



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas

Proyecto Acortador de URLs

Curso: *Diseño y arquitectura de software*

Docente: *Mag. Patrick Cuadros Quiroga*

Integrantes:

Fernandez Villanueva, Daleska Nicolle (2021070308)

Chire Ramos, Mayra Fernanda (2021072620)

Flores Melendez, Andree Sebastian (2017057494)

Tacna – Perú

2024





CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	MPV	ELV	ARV	10/10/2020	Versión Original

Sistema Acortador de URLs

Documento de Arquitectura de Software

Versión {1.0}



CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	MPV	ELV	ARV	10/10/2020	Versión Original

ÍNDICE GENERAL

1. INTRODUCCIÓN	6
1.1. Propósito	6
1.2. Alcance	6
1.3. Definición, siglas y abreviaturas	6
2. OBJETIVOS Y RESTRICCIONES ARQUITECTÓNICAS	6
Objetivos	6
Restricciones Arquitectónicas	7
2.1. Priorización de requerimientos	8
1.1.1. Requerimientos Funcionales	8
1.1.2. Requerimientos No Funcionales	9
2.2. Restricciones	9
3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA	10
3.1. Vista de Caso de uso	10
1.1.3. Diagramas de Casos de uso	10
3.2. Vista Lógica	11
3.2.1. Diagrama de Subsistemas (paquetes)	11
3.2.2. Diagrama de Secuencia (vista de diseño)	11
3.2.3. Diagrama de Colaboración (vista de diseño)	16
3.2.4. Diagrama de Objetos	20
3.2.5. Diagrama de Clases	20
3.2.6. Diagrama de Base de datos (relacional o no relacional)	20
3.3. Vista de Implementación (vista de desarrollo)	20
3.3.1. Diagrama de arquitectura software (paquetes)	20
3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes)	20
3.4. Vista de procesos	20
3.4.1. Diagrama de Procesos del sistema (diagrama de actividad)	21
3.5. Vista de Despliegue (vista física)	21
3.5.1. Diagrama de despliegue	21
4. ATRIBUTOS DE CALIDAD DEL SOFTWARE	21
Escenario de Funcionalidad	21
Escenario de Usabilidad	22
Escenario de confiabilidad	22
Escenario de rendimiento	22
Escenario de mantenibilidad	22



Otros Escenarios



23



1. INTRODUCCIÓN

1.1. Propósito

El propósito del proyecto "LinkEase" es desarrollar una aplicación para acortar, personalizar y gestionar URLs, mejorando así la manera en que los usuarios y las empresas comparten y analizan enlaces en el entorno digital. Al proporcionar herramientas avanzadas de análisis y personalización, "LinkEase" busca facilitar la gestión de enlaces y optimizar las estrategias de marketing y comunicación digital, ofreciendo una experiencia de usuario eficiente y segura.

1.2. Alcance

El proyecto "LinkEase" abarca el desarrollo de una plataforma web intuitiva para acortar URLs, la creación de extensiones de navegador para facilitar su uso, la implementación de una API robusta para integraciones con terceros, el desarrollo de aplicaciones móviles para Android e iOS, y la integración de herramientas avanzadas de análisis y generación de reportes en tiempo real, permitiendo a los usuarios gestionar y rastrear eficazmente sus enlaces acortados.

1.3. Definición, siglas y abreviaturas

- API (Application Programming Interface): Conjunto de protocolos y herramientas para construir y conectar software de aplicaciones.
- URL (Uniform Resource Locator): Dirección específica que se utiliza para acceder a recursos en la web.
- UI (User Interface): Interfaz de usuario, el espacio donde las interacciones entre humanos y máquinas ocurren.
- UX (User Experience): Experiencia de usuario, cómo una persona se siente al interactuar con un sistema.
- HTTP (Hypertext Transfer Protocol): Protocolo de comunicación utilizado en la World Wide Web.
- HTTPS (Hypertext Transfer Protocol Secure): Versión segura de HTTP, que utiliza SSL/TLS para cifrar los datos.

2. OBJETIVOS Y RESTRICCIONES ARQUITECTÓNICAS

Objetivos

Eficiencia en la Gestión de Enlaces: Desarrollar una plataforma que permita a los usuarios acortar, personalizar y gestionar URLs de manera rápida y eficiente.

Experiencia de Usuario: Crear una interfaz de usuario intuitiva y fácil de usar, tanto en la plataforma web como en las aplicaciones móviles, que mejore la experiencia general del usuario.



Seguridad y Privacidad: Asegurar que la plataforma cumpla con las mejores prácticas de seguridad y proteja los datos personales de los usuarios, cumpliendo con las normativas de privacidad vigentes.

Escalabilidad y Rendimiento: Diseñar una arquitectura que soporte el crecimiento del número de usuarios y enlaces sin comprometer el rendimiento del sistema.

Restricciones Arquitectónicas

Tecnologías Específicas:

- **Backend:** El sistema debe utilizar Node.js para el servidor y MongoDB como base de datos, aprovechando su flexibilidad y capacidad de manejo de grandes volúmenes de datos.
- **Frontend:** La interfaz de usuario debe ser desarrollada con React, garantizando una experiencia fluida y dinámica.

Seguridad:

- Todas las comunicaciones entre clientes y servidores deben estar cifradas usando HTTPS.
- Implementar autenticación y autorización seguras, utilizando OAuth 2.0 para la API.

Disponibilidad y Fiabilidad:

- El sistema debe tener una disponibilidad del 99.9%, minimizando el tiempo de inactividad.
- Implementar redundancia en los servidores y balanceo de carga para garantizar la fiabilidad del servicio.

Escalabilidad:

- La arquitectura debe permitir la escalabilidad horizontal, añadiendo más servidores según la demanda.
- Utilizar servicios de cloud computing como AWS o Azure para gestionar la infraestructura de manera flexible y escalable.

Rendimiento:

- El tiempo de respuesta para las operaciones de acortamiento de URL debe ser inferior a 200ms bajo condiciones normales de carga.
- Optimizar las consultas a la base de datos para manejar eficientemente grandes volúmenes de datos y solicitudes concurrentes.

Compatibilidad:

- La plataforma web debe ser compatible con los navegadores más utilizados (Chrome, Firefox, Safari, Edge).



2.1. Priorización de requerimientos

1.1.1. Requerimientos Funcionales

ID. Requerimiento	Nombre del Requisito	Descripción de Requisito	Prioridad
RF-001	Autenticación de Usuarios	Implementación de un sistema de login seguro para que los usuarios puedan acceder de manera segura a sus cuentas.	Alta
RF-002	Acortamiento de URLs	Capacidad para ingresar una URL larga y generar una URL corta única.	Alta
RF-003	Análisis y Estadísticas	Seguimiento y generación de estadísticas detalladas sobre el uso de las URLs acortadas.	Alta
RF-004	Gestión de Usuarios	Registro y gestión de usuarios con información como nombre, correo electrónico y suscripción.	Alta
RF-005	Personalización de URLs	Opción para que usuarios con suscripción VIP o PRO personalicen el alias de sus URLs cortas.	Alta
RF-006	Suscripción FREE	Implementación del nivel de suscripción FREE que permite a los usuarios acortar URLs con un límite específico.	Alta
RF-007	Suscripción VIP	Implementación del nivel de suscripción VIP que ofrece un límite más alto de URLs acortadas y funcionalidades adicionales.	Alta
RF-008	Suscripción PRO	Implementación del nivel de suscripción PRO que permite acortar URLs ilimitadamente y acceso completo a todas las características del servicio.	Alta
RF-009	Renovación de Suscripciones	Funcionalidades para que los usuarios puedan renovar fácilmente sus suscripciones desde su cuenta.	Alta
RF-0010	Cancelación de Suscripciones	Funcionalidades para que los usuarios puedan cancelar sus suscripciones de manera sencilla desde su cuenta.	Alta



1.1.2. Requerimientos No Funcionales

ID. Requerimiento	Nombre del Requisito	Descripción de Requisito	Prioridad
RF-001	Seguridad	Implementar medidas de seguridad como validación de entradas y protección contra ataques.	Alta
RF-002	Rendimiento	Garantizar tiempos de respuesta rápidos al acortar y redirigir URLs, incluso bajo carga alta.	Alta
RF-003	Disponibilidad	Mantener la plataforma disponible y accesible la mayor parte del tiempo (alta disponibilidad)..	Alta
RF-004	Usabilidad	Diseñar una interfaz intuitiva y fácil de usar para los usuarios, tanto para acortar como para administrar URLs.	Media
RF-005	Mantenibilidad	Facilitar la mantenibilidad del sistema mediante código limpio, documentación adecuada y pruebas robustas.	Media

2.2. Restricciones

Presupuestarias:

- El presupuesto asignado para el proyecto es limitado incluyendo desarrollo, infraestructura, marketing y costos operativos iniciales.
- Se requiere minimizar los costos de mantenimiento mensual del sistema para asegurar la rentabilidad a corto plazo.

