

MISO

Maestría en Ingeniería de Software

Entrega 3: Grupo: ByteBros

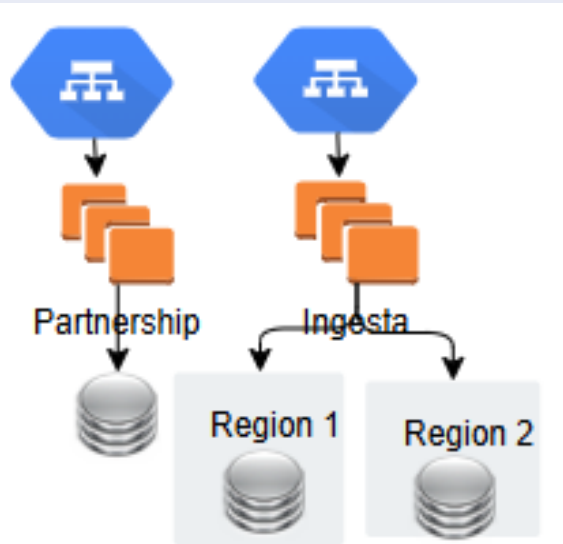
HUMBERTO ENRIQUE MAURY MAURY

MONICA ALEJANDRA MUÑOZ BELTRA

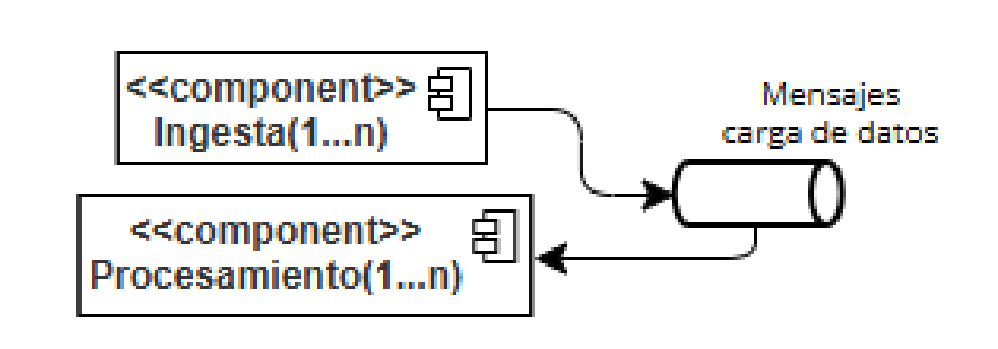
LUZ STELLA OCHOA MARIN

ANDRES FELIPE LOMBO CAICEDO

Atributo de calidad 1: Escalabilidad

Escenario de calidad: Escalado por el aumento del 5% de nuevos centros de salud			
Escenario #: 1	El sistema debe escalar eficientemente, soportando la integración del 5% adicional de nuevos centros de salud (30.000 centros), sin degradar el servicio.		
Fuente	Aumento del 5% del mercado de centros de salud		
Estímulo	Aumento en la cantidad de datos procesados en inscripción de partnership e ingesta		
Ambiente	Producción.		
Artefacto	Microservicios de partnership e ingesta		
Respuesta	Capacidad de soportar la carga sin degradar el servicio.		
Medida de la respuesta	Tiempo en línea del sistema (uptime) del 99% durante la expansión. El proceso de ingesta de datos debe procesar cada lote de 10000 registros en menos de 5 minutos.		
Decisiones Arquitecturales	Punto de sensibilidad	Tradeoff	Riesgo
Uso de base de datos particionada por región	Sincronización entre bases de datos regionales evitando duplicidad y garantizar consistencia de datos.	Costos elevados por implementar replicación geográfica.	Latencia muy alta en la experiencia del usuario.
Despliegue con balanceo de carga global.	Latencia entre distintas zonas geográficas.	Complejidad de implementación multirregión.	Existe el riesgo de que los datos entre distintas regiones se desincronicen.
Justificación	El despliegue de infraestructura distribuida en múltiples regiones con balanceadores de carga globales y bases de datos regionales busca mejorar la escalabilidad del sistema, asegurando que los microservicios afectados puedan procesar los datos sin interrupciones. Además, busca reducir la latencia para mejorar la experiencia de los usuarios en diferentes zonas geográficas. Al particionar las bases de datos por región permite mejorar el rendimiento a cada área. Aunque puede implicar mayores costos y complejidad en el código.		
Diagrama de arquitectura			

