
Documento de Arquitectura de Software (DAS)

Proyecto

Talklate

Arquitectos

Julian David Diaz Medina

Guillermo Andrés Gil Rozo

Diego Alejandro Franco
Atehortua

Contenido

1.	Propósito del proyecto	7
2.	Motivadores de la arquitectura	7
2.1	Restricciones técnicas	7
2.2	Restricciones de negocio	8
2.3	Atributos de calidad	8
2.3.1	Atributo calidad 1: Seguridad	9
2.3.1.1	Característica 1: Brindar seguridad a las contraseñas	9
2.3.1.1.1	Escenario de calidad 1	9
2.3.1.1.2	Escenario de calidad 2	9
2.3.1.1.3	Escenario de calidad 3	10
2.3.1.1.4	Escenario de calidad 4	10
2.3.1.1.5	Escenario de calidad 5	10
2.3.1.2	Característica 2: Autenticación en 2 pasos	11
2.3.1.2.1	Escenario de calidad 1	11
2.3.1.2.2	Escenario de calidad 2	11
2.3.1.2.3	Escenario de calidad 3	12
2.3.1.2.4	Escenario de calidad 4	12
2.3.1.3	Característica 3: Separación de roles	12
2.3.1.3.1	Escenario de calidad 1	13
2.3.1.3.2	Escenario de calidad 2	13
2.3.1.3.3	Escenario de calidad 3	13
2.3.1.3.4	Escenario de calidad 2	14
2.3.1.3.5	Escenario de calidad 4	14
2.3.1.3.6	Escenario de calidad 5	14
2.3.1.4	Característica 4: Ingreso de terceros	15
2.3.1.4.1	Escenario de calidad 1	15
2.3.1.4.2	Escenario de calidad 3	15
2.3.1.5	Característica 5: Restablecer contraseña	15
2.3.1.5.1	Escenario de calidad 1	16
2.3.1.5.2	Escenario de calidad 2	16
2.3.1.5.3	Escenario de calidad 3	16
2.3.2	Atributo calidad 2: Portabilidad	17

2.3.2.1	Característica 1: Disponible en móviles	17
2.3.2.1.1	Escenario de calidad 1	17
2.3.2.1.2	Escenario de calidad 2	17
2.3.2.2	Característica 2: Disponible en web	18
2.3.2.2.1	Escenario de calidad 1	18
2.3.2.2.2	Escenario de calidad 2	18
2.3.3	Atributo calidad 3: Conformidad (Compliance)	18
2.3.3.1	Característica 1: Tutorial	19
2.3.3.1.1	Escenario de calidad 1	19
2.3.3.1.2	Escenario de calidad 2	19
2.3.3.1.3	Escenario de calidad 3	20
2.3.3.1.4	Escenario de calidad 4	20
2.3.4	Atributo calidad 4: Disponibilidad	20
2.3.4.1	Característica 1: Capacidad de usuarios	21
2.3.4.1.1	Escenario de calidad 1	21
2.3.4.1.2	Escenario de calidad 2	21
2.3.4.1.3	Escenario de calidad 3	21
2.3.5	Atributo calidad 5: Flexibilidad	22
2.3.5.1	Característica 1: Durabilidad de la sesión de usuario	22
2.3.5.1.1	Escenario de calidad 1	22
2.3.5.1.2	Escenario de calidad 2	22
2.4	Funcionalidades críticas	23
2.4.1	Funcionalidad 1	23
2.4.1.1	Justificación	23
2.4.2	Funcionalidad 2	23
2.4.2.1	Justificación	23
2.4.3	Funcionalidad 3	23
2.4.3.1	Justificación	23
2.4.4	Funcionalidad 4	23
2.4.4.1	Justificación	23
2.4.5	Funcionalidad 5	24
2.4.5.1	Justificación	24
2.4.6	Funcionalidad 6	24

2.4.6.1	Justificación	24
2.4.7	Funcionalidad 7	24
2.4.7.1	Justificación	24
3.	Tácticas y estrategias	24
4.	Modelo de contexto	25
5.	Arquetipo de solución/referencia	26
6.	Arquitectura de solución/referencia.....	27
6.1	Documentación	27
7.	Línea base arquitectónica	28
7.1	Línea base arquitectónica General.....	28
7.2	Estilos y patrones arquitectónicos adoptados	28
7.2.1	Estilo arquitectónico 1.....	28
7.2.1.1	Nombre	28
7.2.1.2	Problema	28
7.2.1.3	Solución/Motivación	29
7.2.2	Estilo arquitectónico 2.....	29
7.2.2.1	Nombre	29
7.2.2.2	Problema	29
7.2.2.3	Solución/Motivación	29
7.2.3	Estilo arquitectónico 3.....	29
7.2.3.1	Nombre	29
7.2.3.2	Problema	29
7.2.3.3	Solución/Motivación	29
8.	Justificación alternativa de solución	30
8.1	Justificación	30
8.2	Ventajas.....	30
8.3	Desventajas	30
9.	Vistas de arquitectura del sistema	31
9.1	Vista Funcional/Vista de Escenarios/Vista de Casos de Uso	31
9.1.1	Modelo de procesos del negocio	31
9.1.2	Glosario de términos del negocio	31
9.1.3	Especificación de requisitos de software	32
9.1.3.1	Requisitos de usuario	33

