



Universidad Nacional Mayor de San Marcos

Universidad del Perú - Decana de América

Área de Ingeniería

Asignatura: Diseño de Sistemas de Información

Título: Documento Arquitectura - Modelo 4 + 1

Docente: ANDRADE MOGOLLON TEODORO MANUEL

Integrantes:

- Callán Quispe Michael Julio
- Javier Tarapaqui jeanfranco
- Mendoza Segura Sebastián
- Meza Lara Renato Matías
- Sandoval Castillo Robert Alonso
- Vasquez Arias Eduardo Alexis

1. Introducción:

1.1. Propósito del documento:

El propósito de este documento es definir las especificaciones técnicas, funcionales y estructurales del módulo de **Gestión de Solicitudes**. Este informe sirve como guía para los desarrolladores, analistas de calidad y personal de soporte, detallando cómo se debe comportar el sistema desde la creación del requerimiento hasta su seguimiento histórico, asegurando que se cumplan las reglas de negocio y los estándares de integridad documental.

1.2. Alcance (arquitectura del sistema y detalle del módulo del grupo):

El alcance del presente módulo se limita al **Front-to-Back** del proceso de solicitud. Incluye la interfaz de usuario para la selección de servicios, la lógica de validación de documentos obligatorios, el almacenamiento en base de datos de los metadatos de la solicitud y el motor de consulta de estados en tiempo real.

- **Fuera de alcance:** Los flujos de aprobación jerárquica, la ejecución técnica del servicio solicitado y las integraciones con pasarelas de pago (estos corresponden a otros módulos del ecosistema Campus360).

1.3. Público objetivo:

- **Equipo de Desarrollo:** Para la implementación de la lógica y validaciones.
- **Arquitectos de Software:** Para validar la integración con el núcleo de la plataforma.
- **Personal de Soporte TI:** Como referencia para la resolución de incidencias en la carga de documentos.
- **Administradores Académicos:** Para comprender la capacidad de generación de reportes y cumplimiento de SLA.

1.4. Tecnologías principales:

Para garantizar la escalabilidad y el rendimiento descritos en los objetivos, se sugiere el uso de:

- **Frontend:** React o Angular (para una interfaz reactiva de seguimiento en tiempo real).
- **Backend:** Spring Boot o Node.js (para la gestión eficiente de validaciones y adjuntos).
- **Base de Datos:** PostgreSQL u Oracle SQL (para garantizar integridad referencial y manejo de historial).
- **Almacenamiento de archivos:** AWS S3 o Azure Blob Storage (para la gestión de documentos adjuntos).

2. Contexto y visión general del módulo


2.1. Descripción general del sistema

Campus360 es una plataforma de gestión universitaria integral diseñada para centralizar trámites administrativos y académicos. El sistema opera bajo una arquitectura modular donde el núcleo administrativo se comunica con módulos especializados para evitar la redundancia de datos y reducir los tiempos de respuesta operativos.

2.2. Principales actores del módulo

- Solicitante (Estudiante/Docente): Inicia trámites, carga requisitos y monitorea el avance.
- Personal Administrativo: Supervisa la carga operativa y analiza el cumplimiento de los tiempos de respuesta (SLA).
- Soporte TI: Monitorea la salud del módulo y resuelve problemas técnicos con el cargue de archivos o persistencia de datos.
- Aprobador: Toma decisiones con respecto a las solicitudes registradas.

2.3. Restricciones y supuestos

 (Opcional: diagrama de contexto)

3. Vista de casos de uso

3.1. Actores del sistema

Solicitante: Usuario responsable de iniciar y gestionar sus solicitudes dentro del sistema. Puede registrar nuevas solicitudes, adjuntar la documentación requerida, consultar el estado y el historial de sus trámites, así como realizar acciones permitidas sobre sus solicitudes, como la anulación, según las reglas establecidas.

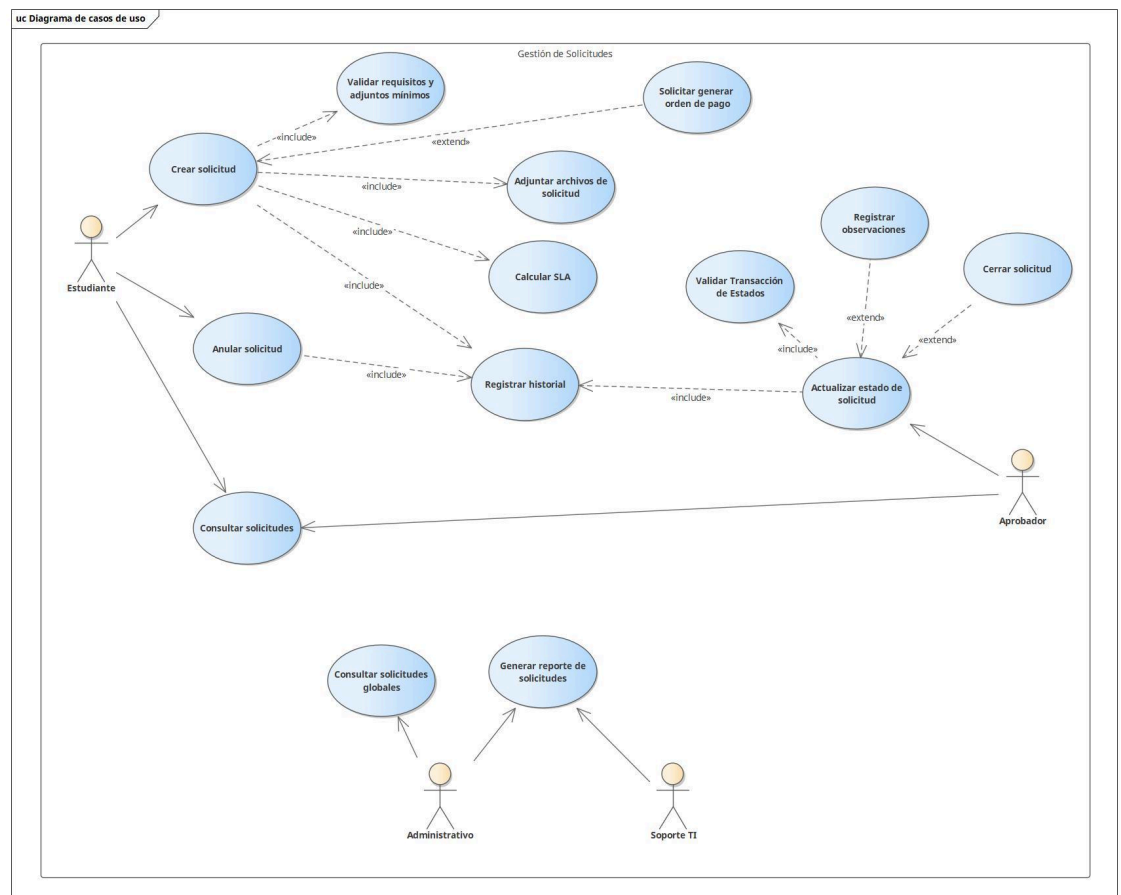
Personal administrativo: Actor encargado de la supervisión operativa del módulo. Tiene acceso a la consulta global de solicitudes y a la generación de reportes para fines de control, seguimiento y toma de decisiones administrativas, incluyendo la visualización del cumplimiento de los niveles de servicio (SLA).