Temporales:

- El proyecto debe completarse en un plazo establecido desde el inicio del desarrollo hasta el lanzamiento oficial.
- Las fases de diseño, desarrollo, pruebas y lanzamiento deben cumplir con calendarios específicos para cumplir con los plazos establecidos.

Tecnológicas:

- La plataforma debe ser compatible con los navegadores web más utilizados como Chrome, Firefox, Safari y Edge.



- Se debe utilizar tecnologías modernas y escalables como Node.js para el backend y React.js para el frontend, garantizando un rendimiento óptimo y una experiencia de usuario fluida.

Seguridad y Privacidad:

- Cumplir con las normativas de protección de datos personales como GDPR o CCPA para garantizar la privacidad de los usuarios.
- Implementar medidas de seguridad robustas, como cifrado HTTPS y autenticación segura, para proteger la plataforma contra ataques cibernéticos y vulnerabilidades.

Recursos Humanos:

- El equipo de desarrollo está limitado a 3 miembros clave, incluyendo desarrolladores backend, frontend, diseñadores
- Se requiere capacitación continua y actualización de habilidades para el equipo de desarrollo para mantenerse al día con las últimas tecnologías y prácticas de seguridad.

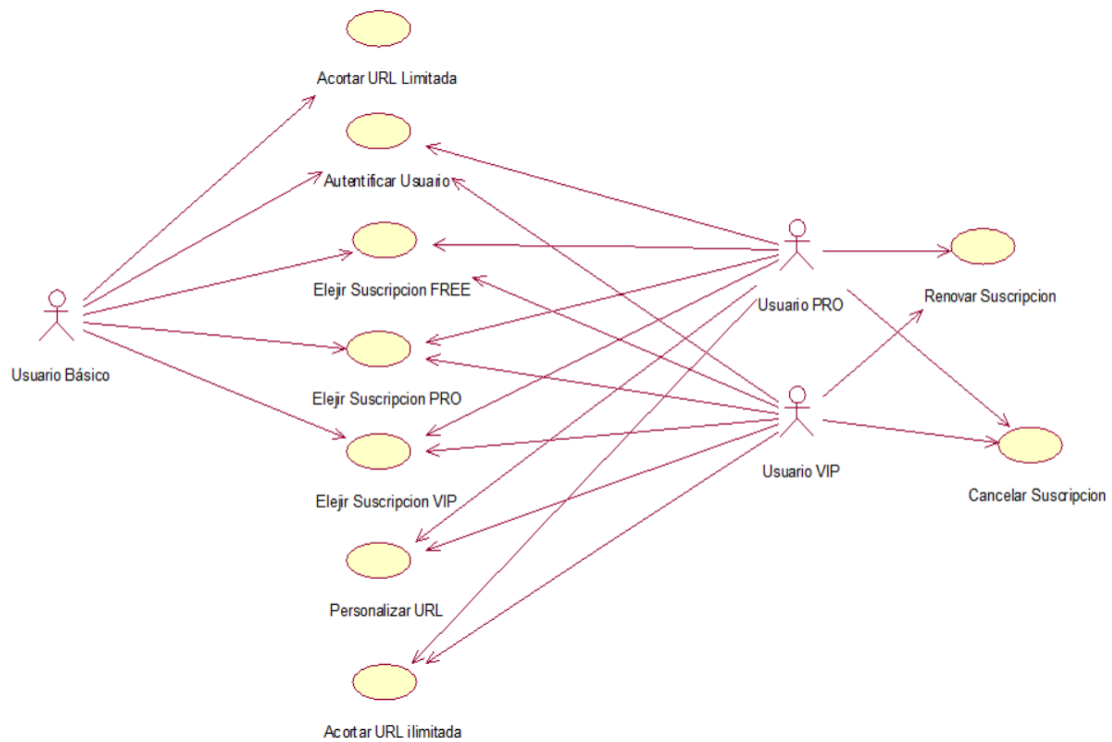
Escalabilidad:

- Diseñar la arquitectura del sistema para que sea escalable horizontalmente, permitiendo manejar un crecimiento significativo en el número de usuarios y solicitudes sin degradación del rendimiento.
- Utilizar servicios en la nube como AWS o Azure para gestionar la infraestructura de manera flexible y escalable.