9.1.3.1.1	Diagrama de actividades	34
9.1.3.1.1.1	Diagrama 1	35
9.1.3.1.1.1.1.1	Diagrama 2	36
9.1.3.1.1.1.1.2	Diagrama 3	37
9.1.3.1.1.1.1.3	Diagrama 4	38
9.1.3.1.1.1.1.4	Diagrama 5	39
9.1.3.1.1.1.1.5	Diagrama 6	40
9.1.3.1.1.1.1.6	Diagrama 7	40
9.1.3.1.1.1.1.7	Documentación	41
9.1.4	Incepción Ágil	41
9.1.4.1	Por qué estamos aquí.....	41
9.1.4.2	Visión/Elevator Pitch	41
9.1.4.2.1	Visión	42
9.1.4.2.2	Project Canvas	42
9.1.4.2.3	Mapa de impacto	42
9.1.4.3	Caja de producto	43
9.1.4.4	Lo que sí, lo que no	44
9.1.4.5	Mapa de historias de usuario	45
9.1.4.6	Product Backlog Item	46
9.1.4.7	La comunidad	46
9.1.4.8	La solución.....	47
9.2	Vista Lógica.....	48
9.2.1	Diagrama de clases.....	48
9.2.1.1	Componente 1.....	48
9.2.1.1.1	Diagrama	48
9.2.1.2	Componente 2.....	49
9.2.1.2.1	Diagrama	49
9.2.1.3	Componente 3.....	49
9.2.1.3.1	Diagrama	50
9.2.1.4	Componente 4.....	50
9.2.1.4.1	Diagrama	50
9.2.1.5	Componente 5.....	51
9.2.1.5.1	Diagrama	51

9.2.1.6	Componente 6.....	51
9.2.1.6.1	Diagrama	52
9.2.1.7	Componente 7.....	52
9.2.1.7.1	Diagrama	53
9.2.1.8	Componente 8.....	53
10.1.1.1.1	Diagrama	54
10.1.1.2	Componente 9.....	54
10.1.1.2.1	Diagrama	54
10.2	Vista de Despliegue/Vista de Desarrollo/Vista de Implementación	54
10.2.1	Diagrama de componentes	55
10.2.1.1	Componente 1.....	55
10.2.1.1.1	Diagrama	55
10.2.1.1.2	Documentación	55
10.2.1.2	Componente 2.....	55
10.2.1.2.1	Diagrama	56
10.2.1.2.2	Documentación	56
10.2.2	Diagrama de paquetes	56
10.2.2.1	Componente 1.....	57
10.2.2.1.1	Diagrama	57
10.2.2.1.2	Documentación	58
10.2.2.2	Componente 2.....	58
10.2.2.2.1	Diagrama	59
10.2.2.2.2	Documentación	59
10.3	Vista de Procesos	60
10.3.1	Diagrama de secuencia	60
10.3.1.1	Componente general.....	60
10.3.1.1.1	Diagrama	60
10.3.1.1.2	Documentación	61
10.3.2	Diagrama de colaboración	61
10.3.2.1	Componente 1.....	61
10.3.2.1.1	Diagrama	61
10.3.2.1.2	Documentación	62
10.4	Vista Física/Vista de Implantación	62

1. Propósito del proyecto

Para Cualquier persona (especialmente foráneos) que necesite comunicarse con otras personas en idiomas que no conozca o domine, TalkLate es un una aplicación web/móvil que brindará a sus usuarios la oportunidad de comunicarse con personas en el lugar donde se encuentren y que no hablen su lengua, sea por turismo, trabajo u otras razones; Esto será posible gracias a hablantes nativos que desean ofrecer sus servicios como traductores presenciales, permitiendo así que los usuarios puedan desenvolverse con más facilidad en un entorno cultural y/o espacial probablemente desconocido. A diferencia de Google Translator, Upwork, entre otras aplicaciones para traducir, nuestro producto se basa en traductores nativos del lugar, lo que implica que la traducción no solo será precisa desde el punto de vista lingüístico, sino que también estará impregnada de la autenticidad cultural y los matices que solo los hablantes nativos pueden ofrecer.

2. Motivadores de la arquitectura

elemento o factor clave que impulsa las decisiones y el diseño arquitectónico de un sistema, ya sea un sistema de software, un edificio, un proyecto de ingeniería o cualquier otro tipo de arquitectura. Estos "drivers" son consideraciones fundamentales que influyen en cómo se estructura y se desarrolla el sistema o proyecto arquitectónico. Los drivers arquitectónicos son esenciales para garantizar que la arquitectura cumpla con los objetivos y requisitos clave.

2.1 Restricciones técnicas

se refiere a cualquier limitación o condición que influye en cómo se diseña, desarrolla y opera un sistema de software. Estas restricciones son factores que deben ser considerados durante el proceso de desarrollo de software para asegurarse de que el producto final cumple con los requisitos y necesidades del negocio.

Tipo	Restricción	Justificación
Lineamiento	La versión móvil de la aplicación debe correr en sistemas operativos iOS 16 y Android 8 en adelante	Debe seguirse el lineamiento ya que es una decisión que tiene por sentado el cliente
Prácticas de diseño	El desarrollo de la arquitectura debe seguir principios de DDD	Se debe seguir la técnica de Diseño Dirigido por el Dominio ya que permitirá crear un software de alta calidad que se ajuste a los requisitos del negocio y que además permitirá que el producto sea fácil de mantener y modificar.
	El desarrollo de la arquitectura debe seguir el principio KISS	Se debe seguir el principio KISS porque favorecerá a un diseño organizado y de fácil manejo
	Se debe desarrollar la aplicación siguiendo una arquitectura en capas	El uso de una arquitectura en capas facilitará la escalabilidad y el mantenimiento del producto
	La arquitectura de la aplicación debe ser orientada a microservicios	Al actuar como servicios pequeños y autónomos, los microservicios brindan una mayor escalabilidad y desarrollo ágil, además facilitan su recuperación en caso de fallos, mejorando así la disponibilidad y resiliencia
Prácticas de desarrollo	Se debe gestionar y controlar las versiones del código fuente de la aplicación	Al gestionar y versionar el código se facilita la reversión de cambios, la realización de pruebas, la colaboración de equipo entre otras ventajas
	Se deben realizar pruebas de estrés y rendimiento	Realizar pruebas de estrés y rendimiento garantizará la disponibilidad y rendimiento de la aplicación sometiéndola a ambientes de alta carga o a condiciones adversas
	El desarrollo de la arquitectura debe seguir el principio DRY	Se debe seguir el principio DRY ya que este ayuda a facilitar el mantenimiento y comprensión del código evitando la aparición de código duplicado
	El desarrollo de la arquitectura debe seguir principios SOLID	Los principios SOLID brindan buenas prácticas ayudando a un desarrollo más limpio, mantenible y escalable
Patrones de diseño	Se utilizarán patrones de diseño creacionales	El uso de patrones de diseño creacionales proporcionan varios mecanismos de creación de objetos que incrementan la flexibilidad y la reutilización del código existente.
	Se utilizarán patrones de diseño estructurales	Los patrones de diseño estructurales explican como ensamblar objetos y clases en estructuras más grandes, a la vez que se mantiene la flexibilidad y eficiencia de estas estructuras
Patrones de DevOps	Se adoptarán prácticas de integración y entrega continua	Mediante la integración y entrega continua se mejora la calidad del código y se acelera el proceso de desarrollo

