos de negocio, impacto en las pruebas.	Complejidad en la gestión de eventos y mediadores, dificultad en la depuración y afectación en la integridad de los datos si no se hace una correcta gestión de las transacciones.
Justificación	Para permitir la evolución del sistema sin afectar los componentes existentes, es fundamental aplicar patrones de desacoplamiento. Respecto al uso de adaptadores facilita la conexión con nuevas fuentes de datos sin modificar el microservicio de suscripción, además los mediadores permiten gestionar la lógica de negocio de manera centralizada sin generar dependencias directas entre servicios y los eventos puedan ser manejados de manera extensible sin afectar la lógica interna del microservicio.		

Atributo de calidad: Seguridad

Integridad y seguridad de datos

Diagrama de arquitectura



Hipótesis de diseño asociada al experimento

Punto de sensibilidad	La gestión de eventos y mediadores es el punto crítico en el que podrían surgir problemas. Si los eventos no se gestionan correctamente, podría haber desincronización entre los servicios, lo que podría generar inconsistencias en el flujo de información. Además, el uso de mediadores aumenta la complejidad de la gestión centralizada de la lógica de negocio y puede dificultar la depuración si no se implementa de manera adecuada.
Historia de arquitectura asociada	Cada microservicio fue diseñado para ser autónomo y desacoplado, permitiendo la incorporación de nuevos productos y funcionalidades sin necesidad de modificar la lógica central del sistema. En este proceso de evolución, se implementaron patrones como adaptadores , mediadores y eventos para reducir aún más el acoplamiento entre los microservicios y facilitar la evolución continua del sistema.
Nivel de incertidumbre	<p>El nivel de incertidumbre se considera moderado a alto debido a la complejidad añadida por la implementación de eventos y mediadores. Estos patrones pueden generar problemas de desincronización entre los microservicios si no se gestionan correctamente, afectando la coherencia de los datos y la integridad de las transacciones. Además, el uso de comunicación asíncrona mediante eventos podría causar latencia, impactando el rendimiento y la capacidad de respuesta del sistema si la gestión de los eventos no es eficiente.</p> <p>Otro factor de incertidumbre es la complejidad en la depuración y las pruebas, ya que la lógica centralizada y la asincronía dificultan el monitoreo y la identificación de errores en el sistema. También, aunque la arquitectura es extensible, el creciente número de eventos generados por nuevos productos o funcionalidades podría sobrecargar la infraestructura si no se controla adecuadamente la escalabilidad de los microservicios, afectando su capacidad de procesamiento.</p>

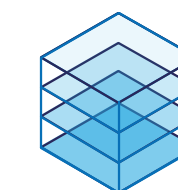
Análisis de los resultados obtenidos

- 1- Indique si la hipótesis de diseño pudo ser confirmada o no
- 2- En caso de que la hipótesis se haya confirmado, explique las decisiones de arquitectura que favorecieron el resultado
- 3- En caso de que los resultados del experimento no hayan sido favorables, explique por qué y cuáles cambios realizaría en el diseño

Sí, la hipótesis de diseño fue confirmada. El sistema demostró ser flexible y extensible, permitiendo la integración de nuevos productos sin afectar los microservicios existentes. El uso de **adaptadores, mediadores y eventos** para reducir el acoplamiento entre los servicios facilitó la incorporación de nuevas funcionalidades, manteniendo la estabilidad y escalabilidad de la arquitectura sin causar interrupciones en el servicio.

Las decisiones clave que favorecieron la confirmación de la hipótesis incluyen el uso de **microservicios modulares**, que permitieron la integración de nuevos productos sin modificar los servicios existentes, reduciendo el riesgo de errores. Además, la **implementación de adaptadores** facilitó la integración de nuevas fuentes de datos sin afectar la arquitectura principal. La **centralización de la lógica con mediadores** mejoró la flexibilidad del sistema, gestionando las interacciones sin generar dependencias directas. Finalmente, el **uso de eventos para comunicación asíncrona** redujo el acoplamiento entre microservicios, permitiendo una integración más flexible de nuevos productos sin generar dependencias rígidas.

Se obtuvieron los siguientes resultados:



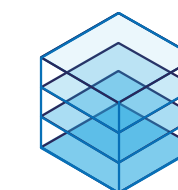
Evidencias

1- Presente evidencias de los resultados obtenidos en el experimento.

Para realizar las pruebas en el microservicio de suscripción se validaron varios planes mas de los que se encontraban diseñados, para agregarlos fue necesario agregar a nivel de base de datos la información de los planes que quedarían aprobados, lo cuál no requirio ningún cambio en el microservicio. Además, por el manejo de eventos gordos en una transacción larga entre los microservicios asociados se comparte la información del código del plan. Como se muestra en el llamado al BFF para la ejecución de la Saga.

```
{
  "id_suscripcion": "{{ $randomUUID }}",
  "cliente_codigo": "0001",
  "cliente_nombres": "Pablo Saga",
  "cliente_apellidos": "Perez Prieto",
  "cliente_usuario": "pperez",
  "cliente_rut": "1234567890",
  "cliente_cedula": "987654321",
  "cliente_email": "pperez@domain.com",
  "plan_codigo": "pro",
  "plan_nombre": "PRO"
}
```

	id	codigo	nombre
1	1	PRO	PRO
2	2	STANDARD	STANDARD
3	3	ENTERPRISE	ENTERPRISE
4	4	VIP	VIP
5	5	BASICO	BASICO



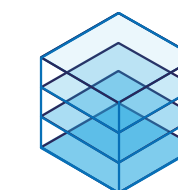
Evidencias

1- Presente evidencias de los resultados obtenidos en el experimento.