3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA

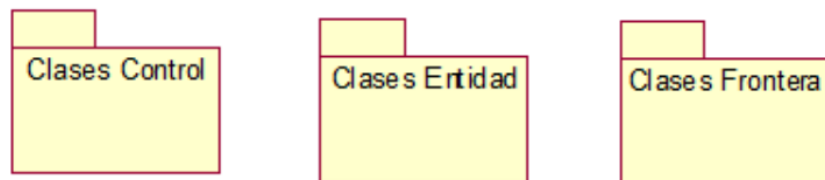
3.1. Vista de Caso de uso

1.1.3. Diagramas de Casos de uso



3.2. Vista Lógica

3.2.1. Diagrama de Subsistemas (paquetes)



3.2.2. Diagrama de Secuencia (vista de diseño)

Diagrama de Secuencia del CUS Acortar URL Limitado

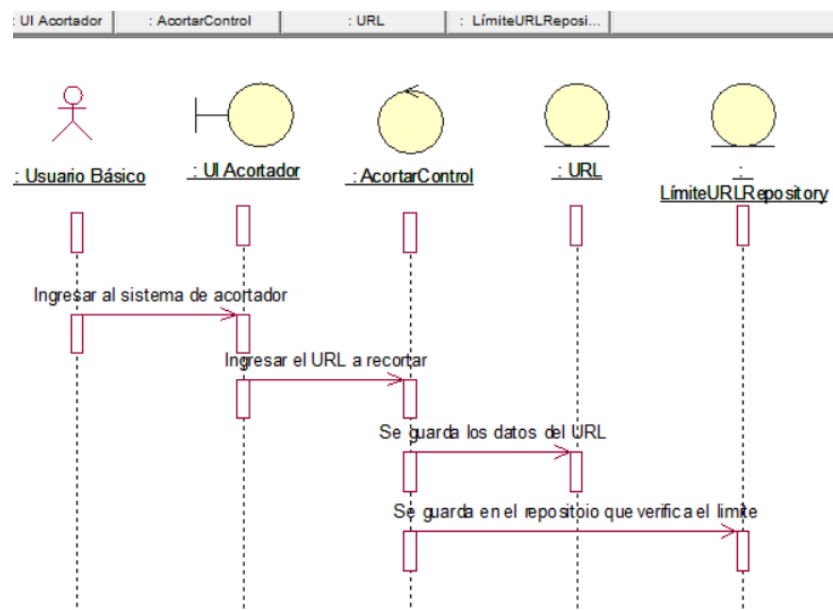


Diagrama de Secuencia del CUS Acortar URL Ilimitada

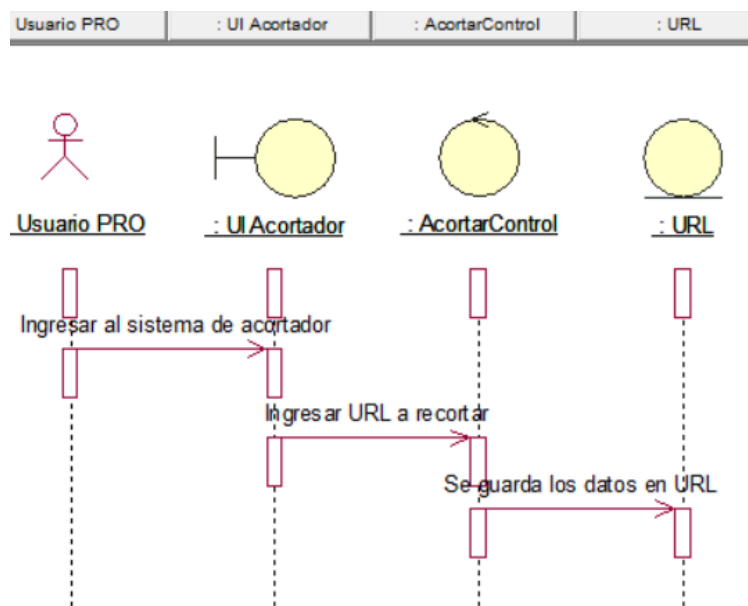




Diagrama de Secuencia del CUS Autenticar Usuario

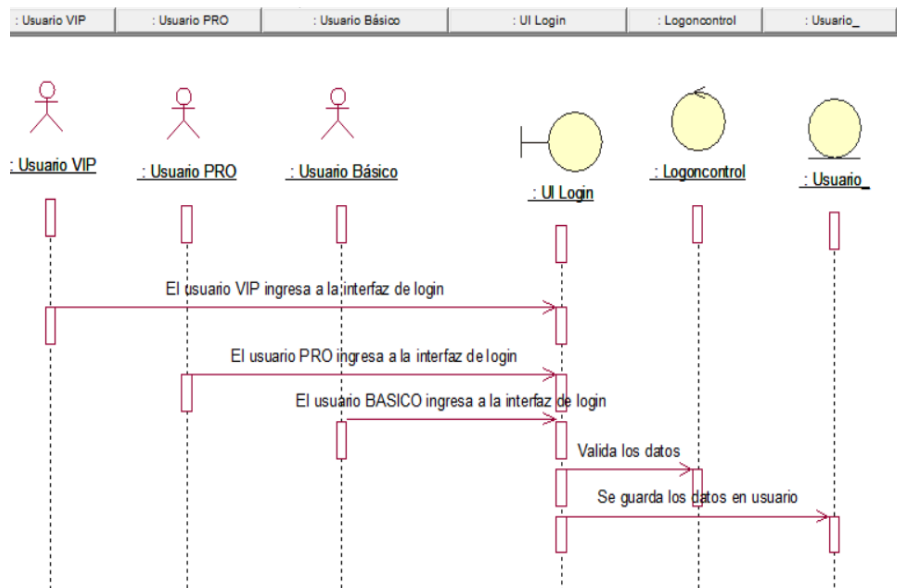