Soporte TI: Actor responsable de brindar soporte técnico al sistema. Atiende incidencias relacionadas con errores de funcionamiento, problemas en el registro de solicitudes o carga de adjuntos, y realiza análisis mediante la consulta y generación de reportes para garantizar la correcta operatividad del módulo.

Aprobador: es el actor responsable de gestionar y tomar decisiones sobre las solicitudes registradas en el sistema. Interactúa directamente con el módulo de Gestión de Solicitudes para revisar, evaluar y actualizar el estado de cada trámite.

Puede aprobar, observar o rechazar solicitudes según las reglas establecidas, registrar observaciones o comentarios, y cerrar solicitudes una vez completado su flujo.

3.2. Diagrama de casos de uso



3.3. Relación con las vistas de arquitectura

3.1. Relación con la Vista Lógica

Los casos de uso definen el comportamiento que deben implementar las clases y componentes del sistema.

- Conexión: El caso de uso CU-01 (Crear Solicitud) justifica la existencia de entidades como Solicitud, Documento y HistorialEstado.

- Impacto: La lógica de "Validación de Requisitos" descrita en el escenario se traduce en la capa de servicios mediante el componente ValidadorDeRequisitos, asegurando que las reglas de negocio se apliquen antes de la persistencia de datos.

3.2. Relación con la Vista de Proceso

Esta relación describe cómo los escenarios gestionan el flujo de información y la integridad de los datos en tiempo real.

- Conexión: El caso de uso CU-02 (Consultar Registro) se relaciona directamente con el diagrama de secuencia de consulta, donde se observa la interacción entre el SolicitudController y el SolicitudService.
- Impacto: El requerimiento de visualizar el estado del trámite justifica la implementación del MotorDeCalculo_SLA, el cual procesa dinámicamente los tiempos de respuesta cada vez que se solicita el detalle de una solicitud.

3.3. Relación con la Vista de Desarrollo

Define cómo se organiza el código fuente y los módulos del proyecto para dar soporte a las funcionalidades.

- Conexión: Los casos de uso de gestión y consulta se agrupan en el componente de software Gestion_citas.jar (o similar según el diagrama de componentes).
- Impacto: La necesidad de separar las funciones del Solicitante de las del Administrativo y Aprobador organiza el proyecto en paquetes específicos dentro de Spring Boot, facilitando la mantenibilidad y la escalabilidad del módulo.

3.4. Relación con la Vista de Despliegue

Determina la distribución física de los componentes para asegurar que el usuario pueda ejecutar los casos de uso.

- Conexión: El flujo de "Adjuntar archivos" en el registro de solicitud requiere un nodo de almacenamiento (Servidor de Archivos/Base de Datos) persistente.
- Impacto: La interacción entre el Servidor Frontend (donde el usuario inicia el caso de uso) y el Servidor Backend (donde reside la lógica) a través de protocolos HTTPS garantiza que los escenarios se ejecuten de forma segura y distribuida, tal como se plantea en el diagrama de despliegue.

4. Vista Lógica

4.1. Arquitectura en capas

- Capa de presentación
 - Interfaz entre el usuario y el sistema.
 - Recibe las solicitudes, valida datos a nivel de formato y envía la información a la capa de aplicación.
 - Muestra las respuestas al usuario.
- Capa de aplicación / servicios
 - Implementa los casos de uso del sistema.
 - Coordina el flujo de la aplicación, invoca entidades del dominio y utiliza repositorios para acceder a los datos.
- Capa de dominio
 - Contiene la lógica de negocio central.
 - Modela el problema mediante entidades, reglas de negocio y validaciones propias del dominio.
- Capa de persistencia (solo conceptual, sin tecnología)
 - Define cómo se almacenan y recuperan los datos del dominio.
 - Implementa los contratos de repositorio sin detallar tecnologías específicas.

4.2. Módulos y paquetes

- Descripción de los módulos del sistema
- Relación entre módulos
- Módulo desarrollado por el grupo (detalle)

4.3. Modelo lógico (clases de dominio)

- Entidades principales

Solicitud: Entidad central que gestiona el ciclo de vida de una solicitud (estado, SLA, adjuntos e historial).

SolicitudTrámite: Especialización de Solicitud asociada a un trámite del catálogo.

SolicitudServicio: Especialización de Solicitud asociada a un servicio del catálogo.

Usuario: Representa a los actores del sistema que interactúan con las solicitudes según su rol.

Trámite: Define los requisitos, documentos obligatorios y SLA de cada tipo de trámite.

Servicio: Define reglas, condiciones y SLA de cada tipo de servicio.

Adjunto: Archivo o documento cargado como evidencia de una solicitud.

HistorialEstado: Registro de auditoría que almacena los cambios de estado de una solicitud.

- Servicios de dominio

GestiónSolicitudService: crea, valida, anula y cambia estados de solicitudes.

ValidaciónRequisitosService: verifica adjuntos y requisitos mínimos según trámite/servicio.

SLAService: calcula vencimientos y cumplimiento del SLA.

ReporteService: genera consultas y reportes de solicitudes para personal administrativo.

- Relaciones relevantes

Herencia: SolicitudTrámite y SolicitudServicio heredan de Solicitud.