2.2 Restricciones de negocio

En el contexto del desarrollo de software, las restricciones de negocio se refieren a las limitaciones o condiciones que influyen en cómo se planifica, diseña y desarrolla un sistema de software. Estas restricciones son impuestas por las necesidades y los requisitos específicos de la empresa o la organización que está impulsando el proyecto de software.

Tipo	Restricción	Justificación
Humanas	Los desarrolladores no podrán trabajar después de las 4pm	Esto se debe a que los desarrolladores en su gran mayoría son estudiantes de la universidad y después de las 4 deben asistir a clases
	El arquitecto solo podrá trabajar 6 horas al día	El arquitecto al estar involucrado en otros proyectos, solo podrá trabajar 6 horas por la mañana
	No se podrá hacer reuniones con el cliente de la aplicación por más de 5 horas semanales	Debe seguirse esta restricción ya que es una decisión que tiene por sentado el cliente
Financieras	El presupuesto del proyecto será de 30 millones de pesos colombianos	El cliente premeditadamente aclaró que el dinero que se invertirá al proyecto será lo mínimo posible, por ende el precio será de 30 millones
Tiempo	La aplicación debe entregarse con un plazo máximo de 7 meses	El cliente aclaró que la fecha de entrega del proyecto tiene que ser 7 meses una vez iniciado el desarrollo
Procesos	A la fecha existen funcionalidades que no son claras	Esta restricción puede deberse a la evolución constante del software, su complejidad y errores.
Negocio	La información del desarrollo no deberá salir fuera del entorno de trabajo	Debe seguirse esta restricción de negocio porque es una decisión premeditadamente tomada por el cliente

2.3 Atributos de calidad

son las características específicas que se utilizan para evaluar y medir la calidad de un producto, sistema, servicio o proceso. Estos atributos son fundamentales para determinar si un producto o sistema cumple con los requisitos y expectativas establecidos.

2.3.1 Atributo calidad 1: Seguridad

La seguridad se refiere a la capacidad de un sistema o proyecto para proteger la confidencialidad, integridad y disponibilidad de los datos y recursos. Implica prevenir el acceso no autorizado, la modificación no autorizada y la pérdida de datos críticos. La seguridad es esencial en el desarrollo del proyecto para garantizar que la información y los activos críticos estén protegidos contra amenazas y riesgos. Esto implica la implementación de medidas de seguridad, como autenticación, autorización, cifrado y protección contra ataques cibernéticos además la seguridad se integra a través de decisiones y características específicas. Esto puede incluir la arquitectura de seguridad, la gestión de identidades y accesos, la encriptación de datos, la detección de intrusiones y la auditoría de seguridad. El diseño debe considerar cómo se aplicarán estas medidas para garantizar que el sistema sea resistente a amenazas y vulnerabilidades conocidas.

2.3.1.1 Característica 1: Brindar seguridad a las contraseñas

Para brindar más seguridad a las cuentas de los usuarios la aplicación requerirá que las contraseñas sigan un patrón de formación

2.3.1.1.1 Escenario de calidad 1

Identificador	ESC-CAL-CAR-SEG-08-01
Característica	CAR-SEG-08
Taxonomía	Autorización
Atributo de calidad	Seguridad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación habilita funcionalidades a las que los traductores tienen acceso cuando un traductor ingresa y usa la aplicación
Origen del estímulo	Traductor
Estímulo	Ingresar y usar la aplicación
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El sistema habilita las funcionalidades a las que el traductor tiene acceso
Medida de la respuesta	Cuando un traductor use la aplicación, el sistema se encargará de habilitar funcionalidades a las que el traductor tiene acceso

2.3.1.1.2 Escenario de calidad 2

Identificador	ESC-CAL-CAR-SEG-08-02
Característica	CAR-SEG-08
Taxonomía	Autorización
Atributo de calidad	Seguridad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación habilita funcionalidades a las que los clientes tiene acceso cuando un cliente ingresa y usa la aplicación
Origen del estímulo	Cliente
Estímulo	ingresa y usa la aplicación
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El sistema habilita las funcionalidades a las que el cliente tiene acceso
Medida de la respuesta	Cuando un cliente use la aplicación, el sistema se encargará de habilitar funcionalidades a las que el cliente tiene acceso

2.3.1.1.3 Escenario de calidad 3

Identificador	ESC-CAL-CAR-SEG-08-03
Característica	CAR-SEG-08
Taxonomía	Autorización
Atributo de calidad	Seguridad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación habilita funcionalidades a las que los administradores tienen acceso cuando un administrador ingresa y usa la aplicación
Origen del estímulo	Administrador
Estímulo	ingresa y usa la aplicación
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El sistema habilita las funcionalidades a las que el administrador tiene acceso
Medida de la respuesta	Cuando un administrador use la aplicación, el sistema se encargará de habilitar funcionalidades a las que el administrador tiene acceso