Atributo de calidad 1: Escalabilidad

Escenario de calidad: Carga masiva de datos en la ingesta			
Escenario #: 2	El sistema debe escalar automáticamente para manejar ingestas masivas con un volumen de hasta 4TB, siendo capas de procesar todos los datos sin caídas del sistema.		
Fuente	Proveedor de datos		
Estímulo	Ingesta de 4TB de datos.		
Ambiente	Producción con simulación de carga.		
Artefacto	Microservicio de ingesta y procesamiento.		
Respuesta	Procesar eficientemente los datos garantizando disponibilidad del 99,9%.		
Medida de la respuesta	Tiempos de procesamiento menores a 20 minutos para 4TB.		
Decisiones Arquitecturales	Punto de sensibilidad	Tradeoff	Riesgo
Uso de colas de mensajería para desacoplar ingesta y procesamiento.	Latencia en la cola de mensajería.	Complejidad para monitorear múltiples instancias.	Latencia en la ingesta si no se configura un sistema de prioridad de mensajes.
Auto escalado de nodos de procesamiento en función de la carga	Capacidad de nodos para manejar carga de trabajo.	Costos elevados por escalar la infraestructura.	Error en el escalado puede provocar sobrecarga innecesaria o subutilización de recursos.
Justificación	Al manejar volúmenes masivos de datos, se requiere usar una arquitectura escalable basada en eventos para desacoplar la ingesta y el procesamiento.		
Diagrama de arquitectura			