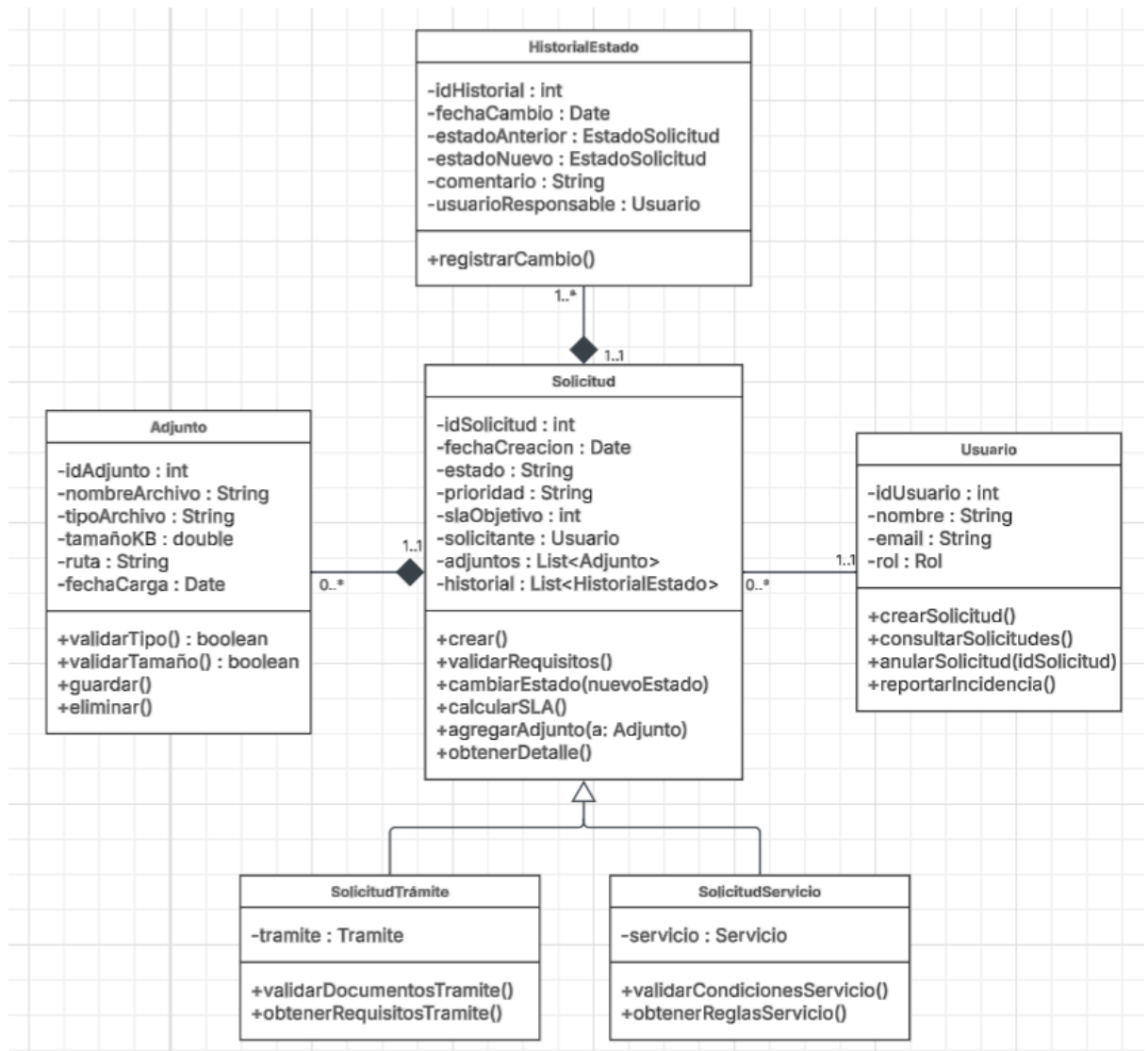
Composición: Solicitud contiene Adjunto y HistorialEstado.

Asociación:

- Solicitud – Usuario
- SolicitudTrámite – Trámite
- SolicitudServicio – Servicio

Diagramas

- Diagrama de paquetes
- Diagrama de clases (alto nivel, dominio)



5. Vista de Procesos

5.1. Procesos clave

- El módulo de Gestión de Solicitudes se sustenta en un conjunto de procesos dinámicos que coordinan la interacción entre la interfaz de usuario, la capa de control

y los servicios de aplicación, estos procesos permiten ejecutar los casos de uso principales del sistema, asegurando que las solicitudes sean registradas, consultadas y anuladas de manera consistente, respetando las reglas de negocio definidas.

Los procesos clave identificados en el módulo son:

- Creación de Solicitud con validación de requisitos.
- Consulta de solicitudes y visualización del detalle.
- Anulación de solicitudes según reglas de negocio.

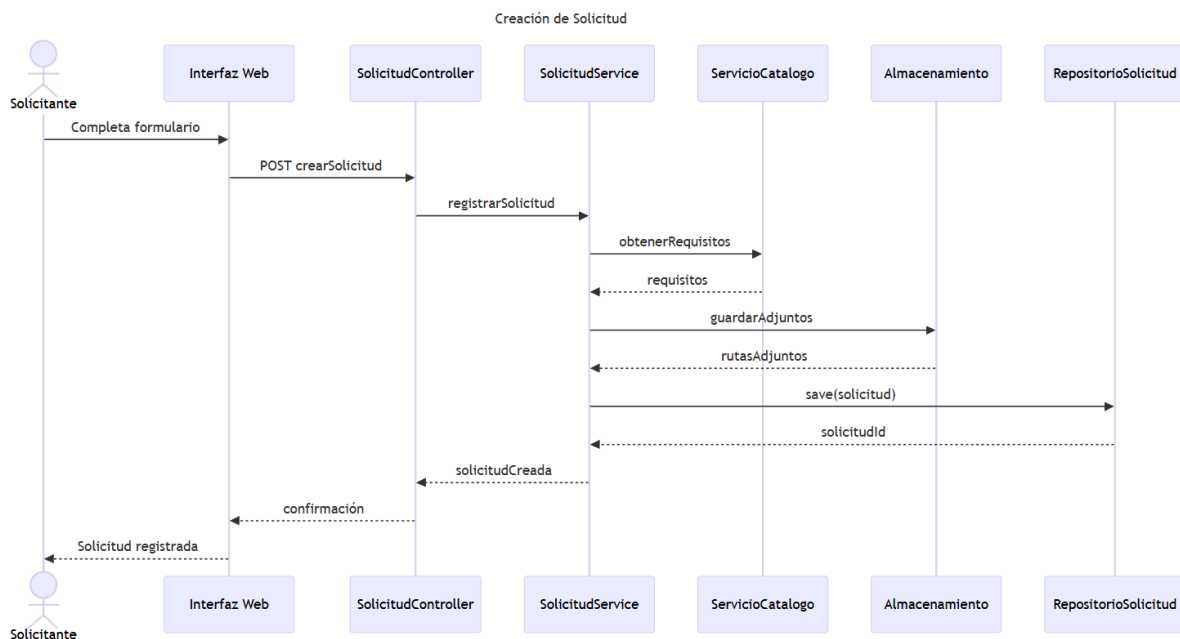
Descripción general de los flujos principales:

Los flujos del sistema se inician cuando el solicitante interactúa con la interfaz web para registrar, consultar o anular una solicitud. La interfaz envía las peticiones a la capa de control, la cual coordina la ejecución del proceso mediante los servicios de aplicación correspondientes.

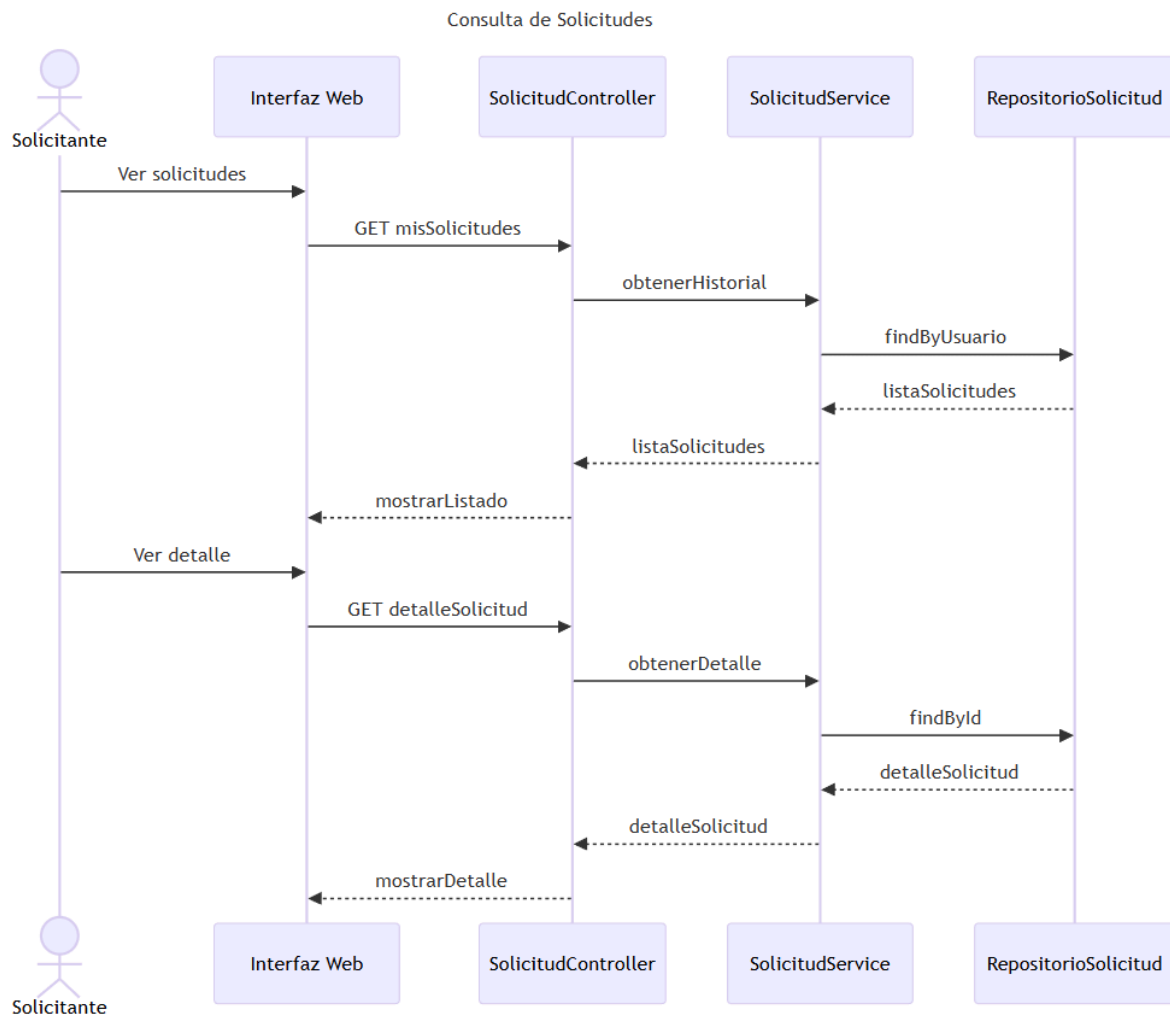
Durante el proceso de creación de una solicitud, el sistema valida que los datos ingresados y los documentos adjuntos cumplan con los requisitos establecidos para el trámite o servicio seleccionado antes de permitir su registro. En los procesos de consulta y anulación, el sistema verifica el estado actual de la solicitud y las reglas de negocio aplicables antes de retornar la información solicitada o actualizar el ciclo de vida de la solicitud. Todos los flujos garantizan la integridad de la información y el mantenimiento del historial de estados.