2.3.1.1.4 Escenario de calidad 4

Identificador	ESC-CAL-CAR-SEG-08-04
Característica	CAR-SEG-08
Taxonomía	Autorización
Atributo de calidad	Seguridad
Tipo escenario	No exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación no mostrará funcionalidades habilitadas a la perfilación correspondiente
Origen del estímulo	Cualquier usuario final
Estímulo	ingresa y usa la aplicación
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El sistema no habilita funcionalidades exclusivas para el usuario
Medida de la respuesta	Cuando un usuario use la aplicación, el sistema no se encargará de habilitar funcionalidades exclusivas de acuerdo a la perfilación del usuario

2.3.1.1.5 Escenario de calidad 5

Identificador	ESC-CAL-CAR-SEG-08-05
Característica	CAR-SEG-08
Taxonomía	Autorización
Atributo de calidad	Seguridad
Tipo escenario	No exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación mostrará funcionalidades habilitadas a la perfilación no correspondiente
Origen del estímulo	Cualquier usuario final
Estímulo	ingresa y usa la aplicación
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El sistema habilita funcionalidades para el perfil que no correspondía
Medida de la respuesta	Cuando un usuario use la aplicación, el sistema se encargará de habilitar funcionalidades exclusivas erróneamente para perfilación del usuario

2.3.1.2 Característica 2: Autenticación en 2 pasos

Para brindar más seguridad al usuario final la aplicación enviará un código de seguridad mediante correo electrónico, el cual se le solicitará para permitirle el ingreso a la aplicación

2.3.1.2.1 Escenario de calidad 1

Identificador	ESC-CAL-CAR-SEG-06-01
Característica	CAR-SEG-06
Taxonomía	Autenticación
Atributo de calidad	Seguridad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación permita al usuario activar una autenticación de 2 pasos que funcione correctamente
Origen del estímulo	Cualquier usuario final
Estímulo	Activar la autenticación en dos pasos
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	Aparece un mensaje que confirma la activación
Medida de la respuesta	Luego del usuario ingresar a las configuraciones, el usuario activa la autenticación y se muestra un mensaje de confirmación

2.3.1.2.2 Escenario de calidad 2

Identificador	ESC-CAL-CAR-SEG-06-02
Característica	CAR-SEG-06
Taxonomía	Autenticación
Atributo de calidad	Seguridad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación permita al usuario desactivar una autenticación de 2 pasos que funcione correctamente
Origen del estímulo	Cualquier usuario final
Estímulo	desactivar la autenticación en dos pasos
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	Aparece un mensaje que confirma la desactivación
Medida de la respuesta	Luego del usuario ingresar a las configuraciones, el usuario desactiva la autenticación y se muestra un mensaje de confirmación

2.3.1.2.3 Escenario de calidad 3

Identificador	ESC-CAL-CAR-SEG-06-03
Característica	CAR-SEG-06
Taxonomía	Autenticación
Atributo de calidad	Seguridad
Tipo escenario	No exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación no permita al usuario activar una autenticación de 2 pasos
Origen del estímulo	Cualquier usuario final
Estímulo	Activar la autenticación en dos pasos
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	No aparece ningún mensaje de activación
Medida de la respuesta	Luego del usuario ingresar a las configuraciones, el usuario intenta activar la autenticación de dos pasos pero no funciona

2.3.1.2.4 Escenario de calidad 4

Identificador	ESC-CAL-CAR-SEG-06-04
Característica	CAR-SEG-06
Taxonomía	Autenticación
Atributo de calidad	Seguridad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación no permita al usuario desactivar una autenticación de 2 pasos que funcione correctamente
Origen del estímulo	Cualquier usuario final
Estímulo	desactivar la autenticación en dos pasos
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	No aparece ningún mensaje de desactivación
Medida de la respuesta	Luego del usuario ingresar a las configuraciones, el usuario intenta desactivar la autenticación de dos pasos pero no funciona

2.3.1.3 Característica 3: Separación de roles

Para evitar que una persona realice acciones en la aplicación sin los permisos necesarios se habilitarán acciones en función del trabajo o posición de la persona

2.3.1.3.1 Escenario de calidad 1

Identificador	ESC-CAL-CAR-SEG-08-01
Característica	CAR-SEG-08
Taxonomía	Autorización
Atributo de calidad	Seguridad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación habilita funcionalidades a las que los traductores tienen acceso cuando un traductor ingresa y usa la aplicación
Origen del estímulo	Traductor
Estímulo	Ingresar y usar la aplicación
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El sistema habilita las funcionalidades a las que el traductor tiene acceso
Medida de la respuesta	Cuando un traductor use la aplicación, el sistema se encargará de habilitar funcionalidades a las que el traductor tiene acceso

2.3.1.3.2 Escenario de calidad 2

Identificador	ESC-CAL-CAR-SEG-08-02
Característica	CAR-SEG-08
Taxonomía	Autorización
Atributo de calidad	Seguridad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación habilita funcionalidades a las que los clientes tiene acceso cuando un cliente ingresa y usa la aplicación
Origen del estímulo	Cliente
Estímulo	ingresa y usa la aplicación
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El sistema habilita las funcionalidades a las que el cliente tiene acceso
Medida de la respuesta	Cuando un cliente use la aplicación, el sistema se encargará de habilitar funcionalidades a las que el cliente tiene acceso