Diagrama de Secuencia del CUS Elegir Suscripción FREE

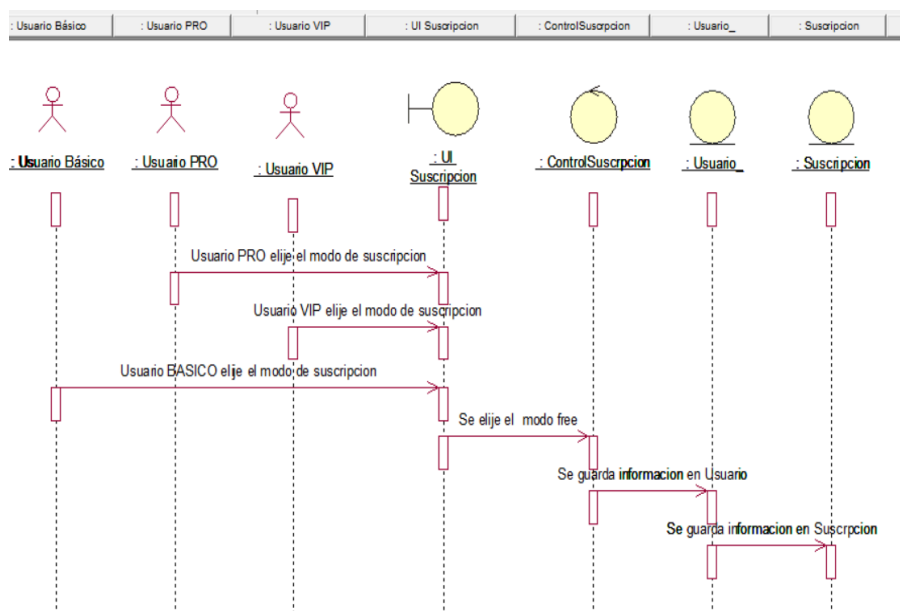




Diagrama de Secuencia del CUS Elegir Suscripción PRO

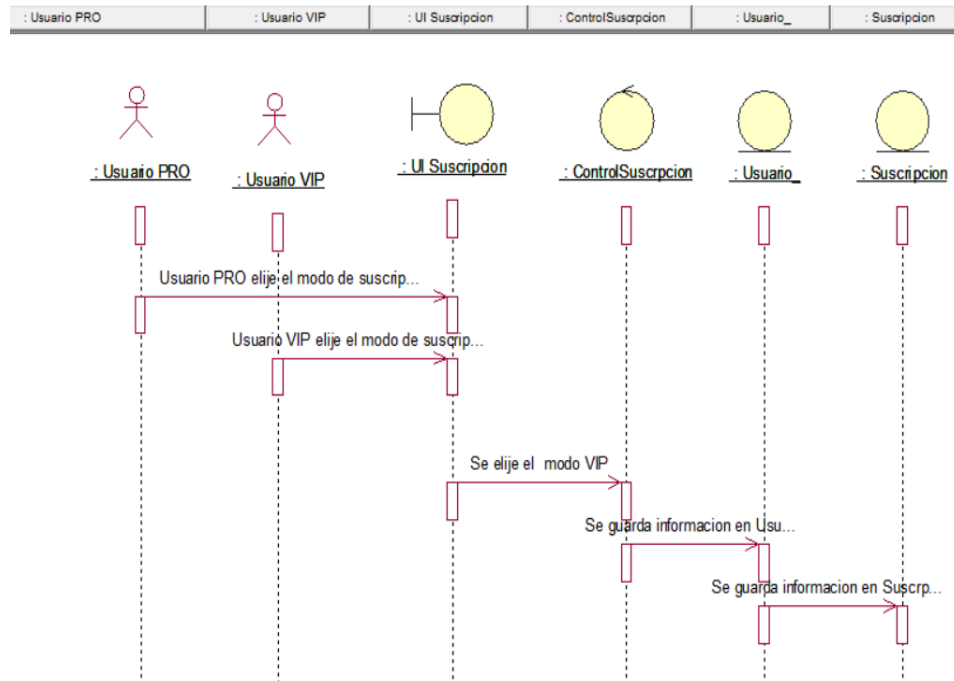


Diagrama de Secuencia del CUS Elegir Suscripción VIP

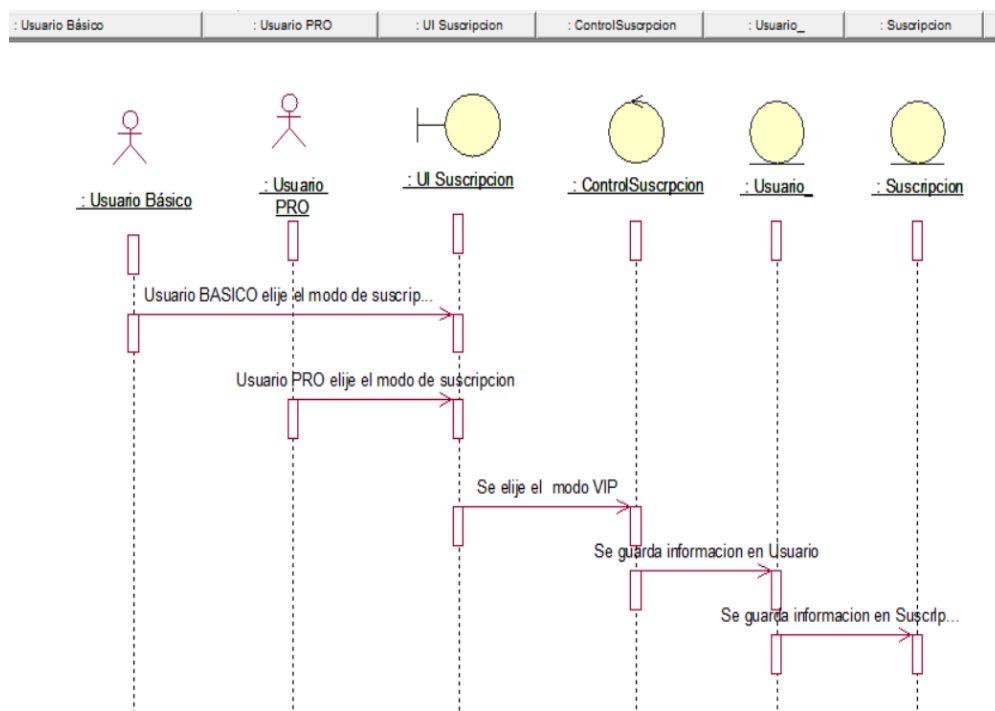




Diagrama de Secuencia del CUS Personalizar URL

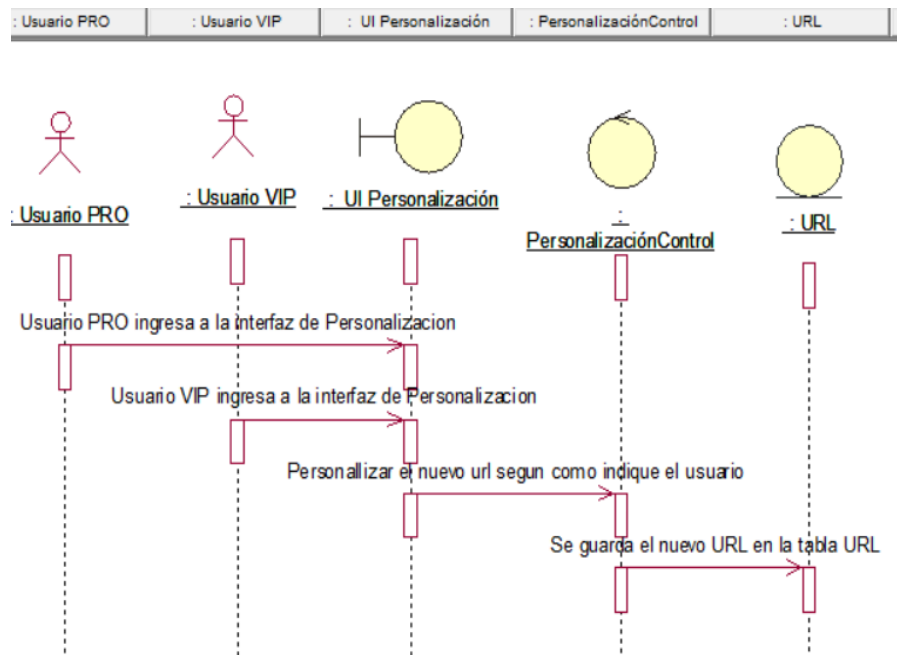


Diagrama de Secuencia del CUS Renovar Suscripción

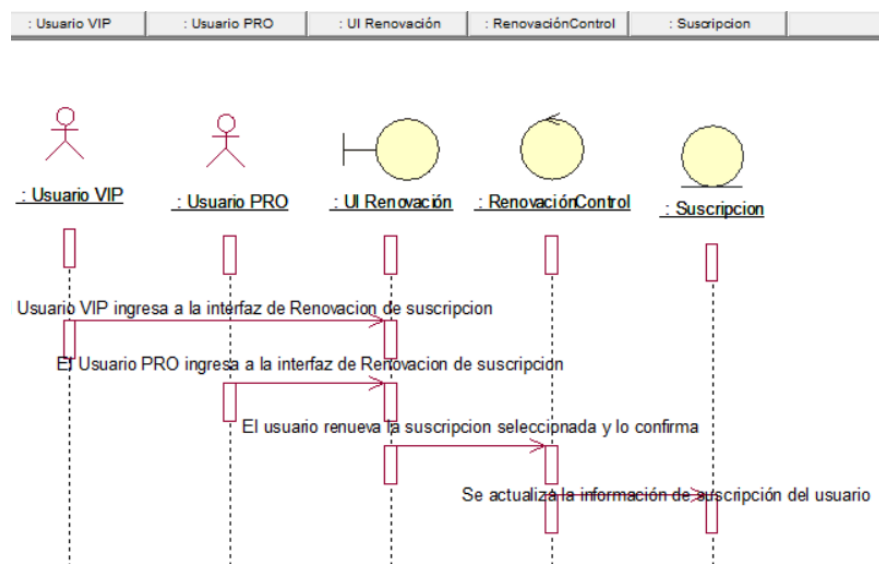
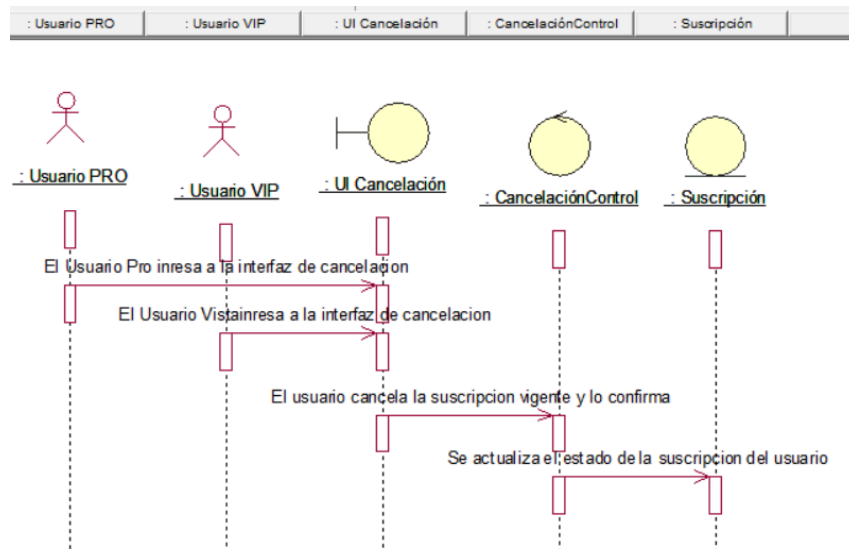