5.2. Diagramas de secuencia

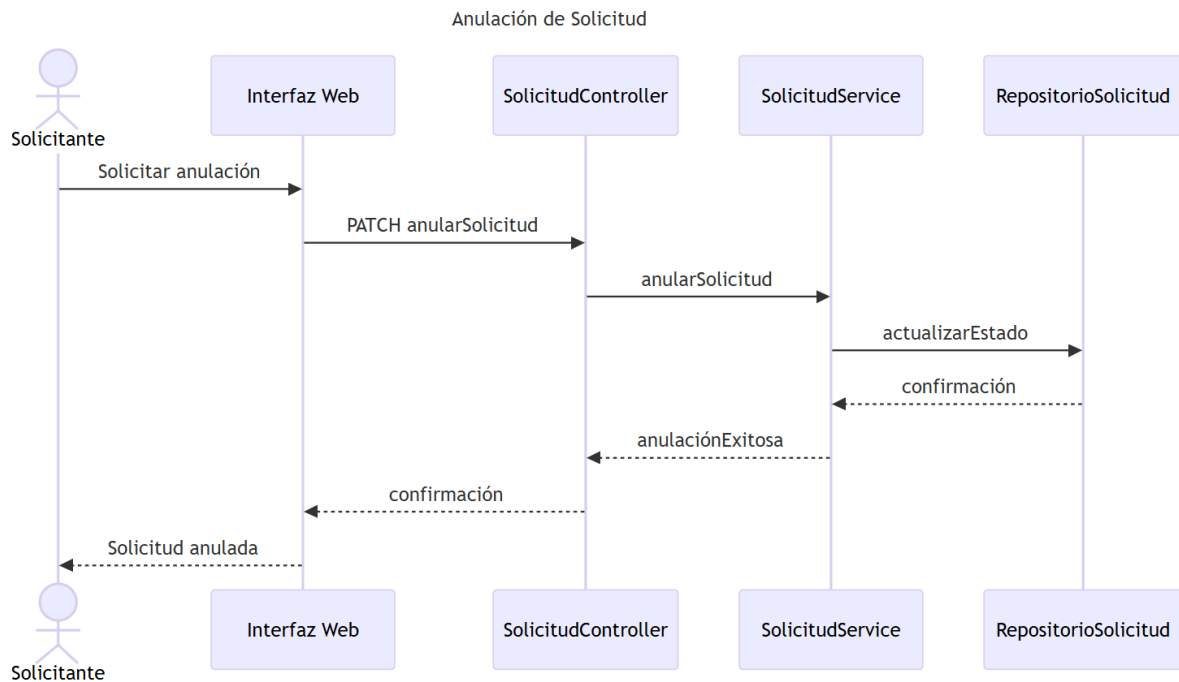
- Secuencia 1: Creación de Solicitud con validación de requisitos.



- Secuencia 2: Consulta de solicitudes y visualización del detalle.



- Secuencia 3: Anulación de solicitudes según reglas de negocio.



6. Vista de Desarrollo

6.1. Componentes del sistema

- Frontend: Interfaz de usuario (React/HTML) que reside en el servidor frontend. Captura datos y los presenta mediante el artefacto `solicitud.html`, comunicándose con el backend vía **API REST**.
- Backend: Núcleo del sistema empaquetado en `Gestion_solicitudes.jar`. Incluye:
 - **SolicitudController:** Lógica de control y exposición de endpoints.
 - **SolicitudService:** Lógica de negocio y orquestación de servicios.
 - **Domain:** Definición de entidades y reglas del dominio.
 - **Sistema de almacenamiento:** Gestión de archivos mediante la interfaz `GuardarAdjuntos`.
 - **Repositorio Solicitud:** Componente de persistencia que gestiona el acceso a la **Database**.

6.2. Dependencias entre componentes

- Comunicación entre servicios:
 - **Frontend - Backend:** La comunicación se realiza mediante peticiones RESTful sobre HTTPS, permitiendo una interacción fluida y no bloqueante en el Frontend.
 - **Backend - Base de Datos:** El **RepositorioSolicitud** gestiona la persistencia de datos hacia el `tablespace_solicitudes` en Oracle utilizando el protocolo **JDBC** (Java Database Connectivity).

- Integración con módulos externos:

1. Fase de Validación y Registro Inicial (Orquestación Síncrona)

Al recibir una petición del **Portal Integrador (G10)**, **Gestión de Solicitudes (G3)** valida la sesión con **Identidad y Accesos (G1)**, consulta requisitos en el **Catálogo de Servicios (G2)** y verifica disponibilidad en **Reservas de Ambientes (G6)**. Antes de guardar, registra el intento en **Auditoría y Monitoreo (G9)** y solicita a **Pagos (G5)** la creación de una orden de cobro.

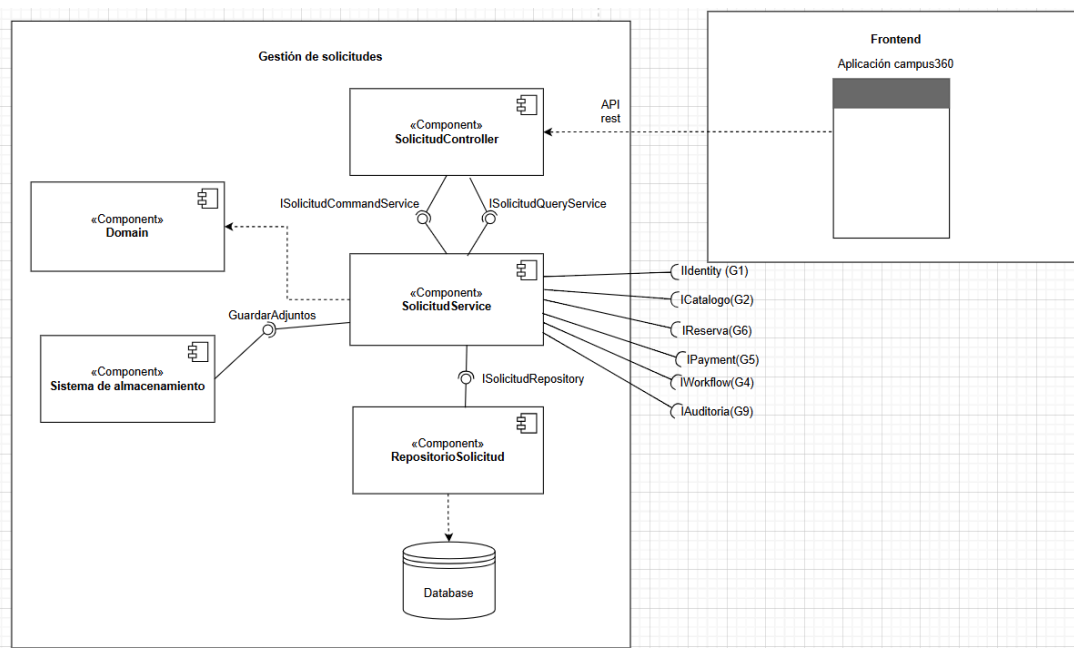
2. Fase de Activación y Transición al Flujo de Trabajo (Callback Asíncrono)

Una vez que el usuario paga, **Pagos (G5)** notifica a **Gestión de Solicitudes (G3)** mediante una interfaz expuesta. **G3** actualiza el estado, registra el hito en **Auditoría y Monitoreo (G9)** e inicia formalmente el flujo de revisión en **Flujo de Aprobaciones (G4)**.