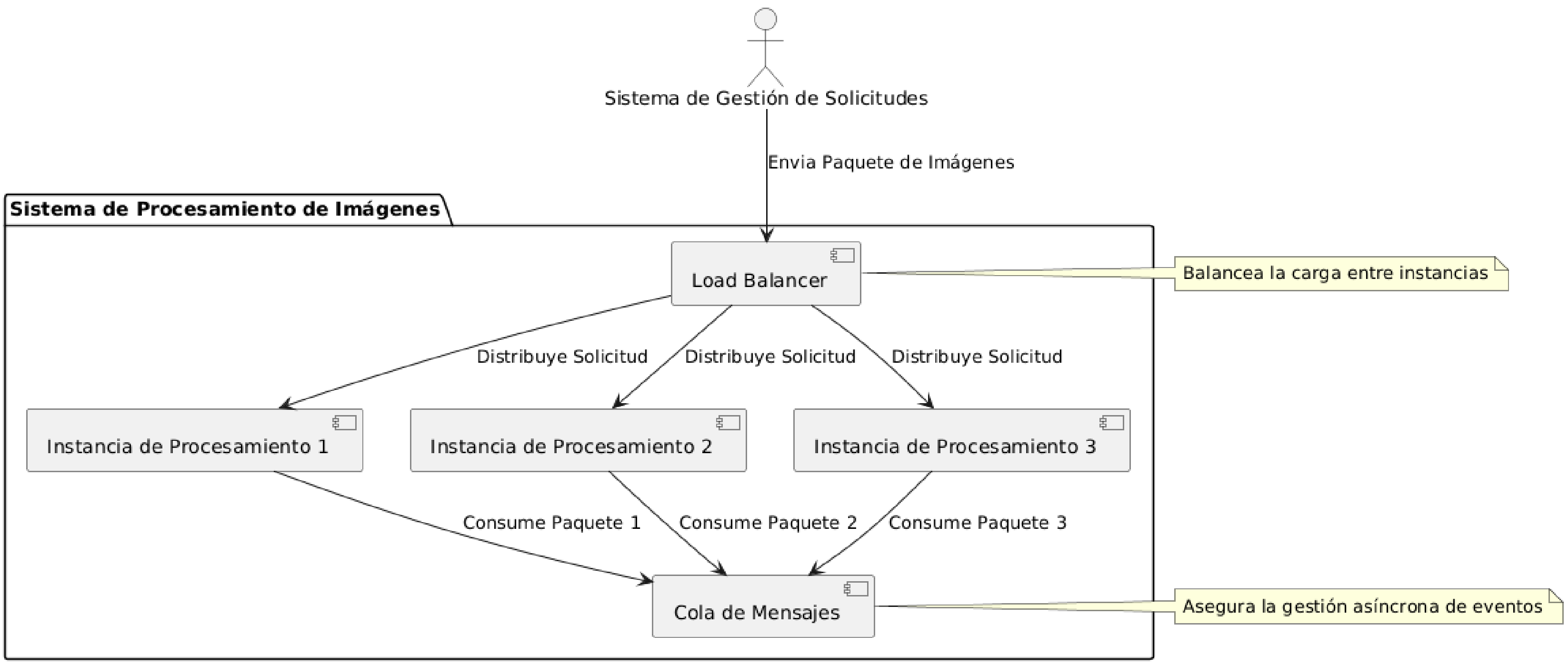
Atributo de calidad 1: Escalabilidad

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



Atributo de calidad 2: Seguridad

Escenario de calidad: Acceso a la plataforma de manera segura.			
Escenario #: 4	Busca garantizar que solo los usuarios autenticados puedan acceder a los datos anonimizados, utilizando mecanismos de seguridad que eviten accesos no autorizados.		
Fuente	Clientes accediendo al sistema de datos anonimizados		
Estímulo	Consultar datos con el servicio STA Pro,		
Ambiente	Producción		
Artefacto	Servicio de ingesta.		
Respuesta	Acceso permitido solo a usuarios autenticados.		
Medida de la respuesta	Porcentaje de accesos no autorizados detectados del 0%.		
Decisiones Arquitecturales	Punto de sensibilidad	Tradeoff	Riesgo
Token JWT para autenticación de usuarios.	Protección de datos sensibles.	Gestión de tokens seguros.	Fallos de autenticación y acceso a datos sensibles.
Autenticación OAuth 2.0	Administración de sesiones.	Gestión de claves seguras y encriptadas.	Robo de tokens de autenticación.
		Latencia al momento de autenticarse.	
Justificación	Probar la seguridad del acceso a la plataforma, implementando autenticación OAuth 2.0 o tokens JWT, asegurando que solo los usuarios con los permisos necesarios puedan acceder a la información.		
Diagrama de arquitectura	<pre>graph LR; AG[API Gateway] --> I[Ingesta]; I --> DB[(Database)]; I <--> Auth([Autenticación OAuth 2.0 o JWT]);</pre>		