2.3.1.3.3 Escenario de calidad 3

Identificador	ESC-CAL-CAR-SEG-08-03
Característica	CAR-SEG-08
Taxonomía	Autorización
Atributo de calidad	Seguridad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación habilita funcionalidades a las que los administradores tienen acceso cuando un administrador ingresa y usa la aplicación
Origen del estímulo	Administrador
Estímulo	ingresa y usa la aplicación
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El sistema habilita las funcionalidades a las que el administrador tiene acceso
Medida de la respuesta	Cuando un administrador use la aplicación, el sistema se encargará de habilitar funcionalidades a las que el administrador tiene acceso

2.3.1.3.4 Escenario de calidad 2

Identificador	ESC-CAL-CAR-SEG-09-02
Característica	CAR-SEG-09
Taxonomía	Autenticación
Atributo de calidad	Seguridad
Tipo escenario	No exitoso
¿Escenario cumplido?	No
Descripción	Asegura que el usuario no pueda ingresar a la aplicación usando una cuenta de terceros como por ejemplo Facebook, Gmail o LinkedIn de la cual el usuario disponga
Origen del estímulo	Cualquier usuario final
Estímulo	Ingresar a la aplicación usando una cuenta de terceros que posea
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El sistema redirige al usuario final de nuevo a la pantalla de inicio de sesión
Medida de la respuesta	Luego de que el usuario final ingrese a la aplicación, y para iniciar sesión seleccione una cuenta de terceros que posea, es redirigido a la pantalla de inicio de sesión

2.3.1.3.5 Escenario de calidad 4

Identificador	ESC-CAL-CAR-SEG-08-04
Característica	CAR-SEG-08
Taxonomía	Autorización
Atributo de calidad	Seguridad
Tipo escenario	No exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación no mostrará funcionalidades habilitadas a la perfilación correspondiente
Origen del estímulo	Cualquier usuario final
Estímulo	ingresa y usa la aplicación
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El sistema no habilita funcionalidades exclusivas para el usuario
Medida de la respuesta	Cuando un usuario use la aplicación, el sistema no se encargará de habilitar funcionalidades exclusivas de acuerdo a la perfilación del usuario

2.3.1.3.6 Escenario de calidad 5

Identificador	ESC-CAL-CAR-SEG-08-05
Característica	CAR-SEG-08
Taxonomía	Autorización
Atributo de calidad	Seguridad
Tipo escenario	No exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación mostrará funcionalidades habilitadas a la perfilación no correspondiente
Origen del estímulo	Cualquier usuario final
Estímulo	ingresa y usa la aplicación
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El sistema habilita funcionalidades para el perfil que no correspondía
Medida de la respuesta	Cuando un usuario use la aplicación, el sistema se encargará de habilitar funcionalidades exclusivas erróneamente para perfilación del usuario

2.3.1.4 Característica 4: Ingreso de terceros

Para que el usuario pueda iniciar sesión, la aplicación le permitirá ingresar con cuentas de terceros como Facebook, Gmail o LinkedIn

2.3.1.4.1 Escenario de calidad 1

Identificador	ESC-CAL-CAR-SEG-09-01
Característica	CAR-SEG-09
Taxonomía	Autenticación
Atributo de calidad	Seguridad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que el usuario pueda ingresar a la aplicación usando una cuenta de terceros como por ejemplo Facebook, Gmail o LinkedIn, de la cual el usuario disponga
Origen del estímulo	Cualquier usuario final
Estímulo	Ingresar a la aplicación usando una cuenta de terceros que posea
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El usuario es redirigido a la pantalla principal de la aplicación
Medida de la respuesta	Luego de que el usuario final ingrese a la aplicación y para iniciar sesión seleccione una cuenta de terceros que posea, es redirigido a la pantalla principal

2.3.1.4.2 Escenario de calidad 3

Identificador	ESC-CAL-CAR-SEG-09-03
Característica	CAR-SEG-09
Taxonomía	Autenticación
Atributo de calidad	Seguridad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que el usuario no pueda ingresar a la aplicación usando una cuenta de terceros que este no posea como por ejemplo Facebook, Gmail o LinkedIn
Origen del estímulo	Cualquier usuario final
Estímulo	Ingresar a la aplicación usando una cuenta de terceros que este no posee
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El sistema redirecciona al usuario final de nuevo a la pantalla de inicio de sesión
Medida de la respuesta	Al iniciar sesión en la aplicación usando cuentas de terceros que no posea, es redirigido a la pantalla de inicio de sesión

2.3.1.5 Característica 5: Restablecer contraseña

Para ayudar en la autenticación de la aplicación, se permitirá restablecer la contraseña de inicio de sesión mediante un correo electrónico

2.3.1.5.1 Escenario de calidad 1

Identificador	ESC-CAL-CAR-SEG-10-01
Característica	CAR-SEG-10
Taxonomía	Autenticación
Atributo de calidad	Seguridad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación permita a los usuarios poder restablecer su contraseña por medio de un correo electrónico
Origen del estímulo	Cualquier usuario final
Estímulo	Hacer clic en restablecer contraseña
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	La aplicación le envía el correo al usuario final para restablecer la contraseña
Medida de la respuesta	Cuando el usuario final haga clic en restablecer contraseña se le enviará un correo electrónico mediante el cual podrá restablecer la contraseña

2.3.1.5.2 Escenario de calidad 2

Identificador	ESC-CAL-CAR-SEG-10-02
Característica	CAR-SEG-10
Taxonomía	Autenticación
Atributo de calidad	Seguridad
Tipo escenario	No exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación no permita a los usuarios poder restablecer su contraseña por medio de un correo electrónico
Origen del estímulo	Cualquier usuario final
Estímulo	Hacer clic en restablecer contraseña
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	La aplicación no le envía el correo al usuario final para restablecer la contraseña
Medida de la respuesta	Cuando el usuario final haga clic en restablecer contraseña este no le enviará un correo electrónico mediante el cual podrá restablecer la contraseña