3. Monitoreo de Desempeño y Cierre de Ciclo

Flujo de Aprobaciones (G4) consulta detalles a **G3** (vía Pull) y le devuelve el resultado final (Aprobado/Rechazado). Finalmente, el módulo de **Reportes y KPIs (G8)** extrae datos de **G3** para medir tiempos de respuesta (TAT) y cumplimiento de niveles de servicio (SLA).

6.3. Diagrama de componentes



7. Vista Física (Despliegue)

7.1. Entorno de despliegue

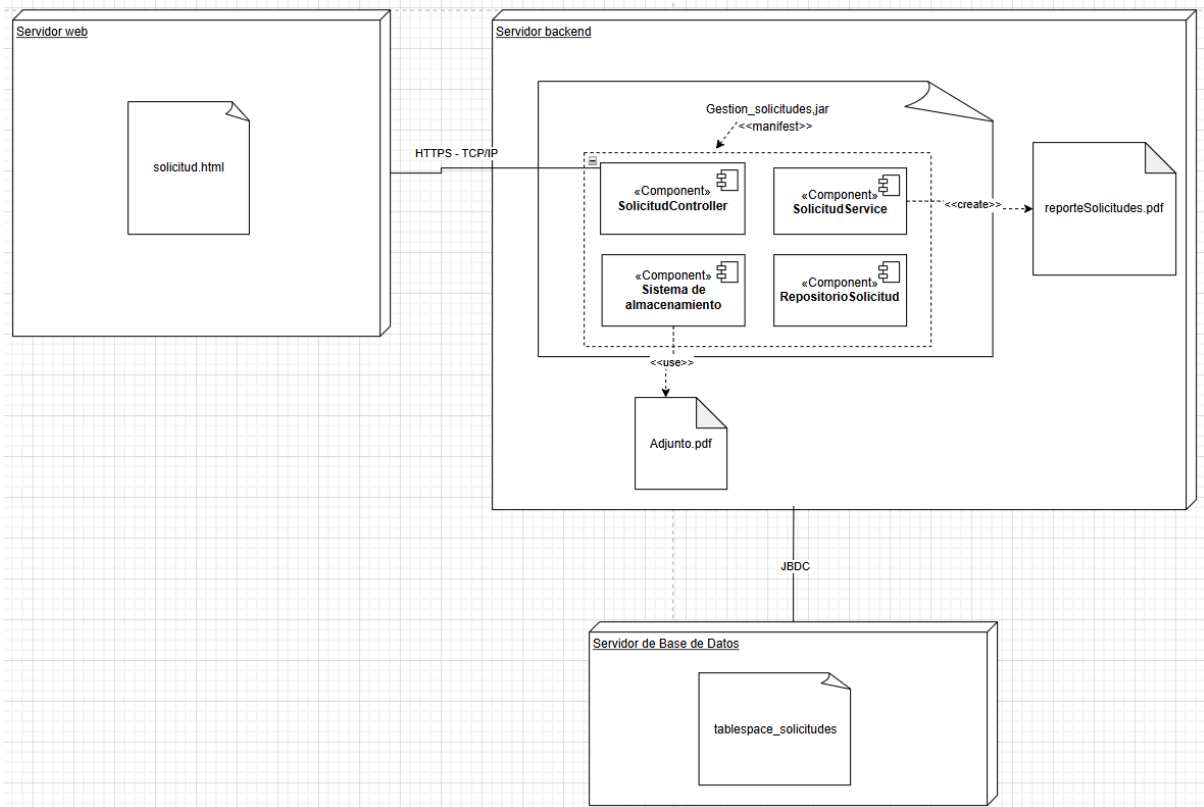
- Despliegue local:
Para el desarrollo y pruebas del módulo de **Gestión de Solicitudes**, se utiliza un entorno de despliegue local que simula la arquitectura productiva. Esto permite validar la interoperabilidad entre los artefactos físicos y la persistencia de datos bajo condiciones controladas.
- Nodos principales:
 - **Servidor Frontend (Host de Interfaz):** Nodo encargado de la entrega de recursos estáticos. Aloja el artefacto **solicitud.html**, el cual captura las peticiones del usuario y las redirige al backend mediante el protocolo **HTTPS (TCP/IP)**.
 - **Servidor Backend (Entorno de Ejecución):** Nodo que ejecuta la máquina virtual de Java (JVM). Aloja el artefacto **Gestion_solicitudes.jar**, el cual manifiesta los componentes lógicos de **SolicitudController**, **SolicitudService**, **Domain**, **Sistema de almacenamiento** y **Repositorio Solicitud**. Este nodo también gestiona el almacenamiento local de archivos externos como **Adjunto.pdf** y la generación de **reporteSolicitudes.pdf**.
 - **Servidor de Base de Datos (Persistencia):** Nodo encargado de la gestión de datos en el **tablespace_solicitudes** de Oracle. Se comunica con el backend mediante el protocolo **JDBC**, asegurando la integridad de las transacciones y el almacenamiento de las referencias (rutas) de los archivos adjuntos.

7.2. Puertos y comunicaciones

- Puertos utilizados:
 - **Puerto 443 (HTTPS):** Puerto principal de entrada para el **Servidor Frontend**, asegurando que la navegación del usuario esté cifrada.
 - **Puerto 8080:** Puerto de escucha del **Servidor Backend** para recibir las peticiones REST provenientes del frontend tras pasar por las reglas de red.
 - **Puerto 1521:** Puerto exclusivo de comunicación interna para que el backend acceda a la instancia de **Oracle Database**.
- Protocolos:
 - **HTTPS (SSL/TLS):** Utilizado para toda la comunicación externa, garantizando la seguridad en el intercambio de datos JSON entre el cliente y el backend.
 - **JDBC (Java Database Connectivity):** Protocolo de conexión directa que permite al **RepositorioSolicitud** ejecutar sentencias SQL y procedimientos almacenados en la base de datos.