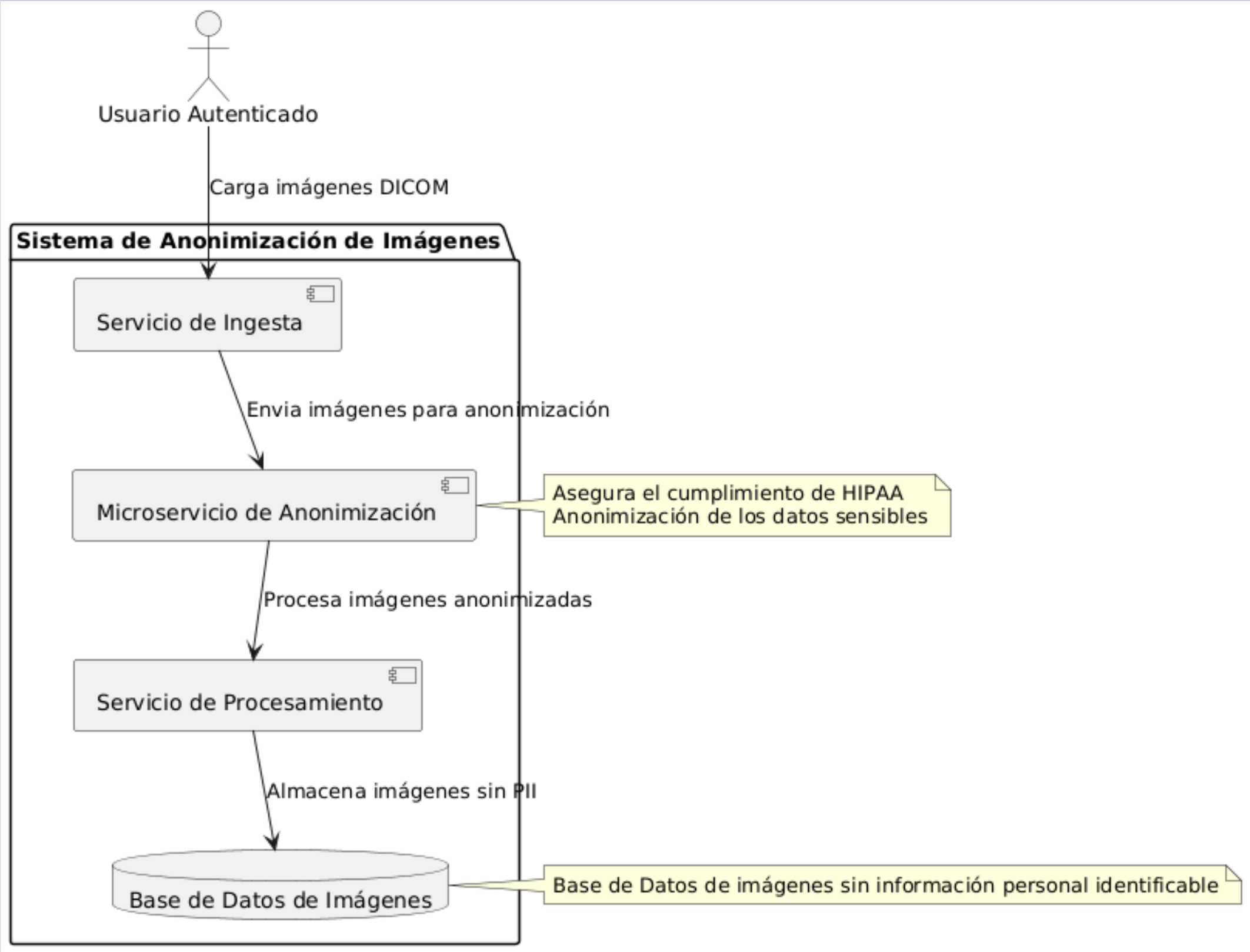
Atributo de calidad 2: Seguridad

Escenario de calidad: Protección de los datos ingestados.			
Escenario #: 5	Valida la anonimización de los datos por parte del sistema, asegurando que ninguna imagen contenga Información personal identificable, evitando que se procesen y almacenen sin protección, asegurando el cumplimiento con regulaciones de privacidad (HIPAA).		
Fuente	Usuario autenticado cargando imágenes.		
Estímulo	Carga de imágenes DICOM con metadatos sensibles.		
Ambiente	Producción.		
Artefacto	Servicio de ingesta y procesamiento.		
Respuesta	Los datos se deben almacenar sin información personal identificable.		
Medida de la respuesta	El 100% de las imágenes cargadas no contienen información personal identificable.		
Decisiones Arquitecturales	Punto de sensibilidad	Tradeoff	Riesgo
Arquitectura basada en eventos para aislar microservicio de ingesta y de procesamiento.	Ejecución correcta de los scripts de anonimización.	Mayor tiempo de procesamiento vs seguridad de los datos.	Fallas en el servicio de anonimización podría exponer datos sensibles.