Diagrama de Secuencia del CUS Cancelar Suscripción



3.2.3. Diagrama de Colaboración (vista de diseño)

Diagrama de Secuencia del CUS Acortar URL Limitado

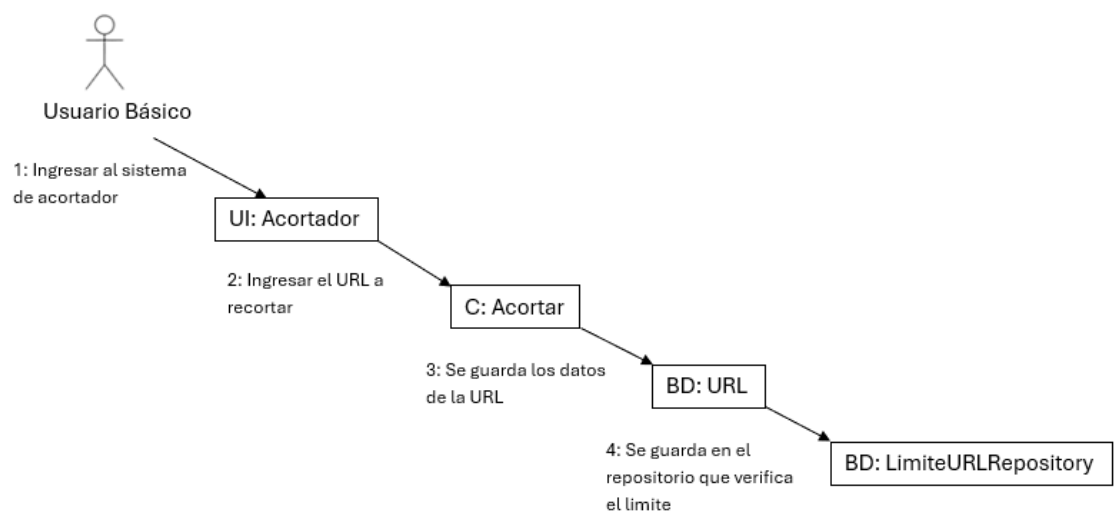




Diagrama de Secuencia del CUS Acortar URL Ilimitada

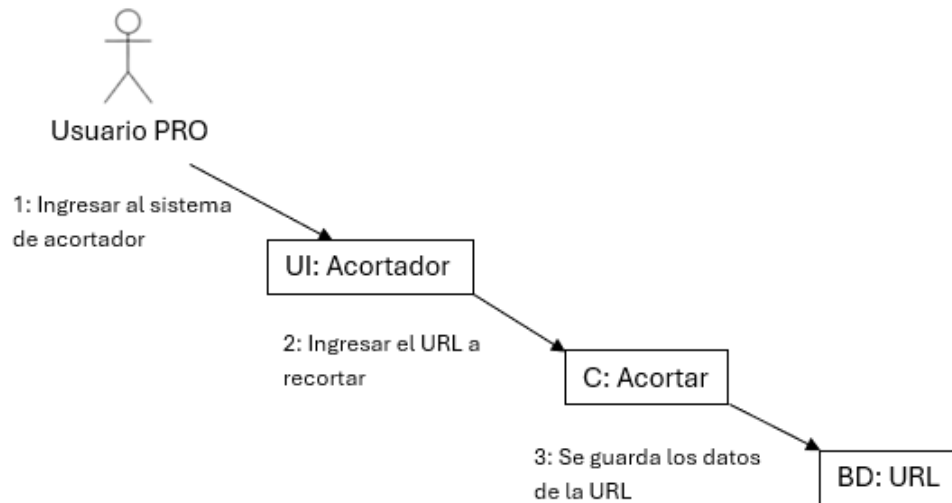


Diagrama de Secuencia del CUS Autenticar Usuario

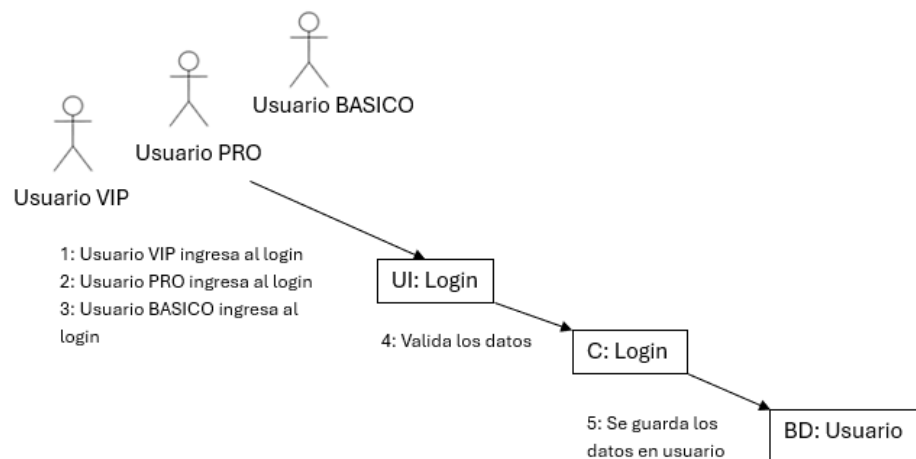


Diagrama de Secuencia del CUS Elegir Suscripción FREE