- **TCP/IP:** Protocolo de red de nivel de transporte que permite la interconexión estable y persistente entre los tres nodos del sistema.

7.3. Diagrama de despliegue



8. +1 Escenarios Arquitectónicos

8.1. Escenario arquitectónico 1

- Nombre del escenario

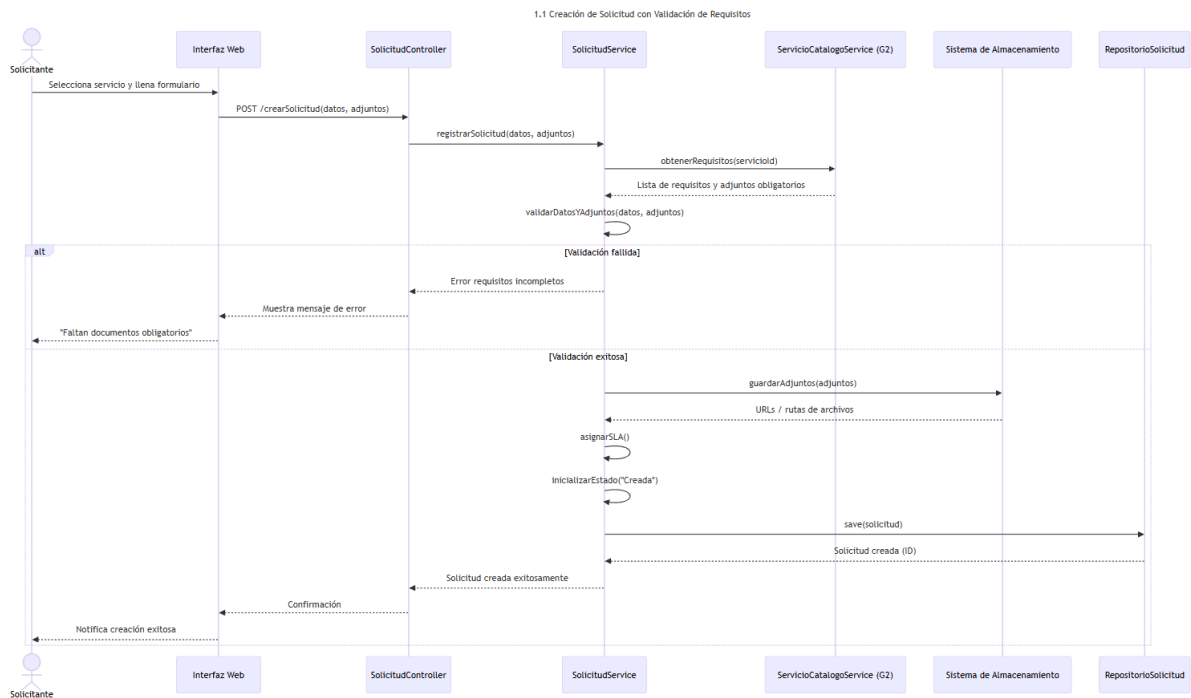
Creación de Solicitud con Validación de Requisitos

- Descripción

Este escenario describe el proceso mediante el cual el solicitante registra una nueva solicitud en el sistema. El sistema valida que los datos ingresados y los documentos adjuntos cumplan con los requisitos mínimos definidos antes de permitir el registro de la solicitud.s

- Componentes participantes
 - Solicitante
 - Interfaz Web
 - SolicitudController
 - SolicitudService

- Sistema de Almacenamiento
- RepositorioSolicitud
- Diagrama de secuencia



- Riesgos
 - Registro de solicitudes incompletas
 - Falta de documentos obligatorios
 - Errores en el almacenamiento de adjuntos
- Decisiones arquitectónicas
 - Centralizar la validación de requisitos en la capa de servicio
 - Validar la información antes de persistir la solicitud
 - Separar el almacenamiento de archivos del registro de la solicitud

8.2. Escenario arquitectónico 2

- Nombre del escenario

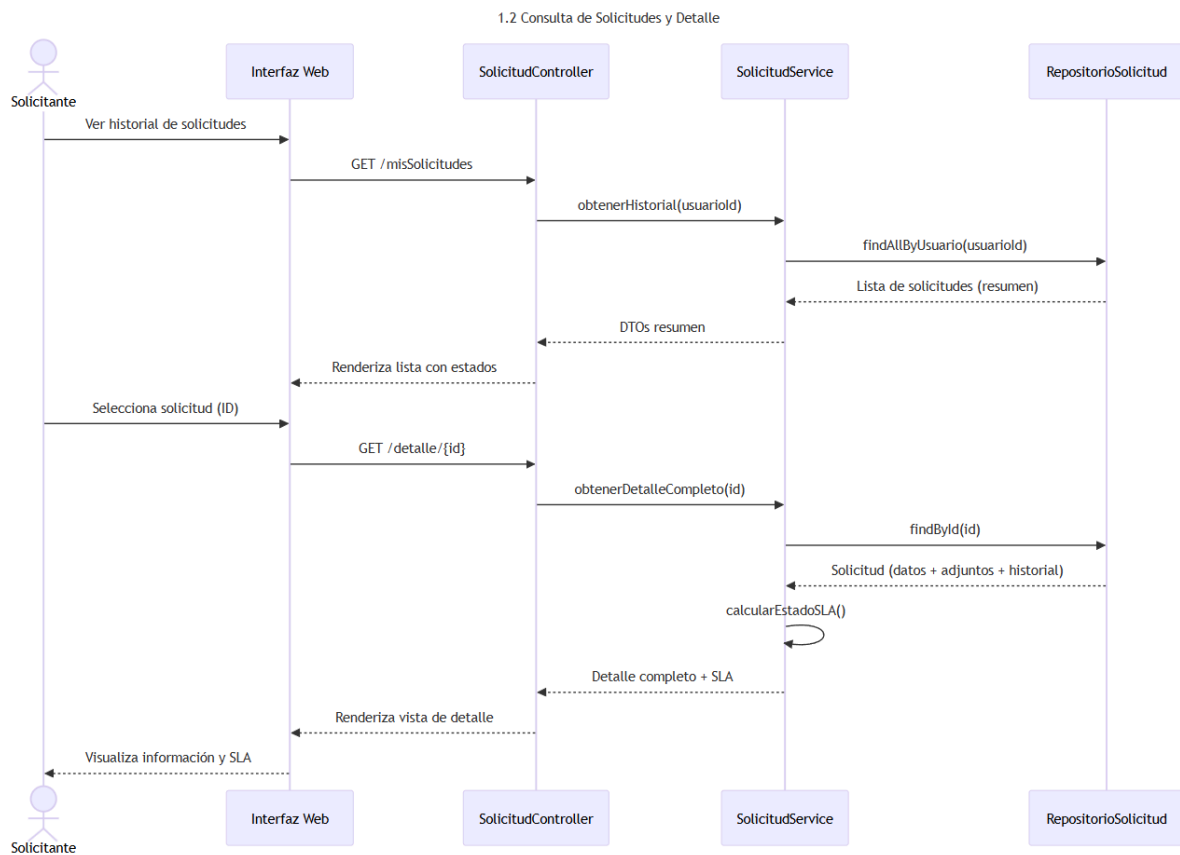
Consulta de Solicitudes y Visualización de Detalle