2.3.1.5.3 Escenario de calidad 3

Identificador	ESC-CAL-CAR-SEG-10-03
Característica	CAR-SEG-10
Taxonomía	Autenticación
Atributo de calidad	Seguridad
Tipo escenario	No exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación permita a los usuarios poder restablecer su contraseña por medio de un correo electrónico
Origen del estímulo	Cualquier usuario final
Estímulo	Hacer clic en restablecer contraseña
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	La aplicación le envía el correo al usuario final para restablecer la contraseña pero este no funciona
Medida de la respuesta	Cuando el usuario final haga clic en restablecer contraseña este le enviará un correo electrónico mediante el cual podrá restablecer la contraseña, pero este no funciona

2.3.2 Atributo calidad 2: Portabilidad

capacidad de un sistema o proyecto para funcionar de manera efectiva en diferentes entornos, dispositivos o plataformas. Implica la flexibilidad para mover o adaptar el sistema sin que se vea afectado su rendimiento o funcionalidad. La portabilidad es fundamental en el desarrollo del proyecto para garantizar que el sistema sea versátil y se pueda desplegar en una variedad de entornos. Esto implica considerar la independencia de la plataforma, la capacidad de migración y la compatibilidad con diferentes sistemas operativos y dispositivos, además, En la definición del diseño, la portabilidad se logra mediante la creación de una arquitectura y una implementación que no estén atadas a una plataforma específica. Esto implica el uso de estándares abiertos, la modularidad y la adaptabilidad del código para facilitar la portabilidad. Además, se deben considerar aspectos como la facilidad de configuración y la gestión de dependencias.

2.3.2.1 Característica 1: Disponible en móviles

Para facilitar el acceso a la aplicación, esta estará disponible en dispositivos móviles

2.3.2.1.1 Escenario de calidad 1

Identificador	ESC-CAL-CAR-POR-05-01
Característica	CAR-POR-05
Taxonomía	Dispositivos compatibles
Atributo de calidad	Portabilidad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación estará disponible en dispositivos móviles
Origen del estímulo	Cualquier usuario final
Estímulo	Ingrese al a la aplicación desde un dispositivo móvil
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	La aplicación funcionará
Medida de la respuesta	Al ingresar a la aplicación desde un dispositivo móvil, esta funcionará correctamente y enviará al usuario a la página principal

2.3.2.1.2 Escenario de calidad 2

Identificador	ESC-CAL-CAR-POR-05-02
Característica	CAR-POR-05
Taxonomía	Dispositivos compatibles
Atributo de calidad	Portabilidad
Tipo escenario	No exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación no estará disponible en dispositivos móviles
Origen del estímulo	Cualquier usuario final
Estímulo	Ingrese al a la aplicación desde un dispositivo móvil
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	La aplicación no funcionara desde dispositivos móviles
Medida de la respuesta	Al ingresar a la aplicación desde un dispositivo móvil, esta no funcionará y no será posible su uso

2.3.2.2 Característica 2: Disponible en web

Para facilitar el acceso a la aplicación, esta estará disponible en plataformas WEB

2.3.2.2.1 Escenario de calidad 1

Identificador	ESC-CAL-CAR-POR-04-01
Característica	CAR-POR-04
Taxonomía	Dispositivos compatibles
Atributo de calidad	Portabilidad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación estará disponible como un sitio web
Origen del estímulo	Cualquier usuario final
Estímulo	Ingresa al sitio web de la aplicación
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	La aplicación funcionara de manera correcta desde su sitio web
Medida de la respuesta	Al ingresar a la página web, esta se encontraría disponible para su uso y será redireccionado hacia la página principal

2.3.2.2.2 Escenario de calidad 2

Identificador	ESC-CAL-CAR-POR-04-02
Característica	CAR-POR-04
Taxonomía	Dispositivos compatibles
Atributo de calidad	Portabilidad
Tipo escenario	No exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación no estará disponible desde un sitio web
Origen del estímulo	Cualquier usuario final
Estímulo	Ingresar al sitio web de la aplicación
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	La aplicación no funcionara desde un sitio web
Medida de la respuesta	Al ingresar a la página web, esta no se encontrara disponible para su uso

2.3.3 Atributo calidad 3: Conformidad (Compliance)

capacidad de un sistema o proyecto para cumplir con regulaciones, estándares y políticas específicas. Esto implica adherirse a las normativas legales y de la industria, así como a las políticas internas de una organización. La conformidad es crucial en el desarrollo del proyecto para garantizar que el sistema cumpla con todos los requisitos legales y reglamentarios aplicables. Esto puede incluir leyes de privacidad, regulaciones de seguridad de datos, estándares de la industria y políticas internas de la organización, además, En la definición del diseño, la conformidad se logra mediante la incorporación de medidas y controles que aseguren el cumplimiento de las regulaciones y políticas relevantes. Esto puede incluir la implementación de auditorías, controles de acceso, cifrado de datos y políticas de retención de datos.