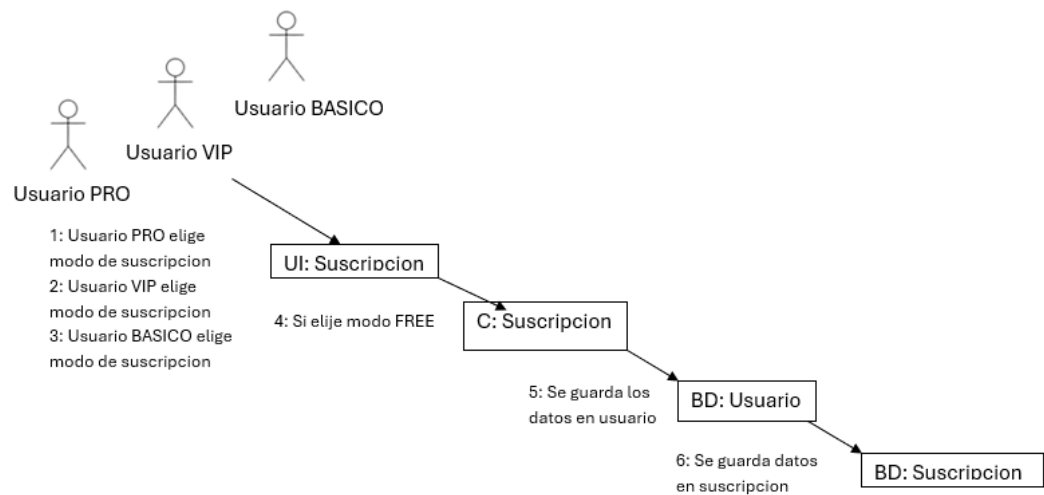


Diagrama de Secuencia del CUS Elegir Suscripción PRO

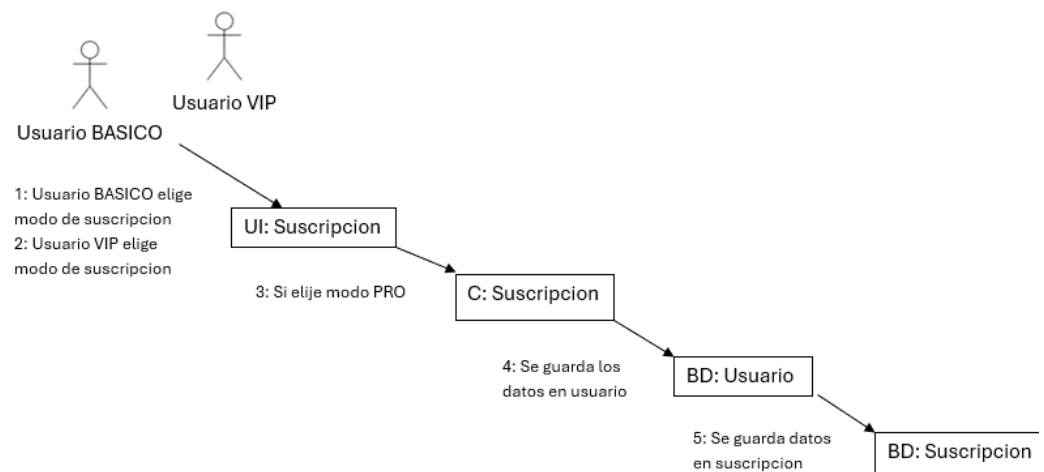


Diagrama de Secuencia del CUS Elegir Suscripción VIP

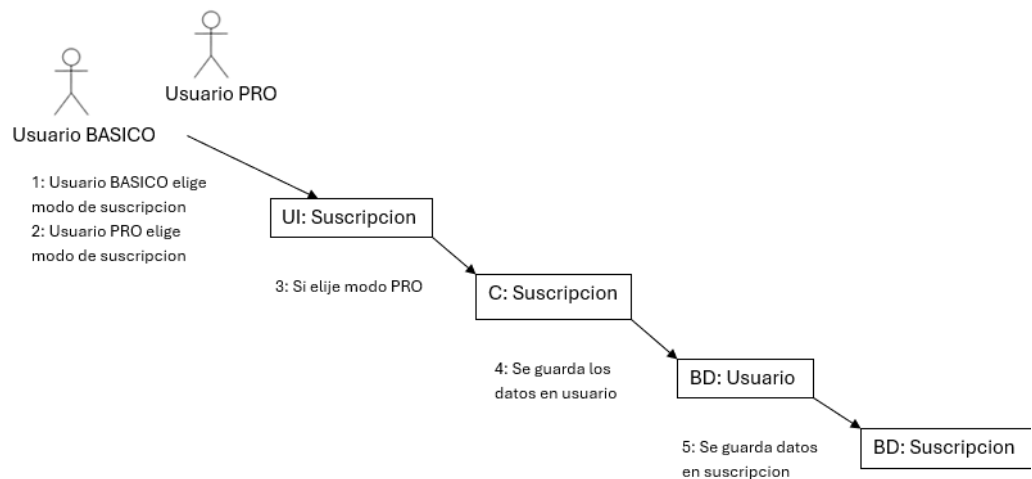




Diagrama de Secuencia del CUS Personalizar URL

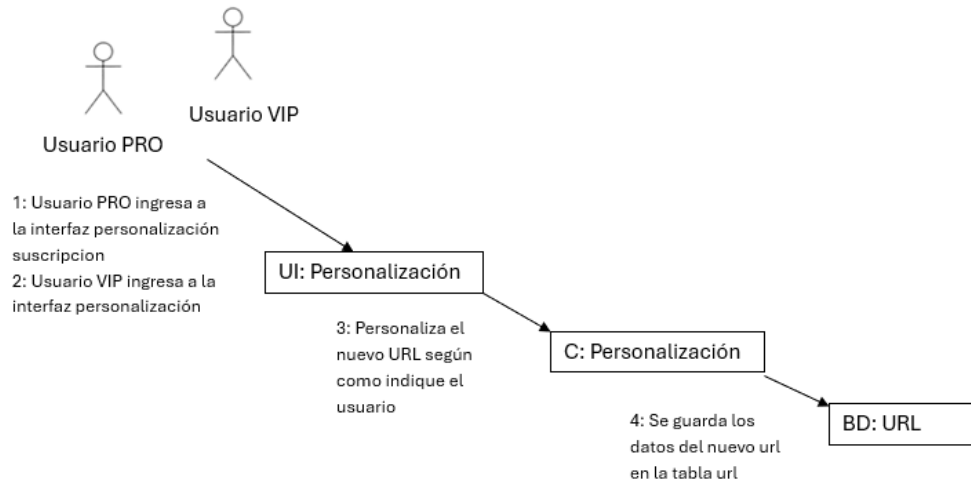


Diagrama de Secuencia del CUS Renovar Suscripción

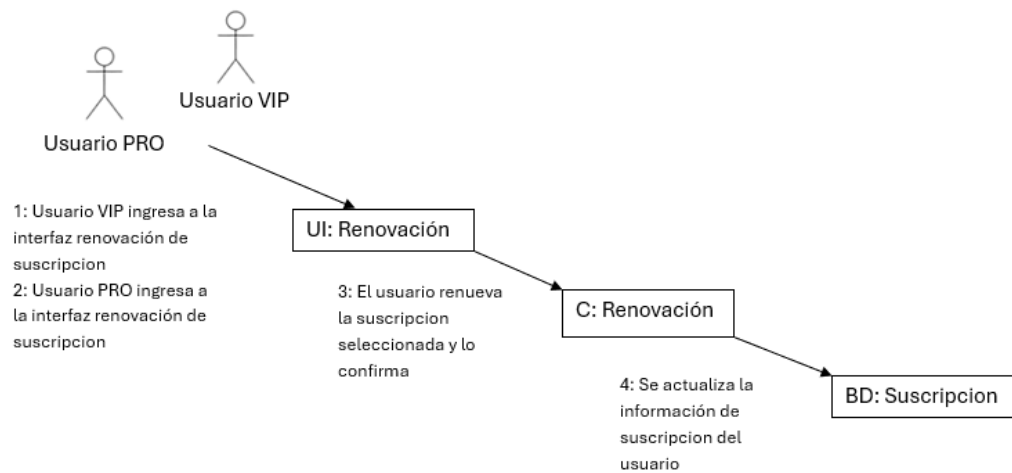
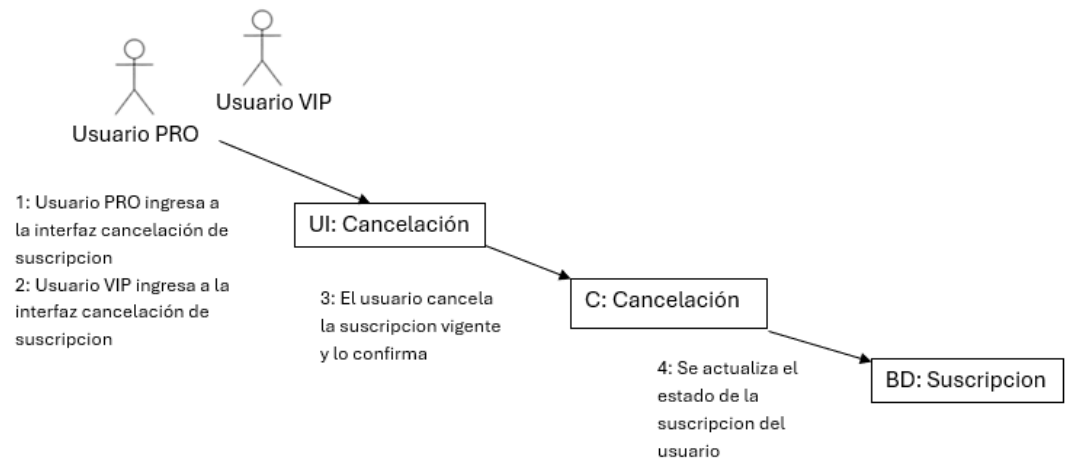