- Descripción

Este escenario permite al solicitante consultar el historial de sus solicitudes y visualizar el detalle completo de una solicitud específica, incluyendo su estado actual y la información registrada.

- Componentes participantes
 - Solicitante
 - Interfaz Web
 - SolicitudController
 - SolicitudService

- RepositorioSolicitud
- Diagrama de secuencia



- Riesgos
 - Visualización incorrecta de información
 - Inconsistencias en el estado mostrado
 - Retrasos en la recuperación de datos
- Decisiones arquitectónicas
 - Centralizar la obtención de información en la capa de servicio
 - Separar la lógica de consulta de la capa de presentación
 - Optimizar las consultas al repositorio

8.3. Escenario arquitectónico 3

- Nombre del escenario

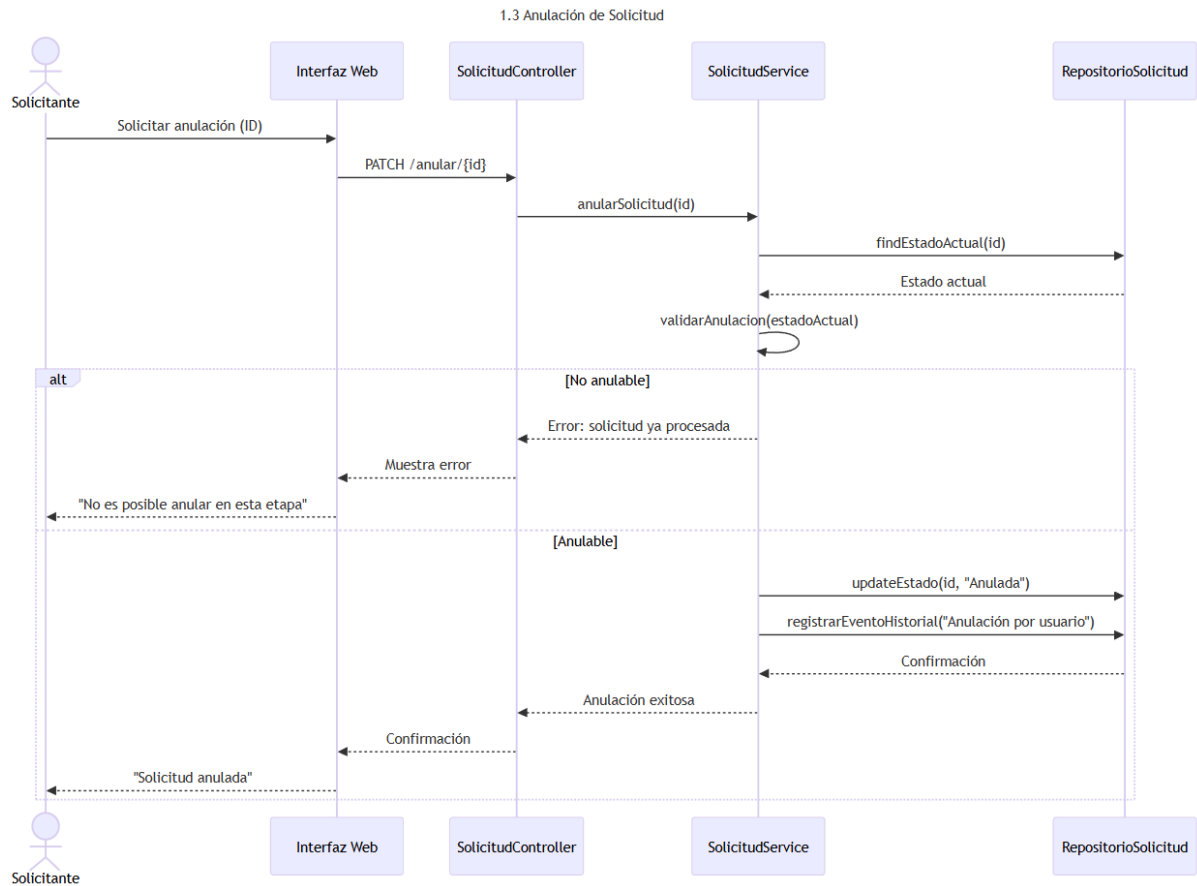
Anulación de Solicitud

- Descripción

Este escenario describe el proceso mediante el cual el solicitante solicita la anulación de una solicitud. El sistema valida que la solicitud se encuentre en un estado permitido antes de actualizar su estado y registrar el evento correspondiente.

- Componentes participantes
 - Solicitante

- Interfaz Web
- SolicitudController
- SolicitudService
- RepositorioSolicitud
- Diagrama de secuencia



- Riesgos
 - Anulación de solicitudes en estados no permitidos
 - Pérdida de trazabilidad del cambio de estado
 - Inconsistencias en el historial de la solicitud
- Decisiones arquitectónicas
 - Validar las reglas de negocio antes de modificar el estado
 - Registrar los cambios de estado para asegurar trazabilidad
 - Centralizar el control del ciclo de vida en la capa de servicio

9. Consideraciones de calidad

- Escalabilidad
- Mantenibilidad
- Seguridad (básica)

- Disponibilidad

10. Conclusiones

- Beneficios de la arquitectura propuesta
- Limitaciones
- Posibles mejoras futuras

11. Anexos (opcional)

- Referencias
- Glosario
- Notas técnicas