2.3.3.1 Característica 1: Tutorial

Para que un traductor pueda usar la aplicación deberá completar el tutorial

2.3.3.1.1 Escenario de calidad 1

Identificador	ESC-CAL-CAR-CON-01-01
Característica	CAR-CON-01
Taxonomía	Capacitación y orientación
Atributo de calidad	Conformidad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que el sistema muestra un tutorial exponiendo las funcionalidades básicas de la aplicación a los traductores que ingresan por primera vez
Origen del estímulo	Traductor
Estímulo	Ingresar a la aplicación por primera vez
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El sistema muestra un tutorial que expone las funcionalidades básicas de la aplicación
Medida de la respuesta	Cuando el traductor ingresa por primera vez al sistema, se le muestra un tutorial que expone las funcionalidades básicas de la aplicación, y posteriormente es redirigido a la pantalla principal

2.3.3.1.2 Escenario de calidad 2

Identificador	ESC-CAL-CAR-CON-01-02
Característica	CAR-CON-0006
Taxonomía	Autenticación
Atributo de calidad	Seguridad
Tipo escenario	No exitoso
¿Escenario cumplido?	No
Descripción	Asegura que el sistema no muestra un tutorial exponiendo las funcionalidades básicas de la aplicación a los traductores que ingresan por primera vez
Origen del estímulo	Traductor
Estímulo	Ingresar a la aplicación posteriormente a la primera vez
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El sistema redirige al traductor a la pantalla principal
Medida de la respuesta	Luego de que el traductor inicie sesión, posterior a la primera vez, el sistema redirige al traductor a la pantalla principal

2.3.3.1.3 Escenario de calidad 3

Identificador	ESC-CAL-CAR-CON-01-03
Característica	CAR-CON-0006
Taxonomía	Autenticación
Atributo de calidad	Seguridad
Tipo escenario	No Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que el sistema muestra un tutorial exponiendo las funcionalidades básicas de la aplicación a los traductores que no ingresan por primera vez
Origen del estímulo	Traductor
Estímulo	Ingresar a la aplicación posteriormente a la primera vez
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El sistema muestra un tutorial que expone las funcionalidades básicas de la aplicación
Medida de la respuesta	Posterior a que el traductor ingresa por primera vez al sistema, se le muestra un tutorial que expone las funcionalidades básicas de la aplicación, y posteriormente es redirigido a la pantalla principal

2.3.3.1.4 Escenario de calidad 4

Identificador	ESC-CAL-CAR-CON-01-04
Característica	CAR-CON-01
Taxonomía	Capacitación y orientación
Atributo de calidad	Conformidad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que el sistema no muestra un tutorial exponiendo las funcionalidades básicas de la aplicación a los traductores que no ingresan por primera vez
Origen del estímulo	Traductor
Estímulo	no ingresar a la aplicación por primera vez
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El sistema no muestra el tutorial a traductores que no ingresan por primera vez
Medida de la respuesta	Cuando el traductor ingresa posterior a la primera vez a la aplicación, no se muestra un tutorial que expone las funcionalidades básicas de la aplicación, y posteriormente es redirigido a la pantalla principal

2.3.4 Atributo calidad 4: Disponibilidad

capacidad de un sistema o proyecto para estar accesible y funcionando correctamente cuando los usuarios lo necesitan. Implica minimizar el tiempo de inactividad y garantizar un alto grado de acceso constante. La disponibilidad es esencial en el desarrollo del proyecto para garantizar que los usuarios puedan acceder al sistema en todo momento. Esto implica la implementación de redundancia, recuperación ante desastres y medidas de resistencia a fallos para minimizar el tiempo de inactividad no planificado, además, En la definición del diseño, la disponibilidad se logra mediante la planificación de la infraestructura y la arquitectura del sistema de manera que se minimice el impacto de las interrupciones. Esto incluye la implementación de copias de seguridad regulares, sistemas de equilibrio de carga, redundancia de servidores y procedimientos de recuperación.

2.3.4.1 Característica 1: Capacidad de usuarios

La aplicación funcionará sin interrupciones sin importar cuantas personas la usan al mismo tiempo

2.3.4.1.1 Escenario de calidad 1

Identificador	ESC-CAL-CAR-DIS-02-01
Característica	CAR-DIS-02
Taxonomía	Operaciones simultaneas
Atributo de calidad	Disponibilidad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación funcione correctamente cuando se tenga un alto flujo de usuarios
Origen del estímulo	Cualquier usuario final
Estímulo	Usar la aplicación
Artefacto	Sistema
Ambiente	Operación con alto flujo de usuarios
Respuesta	El sistema se mantiene con normalidad y no presenta problemas
Medida de la respuesta	Luego de que muchos usuarios ingresen a la aplicación, esta se mantiene funcionando correctamente

2.3.4.1.2 Escenario de calidad 2

Identificador	ESC-CAL-CAR-DIS-02-02
Característica	CAR-DIS-02
Taxonomía	Operaciones simultaneas
Atributo de calidad	Disponibilidad
Tipo escenario	No exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación no funcione correctamente cuando se tenga un alto flujo de usuarios
Origen del estímulo	Cualquier usuario final
Estímulo	Usar la aplicación
Artefacto	Sistema
Ambiente	Operación con alto flujo de usuarios
Respuesta	El sistema se cae y deja de funcionar por el alto flujo
Medida de la respuesta	Luego de que muchos usuarios ingresen a la aplicación, el sistema empieza a presentar fallas y deniega la entrada de usuarios a la aplicación

2.3.4.1.3 Escenario de calidad 3

Identificador	ESC-CAL-CAR-DIS-02-03
Característica	CAR-DIS-02
Taxonomía	Operaciones simultaneas
Atributo de calidad	Disponibilidad
Tipo escenario	No exitoso
¿Escenario cumplido?	No
Descripción	Asegura que la aplicación no funcione correctamente cuando se tenga un bajo flujo de usuarios
Origen del estímulo	Cualquier usuario final
Estímulo	Usar la aplicación
Artefacto	Sistema
Ambiente	Operación con alto flujo de usuarios
Respuesta	El sistema se cae y deja de funcionar
Medida de la respuesta	Luego del ingreso de que algunos usuarios ingresen a la aplicación, el sistema empieza a presentar fallas y deniega la entrada de usuarios a la aplicación

2.3.5 Atributo calidad 5: Flexibilidad

capacidad de un sistema o proyecto para adaptarse y evolucionar en respuesta a cambios en los requisitos, tecnologías o condiciones del entorno. Implica la capacidad de realizar modificaciones sin interrupciones significativas en la funcionalidad. La flexibilidad es esencial en el desarrollo del proyecto para garantizar que el sistema pueda evolucionar con el tiempo. Esto implica la creación de una arquitectura modular y escalable que permita agregar nuevas funcionalidades o realizar modificaciones sin afectar el funcionamiento existente, además, En la definición del diseño, la flexibilidad se logra mediante la adopción de prácticas de desarrollo ágil, la implementación de interfaces bien definidas y la separación de preocupaciones. También implica la creación de una base de código que sea fácil de mantener y extender.