Uso de un microservicio de procesamiento para la anonimización de datos.	Tiempo de procesamiento frente a la carga de datos.	Complejidad en la implementación.	Alta latencia en la carga de datos.
Justificación	Debido a que la carga de imágenes debe procesar los datos sensibles y anonimizarlos, para cumplir con la normativa HIPAA, se debe garantizar la privacidad, implementando un servicio de anonimización, utilizando una arquitectura basada en eventos que permite procesar grandes volúmenes de datos garantizando la seguridad del sistema.		

Atributo de calidad 2: Seguridad

Escenario de calidad: Protección de los datos ingestados.

Diagrama de arquitectura



Atributo de calidad 2: Seguridad

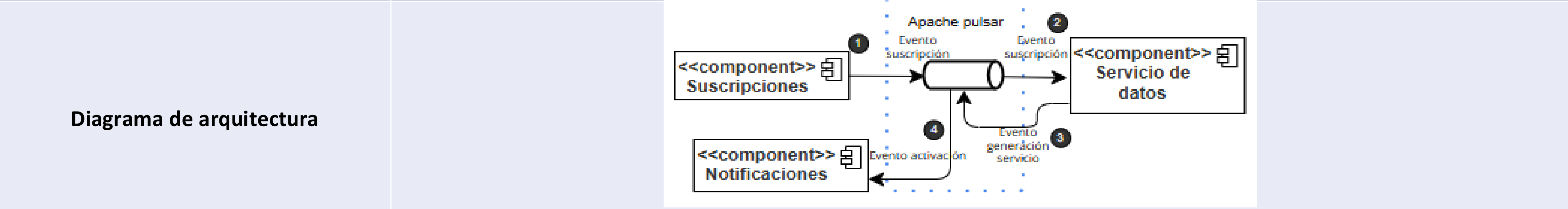
Escenario de calidad: Integridad y seguridad de los datos

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 sin inconsistencias 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