Además dentro del código de microservicio suscripciones vemos la entidad Plan, la cual es usada de la misma forma en el microservicio de servicio de datos. En cada microservicio se almacena la información correspondiente con el fin de no tener dependencias.

```
entidades.py X
microservicio-suscripciones > src > saludtechalpes > modulos > suscripciones > dominio > entidades.py > ...
19     email: Email = field(default_factory=Email)
20
21     @dataclass
22     class Plan(Entidad):
23         codigo: Codigo = field(default_factory=Codigo)
24         nombre: NombrePlan = field(default_factory=NombrePlan)
25
```

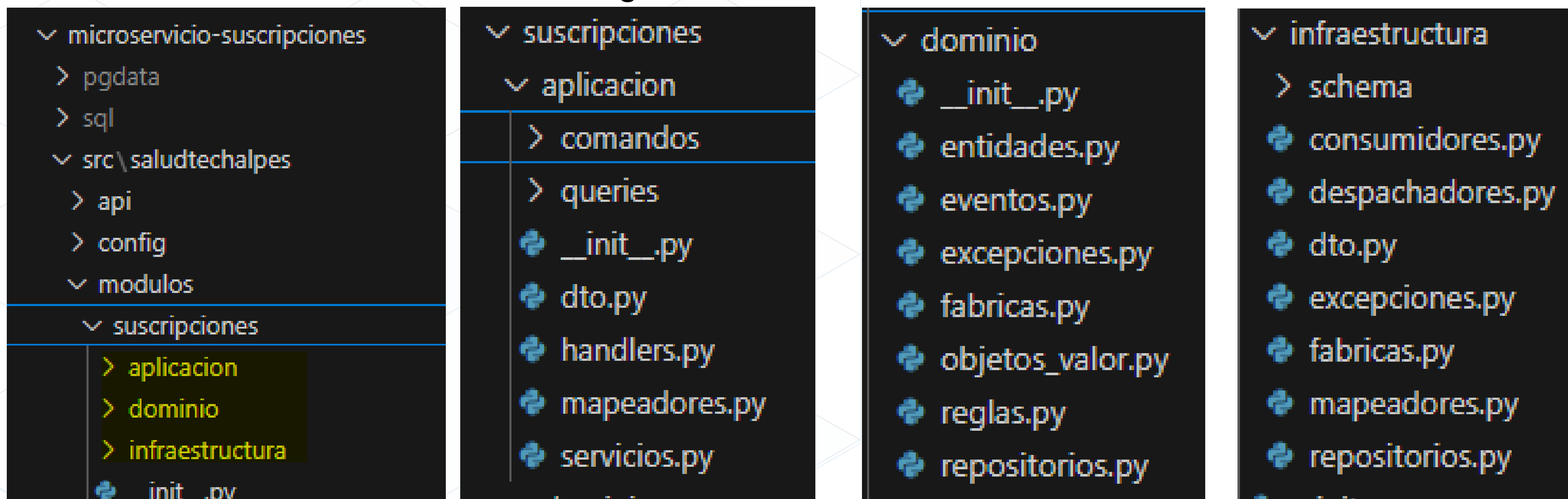
```
entidades.py X
microservicio-servicio-datos > src > saludtechalpes > modulos > servicioDatos > dominio > entidades.py > ...
17
18     @dataclass
19     class Plan(Entidad):
20         codigo: Codigo = field(default_factory=Codigo)
21         nombre: NombrePlan = field(default_factory=NombrePlan)
22
```



Evidencias

1- Presente evidencias de los resultados obtenidos en el experimento.

Además, gracias a la separación entre las capas del microservicio de Suscripciones y al uso de adaptadores, mediadores y eventos, es posible adaptarlo fácilmente a los cambios del negocio sin necesidad de modificaciones en el código.



© - **Derechos Reservados:** la presente obra, y en general todos sus contenidos, se encuentran protegidos por las normas internacionales y nacionales vigentes sobre propiedad Intelectual, por lo tanto su utilización parcial o total, reproducción, comunicación pública, transformación, distribución, alquiler, préstamo público e importación, total o parcial, en todo o en parte, en formato impreso o digital y en cualquier formato conocido o por conocer, se encuentran prohibidos, y solo serán lícitos en la medida en que se cuente con la autorización previa y expresa por escrito de la Universidad de los Andes.

De igual manera, la utilización de la imagen de las personas, docentes o estudiantes, sin su previa autorización está expresamente prohibida. En caso de incumplirse con lo mencionado, se procederá de conformidad con los reglamentos y políticas de la universidad, sin perjuicio de las demás acciones legales aplicables.