Diagrama de Secuencia del CUS Cancelar Suscripción





3.2.4. Diagrama de Objetos

Diagrama de Objetos del CUS Acortar URL Limitado

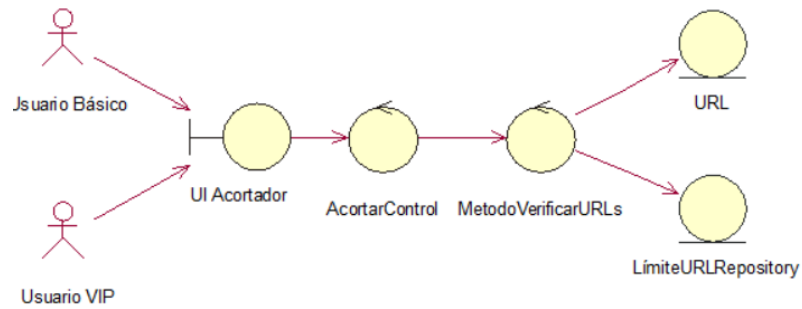


Diagrama de Objetos del CUS Acortar URL Ilimitada

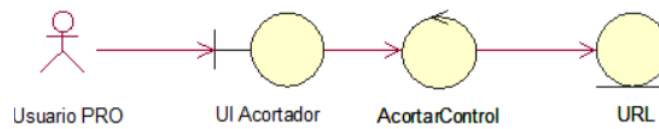


Diagrama de Objetos del CUS Autenticar Usuario

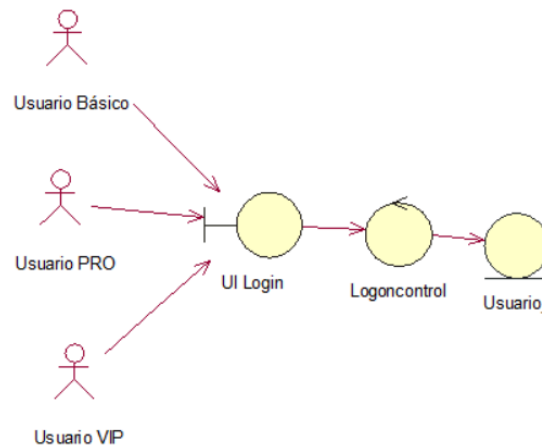


Diagrama de Objetos del CUS Elegir Suscripción FREE

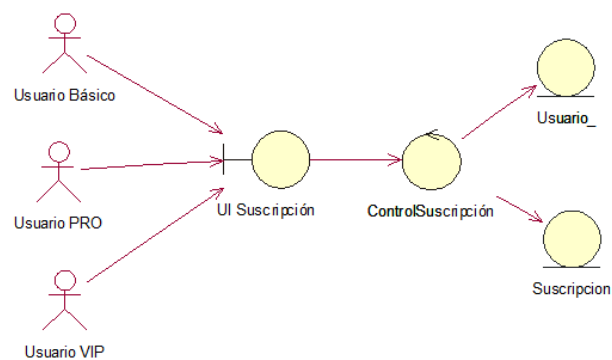




Diagrama de Objetos del CUS Elegir Suscripción PRO

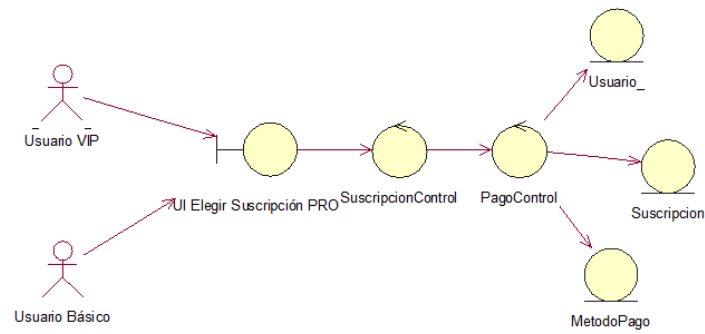


Diagrama de Objetos del CUS Elegir Suscripción VIP

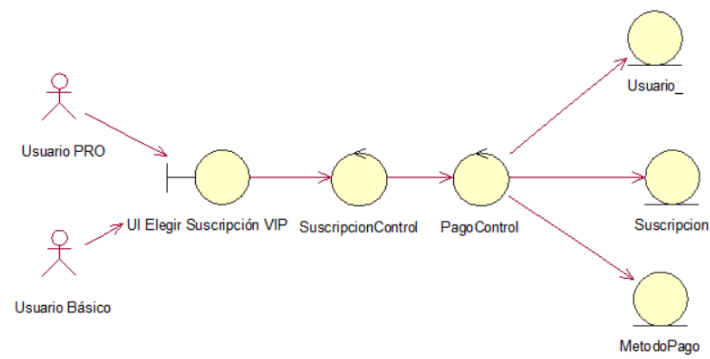


Diagrama de Objetos del CUS Personalizar URL

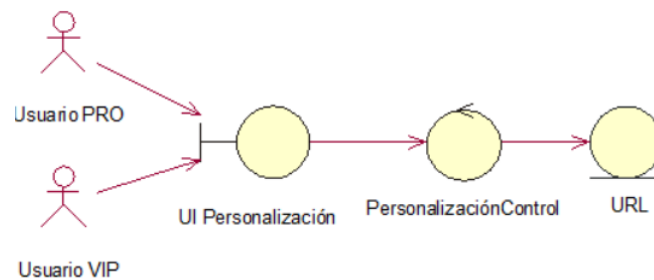


Diagrama de Objetos del CUS Renovar Suscripción

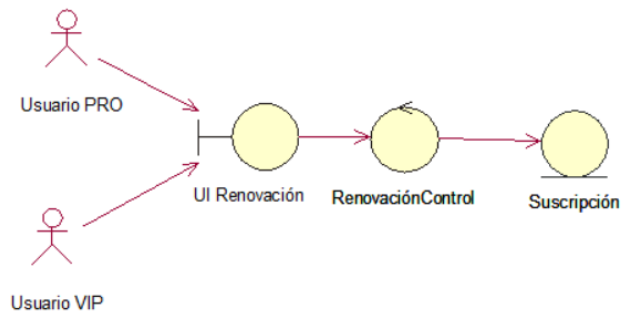
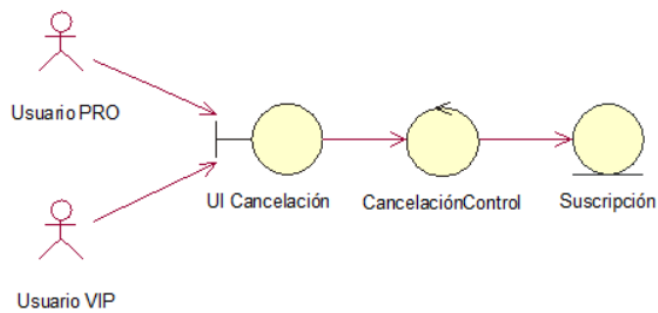
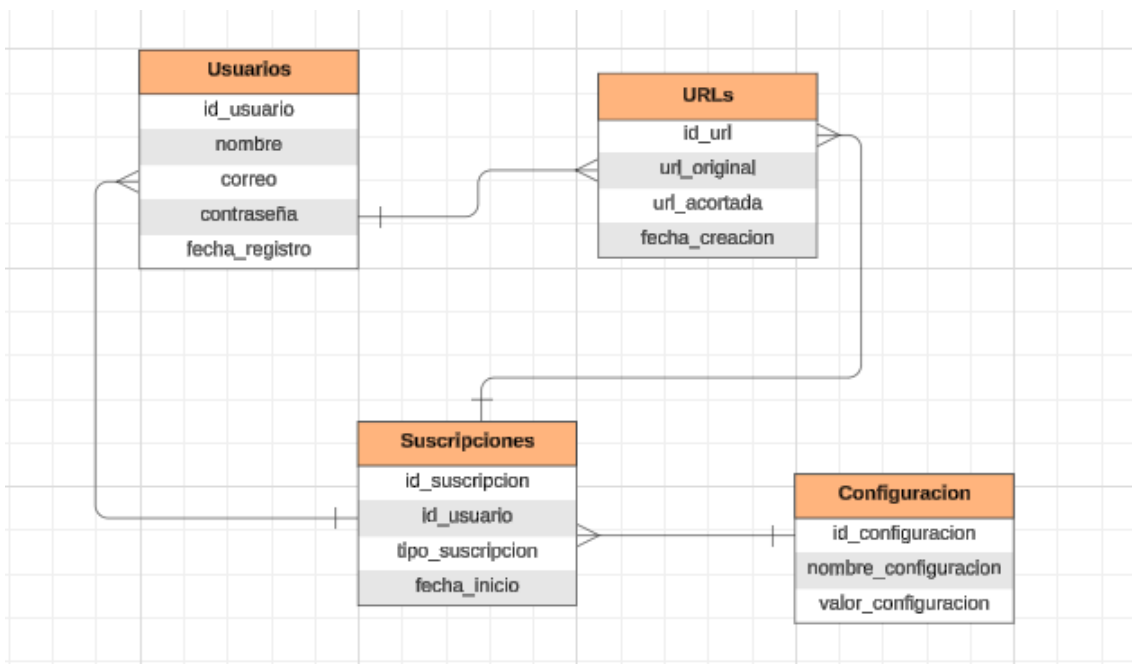


Diagrama de Objetos del CUS Cancelar Suscripción



3.2.5. Diagrama de Clases

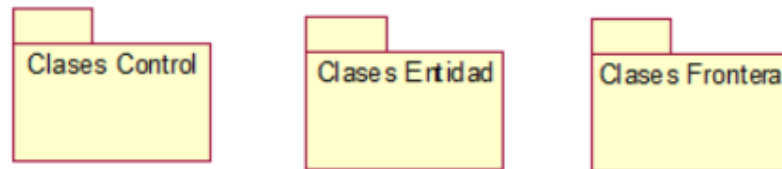
3.2.6. Diagrama de Base de datos (relacional o no relacional)





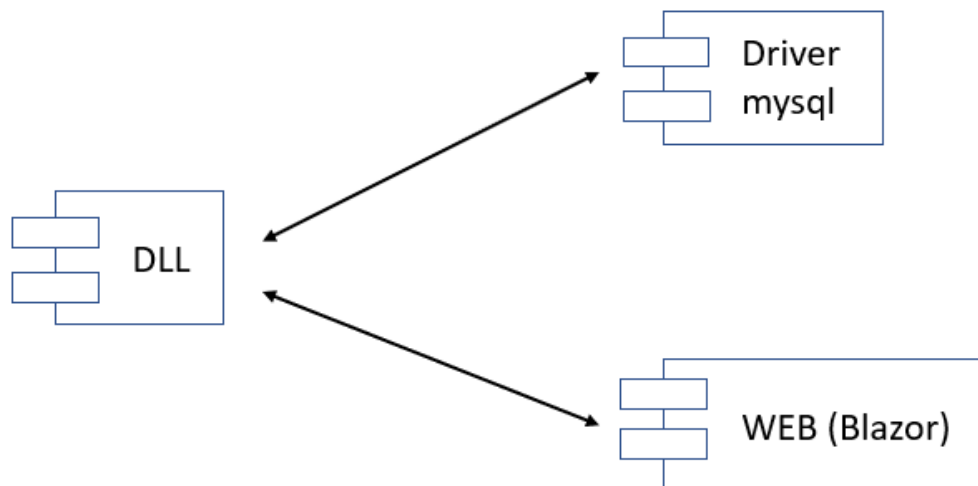
3.3. Vista de Implementación (vista de desarrollo)