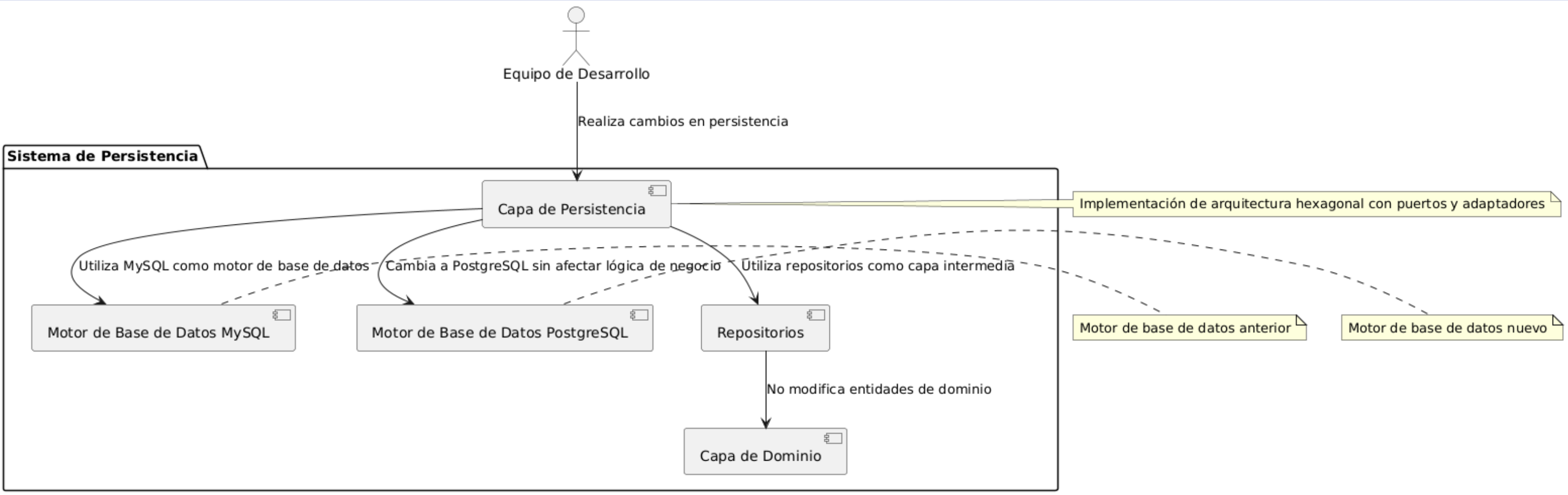
Atributo de calidad 3: Modificabilidad

Escenario de calidad: Cambio en la tecnología de la base de datos sin afectar la capa de dominio.			
Escenario #: 7	Probar la flexibilidad de la arquitectura, permitiendo realizar cambios en el motor de base de datos sin requerir modificaciones en lógicas de negocio ni en entidades de dominio.		
Fuente	Equipo de desarrollo.		
Estímulo	Cambio del motor de base de datos de MySQL a PostresSQL.		
Ambiente	Desarrollo.		
Artefacto	Capa de persistencia.		
Respuesta	Cambios realizados en las bases de datos sin modificar en la capa de dominio.		
Medida de la respuesta	0% de cambios en entidades de dominio al implementar cambios en las bases de datos.		
Decisiones Arquitecturales	Punto de sensibilidad	Tradeoff	Riesgo
Uso de repositorios como capa intermedia.	Mantenibilidad de cara a integración de múltiples tecnologías.	Complejidad en pruebas de integración.	Cambios sin documentar pueden presentar complicaciones futuras.
Arquitectura hexagonal.	Implementación de adaptadores de persistencia.	Complejidad en implementación de código.	Fallas en adaptadores.
Justificación	Implementar una arquitectura hexagonal con puertos y adaptadores, utilizando patrones de repositorios para desacoplar la capa de dominio de la infraestructura de persistencia nos permite que a la hora de modificar la capa de persistencia no afectará las entidades de dominio.		

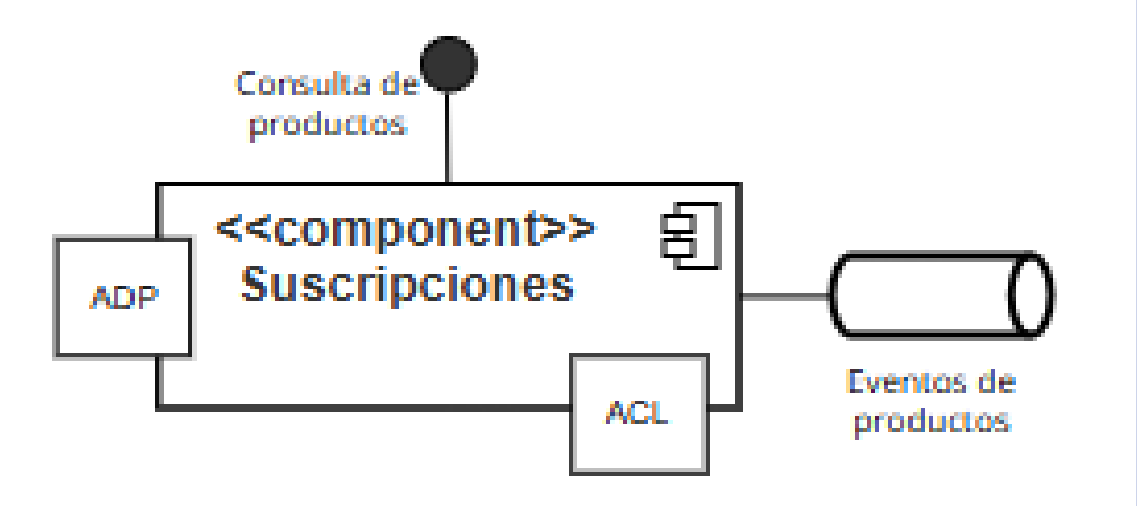
Atributo de calidad 3: Modificabilidad

Escenario de calidad: Cambio en la tecnología de la base de datos sin afectar la capa de dominio.