3.3.1. Diagrama de arquitectura software (paquetes)



3.3.2. Diagrama de arquitectura del sistema

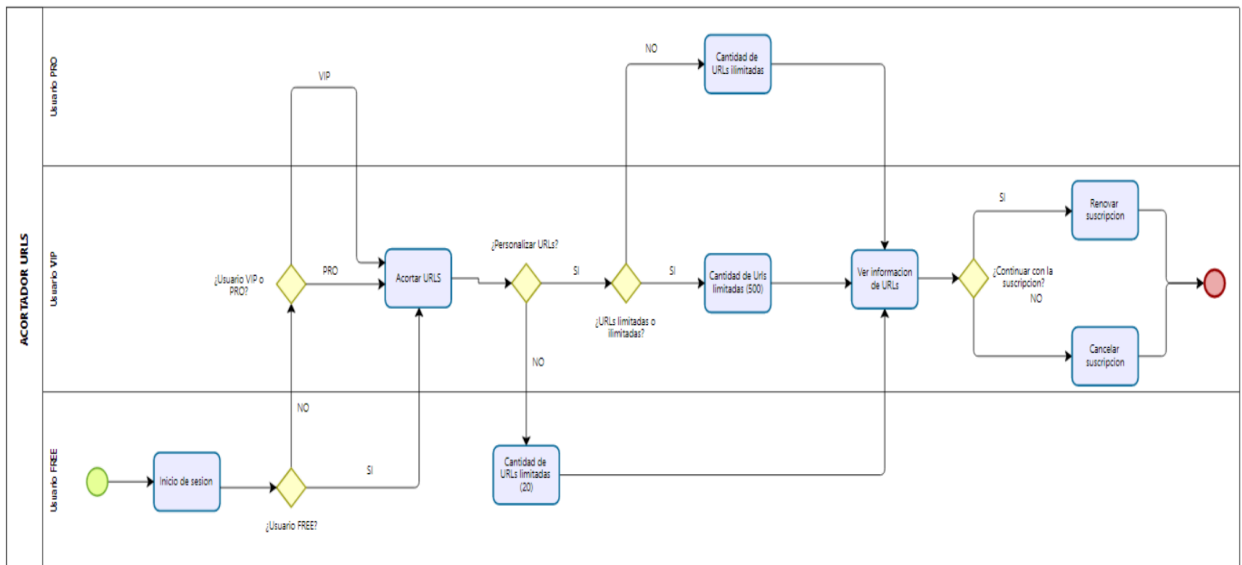
3.3.3. Diagrama de componentes





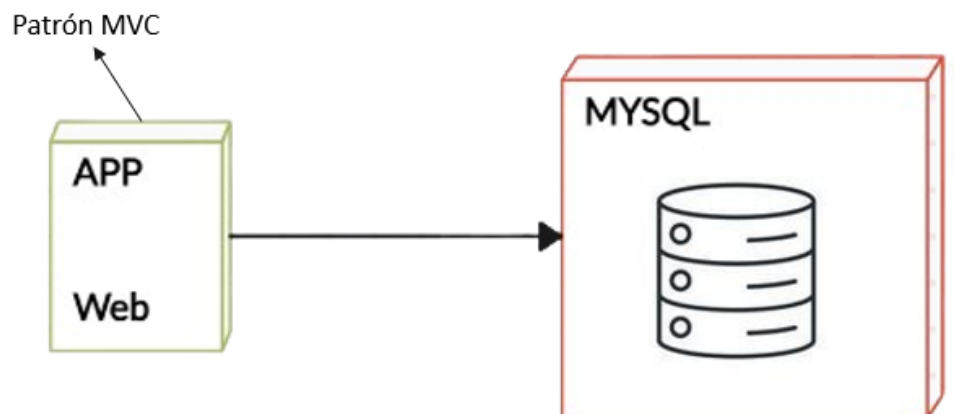
3.4. Vista de procesos

3.4.1. Diagrama de Procesos del sistema (diagrama de actividad)



3.5. Vista de Despliegue (vista física)

3.5.1. Diagrama de despliegue



4. ATRIBUTOS DE CALIDAD DEL SOFTWARE

Escenario de Funcionalidad

En el contexto del proyecto "LinkEase", la funcionalidad del sistema se centra en permitir a los usuarios acortar URLs de manera efectiva y eficiente. Esto implica que el sistema debe ser capaz de procesar URLs de cualquier longitud y formato válido. Además, los enlaces acortados generados deben redirigir de



manera precisa y confiable al URL original correspondiente. Para mejorar la experiencia del usuario, se ofrecerá la opción de personalizar los enlaces acortados, permitiendo a los usuarios crear URLs más significativas y fáciles de recordar.

Escenario de Usabilidad

La usabilidad de "LinkEase" se enfoca en proporcionar una plataforma intuitiva y fácil de usar para usuarios de todos los niveles de experiencia. La interfaz de usuario estará diseñada de manera clara y organizada, minimizando la necesidad de instrucciones adicionales para realizar tareas básicas como acortar URLs. Los usuarios recibirán retroalimentación visual inmediata después de cada acción, asegurando que comprendan el resultado de sus acciones de manera clara. Las funciones principales del sistema serán accesibles en pocos clics desde la página principal, facilitando una navegación fluida y eficiente.

Escenario de confiabilidad

La confiabilidad de "LinkEase" es crucial para garantizar que el sistema sea estable y funcione de manera consistente bajo diferentes condiciones de carga y uso. Se establecerá como objetivo una disponibilidad del sistema del 99.9% durante el horario de operación normal, asegurando que los usuarios puedan acceder al servicio sin interrupciones significativas. Además, se implementarán medidas proactivas para mantener un tiempo medio entre fallos (MTBF) superior a 30 días, minimizando así los tiempos de inactividad no planificados. Se llevarán a cabo pruebas exhaustivas de unidad, integración y sistema para identificar y corregir posibles fallos antes del lanzamiento oficial.

Escenario de rendimiento

El rendimiento de "LinkEase" se enfoca en asegurar que el sistema responda de manera rápida y eficiente bajo cargas de trabajo normales y picos de usuarios. Se establecerá como objetivo un tiempo de respuesta promedio para acortar una URL de menos de 200 milisegundos, garantizando así una experiencia fluida para los usuarios. El sistema estará diseñado para manejar al menos 1000 solicitudes de acortamiento de URLs por minuto sin una degradación significativa del rendimiento. Se implementará un monitoreo continuo para ajustar y optimizar la infraestructura según las demandas de carga cambiantes, asegurando que el rendimiento del sistema se mantenga óptimo en todo momento.

Escenario de mantenibilidad

La mantenibilidad de "LinkEase" se centrará en asegurar que el código y la arquitectura del sistema sean fáciles de mantener y extender en el futuro. El código será estructurado siguiendo principios de diseño limpio y modular, facilitando así la comprensión y la realización de cambios por parte de los



desarrolladores. Se documentará exhaustivamente tanto el código como la arquitectura del sistema, proporcionando una referencia clara y completa para futuras actualizaciones y mejoras. Se utilizarán prácticas de desarrollo ágil y un sistema robusto de control de versiones para gestionar eficazmente los cambios y mantener la colaboración entre el equipo de desarrollo.

Otros Escenarios

El atributo de calidad de Performance en "LinkEase" se refiere a la capacidad del sistema para responder de manera rápida y eficiente, tanto en términos de tiempo de respuesta para eventos específicos como en la cantidad de eventos procesados en un intervalo de tiempo dado. Este aspecto es crucial para garantizar una experiencia de usuario fluida y satisfactoria, especialmente bajo cargas de trabajo variables y picos de actividad.

Criterios de Aceptación:

a. Tiempo de Respuesta:

El tiempo promedio para acortar una URL debe ser inferior a 200 milisegundos.

La redirección desde un enlace acortado al URL original debe ocurrir en menos de 100 milisegundos.

El tiempo de carga de la interfaz de usuario principal no debe superar los 3 segundos en condiciones normales de carga.

b. Capacidad de Procesamiento:

El sistema debe poder manejar al menos 1000 solicitudes de acortamiento de URLs por minuto sin experimentar una degradación significativa del rendimiento.

La capacidad de escalabilidad horizontal debe permitir aumentar la capacidad de procesamiento según sea necesario para cumplir con demandas pico de tráfico.

c. Pruebas de Carga:

Se realizarán pruebas de carga periódicas para simular condiciones de alta demanda y verificar la capacidad de rendimiento del sistema.

Durante las pruebas, se monitorizará y registrará el comportamiento del sistema para identificar cuellos de botella y áreas de mejora potencial.

d. Optimización Continua:



Se implementarán técnicas de optimización de código y configuración de infraestructura para mejorar continuamente el rendimiento del sistema.

Se utilizará monitoreo en tiempo real para ajustar recursos según la carga de trabajo y las necesidades de rendimiento del sistema.