Diagrama de arquitectura



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
Tactica 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 succripción, 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.		
Diagrama de arquitectura			

Atributo de calidad 3: Modificabilidad

Escenario de calidad: Modificabilidad en la Integración de un Microservicio de Notificaciones

Escenario #: 9	Este escenario evalúa la capacidad del sistema para incorporar un nuevo microservicio de notificaciones por correo electrónico de manera eficiente, sin afectar el rendimiento ni la estabilidad de las funcionalidades existentes. La integración del microservicio debe permitir que el sistema gestione notificaciones de manera autónoma y escalable, utilizando la arquitectura de microservicios y el aislamiento del dominio. La táctica de incrementar la cohesión asegurará que el nuevo microservicio se integre de manera coherente y eficaz, sin requerir cambios masivos en otras partes del sistema. El objetivo es que la integración no tome más de dos semanas, desde el desarrollo hasta su puesta en producción.		
Fuente	Equipo de desarrollo y mantenimiento de STA		
Estímulo	La necesidad de agregar un nuevo microservicio para gestionar las notificaciones por correo electrónico, para cumplir con los requisitos de comunicación del sistema, como alertas y actualizaciones a clientes o partnership		
Ambiente	El escenario se desarrolla dentro de un sistema de microservicios distribuido, que ya está en producción. El nuevo microservicio de notificaciones debe ser integrado de manera aislada, sin interrumpir el flujo de trabajo actual.		
Artefacto	Sistema de microservicios distribuido, con el nuevo microservicio de notificaciones por correo electrónico totalmente integrado al sistema principal.		
Respuesta	El sistema debe ser capaz de integrar el nuevo microservicio de notificaciones sin que se vea afectada la operación de otros microservicios. Este microservicio debe ser completamente autónomo, gestionando las notificaciones sin intervención de otros servicios. La integración debe realizarse dentro del tiempo estipulado de dos semanas, asegurando que el microservicio sea funcional y escalable desde su primera versión.		
Medida de la respuesta	El microservicio debe ser integrado y funcional dentro de dos semanas, con una tasa de éxito de al menos 95% en la entrega de notificaciones en los primeros días de producción. Además, el tiempo de respuesta de la integración no debe exceder 2 semanas, y debe garantizarse que el sistema principal no experimente tiempos de inactividad durante la integración.		
Decisiones Arquitecturales	Punto de sensibilidad	Tradeoff	Riesgo
Patrón de Microservicios	Integración con Otros Microservicios	Autonomía vs. Complejidad en su integración inicial	Dependencias Cruzadas
Aislamiento del Dominio	Coherencia en la Gestión de Notificaciones	Tiempo de Desarrollo vs. Integración Rápida	Rendimiento en Escalabilidad
Cohesión			Falta de Pruebas Exhaustivas
Justificación	La arquitectura de microservicios permite desarrollar, probar y desplegar el microservicio de notificaciones de forma independiente, sin que esto afecte al sistema central. Además, el aislamiento del dominio asegura que el nuevo microservicio no interrumpa las funcionalidades de otros microservicios del sistema, incrementando la cohesión y reduciendo los riesgos de fallos en producción.		

Atributo de calidad 3: Modificabilidad

Escenario de calidad: Modificabilidad en la Integración de un Microservicio de Notificaciones

Diagrama de arquitectura