2.3.5.1 Característica 1: Durabilidad de la sesión de usuario

La sesión del usuario deberá mantenerse iniciada al salir y volver a entrar en la aplicación para proporcionar una experiencia continua y conveniente

2.3.5.1.1 Escenario de calidad 1

Identificador	ESC-CAL-CAR-FLX-03-01
Característica	CAR-FLX-03
Taxonomía	Recordar usuario
Atributo de calidad	Flexibilidad
Tipo escenario	Exitoso
¿Escenario cumplido?	No
Descripción	Asegura que el usuario al momento de abrir la aplicación, la aplicación tenga la capacidad de iniciar sesión automáticamente en la mayoría de los casos
Origen del estímulo	Cualquier usuario final
Estímulo	Cuando el usuario entre a la aplicación
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	El usuario final ingresa a la aplicación sin la necesidad de ingresar sus datos nuevamente
Medida de la respuesta	El usuario final ingresa a la aplicación, y es enviado a la página principal automáticamente

2.3.5.1.2 Escenario de calidad 2

Identificador	ESC-CAL-CAR-FLX-03-02
Característica	CAR-FLX-03
Taxonomía	Recordar usuario
Atributo de calidad	Flexibilidad
Tipo escenario	No exitoso
¿Escenario cumplido?	No
Descripción	Asegura que al momento de abrir la aplicación, la sesión del usuario no se encuentra activa
Origen del estímulo	Cualquier usuario final
Estímulo	Cuando el usuario final intenta ingresar a la aplicación
Artefacto	Sistema
Ambiente	Operación normal
Respuesta	La sesión del usuario final no se inicia automáticamente
Medida de la respuesta	Al ingresar a la aplicación, la sesión del usuario final no se inicia automáticamente, así que se redirige al usuario final a la ventana de inicio de sesión

2.4 Funcionalidades críticas

Las funcionalidades críticas, también conocidas como "características críticas" o "funciones críticas", son aquellas características, componentes o aspectos de un sistema, producto o software que son esenciales para su funcionamiento básico y que tienen un impacto significativo en la satisfacción del usuario, la seguridad o el cumplimiento de los objetivos del sistema. Estas funcionalidades son vitales para el éxito y la efectividad del producto o sistema en su conjunto.

2.4.1 Funcionalidad 1

El sistema debe garantizar que en cualquier momento los usuarios puedan agendar un servicio de traducción con un traductor de acuerdo con la disponibilidad de este.

2.4.1.1 Justificación

Es muy importante garantizar que en cualquier momento los usuarios puedan agendar un servicio con un traductor teniendo en cuenta la disponibilidad de este, ya que esto es fundamental para llevar a cabo un servicio de traducción

2.4.2 Funcionalidad 2

Se debe garantizar que en todo momento se pueda observar la disponibilidad de un traductor en tiempo real

2.4.2.1 Justificación

Es muy importante garantizar que en cualquier momento los usuarios puedan ver en tiempo real la disponibilidad de un traductor, incluso si otro usuario agenda o cancela un servicio, esto se debe reflejar en la disponibilidad para que el usuario sepa con claridad a qué hora puede o no agendar un servicio

2.4.3 Funcionalidad 3

Se debe garantizar que un usuario pueda publicar una solicitud de servicio en donde pueda especificar las necesidades que quiere que el traductor pueda suplir

2.4.3.1 Justificación

Es muy importante garantizar que un usuario pueda publicar una solicitud de servicio en donde pueda especificar las necesidades que tiene, para que los traductores que puedan suplir esas necesidades contacten al usuario y así puedan agendar el servicio

2.4.4 Funcionalidad 4

El sistema debe garantizar que los usuarios puedan intercambiar mensajes con traductores

2.4.4.1 Justificación

Es muy importante garantizar que los usuarios y traductores puedan intercambiar mensajes entre ellos para facilitar la comunicación y que ambos puedan conocer mejor las necesidades del usuario y las habilidades del traductor, agilizando así el proceso de agendamiento de un servicio

2.4.5 Funcionalidad 5

Se debe garantizar que un traductor pueda en cualquier momento añadir o eliminar información a su perfil

2.4.5.1 Justificación

Es muy importante garantizar que en cualquier momento los traductores puedan editar su perfil añadiendo o eliminando información para que lo puedan mantener actualizado, aumentando sus posibilidades de ser seleccionado para llevar a cabo un servicio

2.4.6 Funcionalidad 6

El sistema debe garantizar que los usuarios puedan calificar a un traductor tras completar un servicio agendado con él

2.4.6.1 Justificación

Es muy importante garantizar que después de haberse realizado un servicio de traducción, el usuario pueda calificar al traductor en base a la experiencia del servicio que recibió para que así otros usuarios puedan tomar decisiones acertadas al momento de buscar un traductor que se adapte a sus necesidades

2.4.7 Funcionalidad 7

Se debe garantizar el envío de notificaciones a usuarios y traductores que informen sobre novedades que se presenten en un servicio de traducción

2.4.7.1 Justificación

Es muy importante garantizar que en el instante en el que suceda una novedad con un servicio, como por ejemplo su agendamiento, cancelamiento o finalización, se envíen notificaciones tanto a usuarios como a traductores porque así podrán estar informados del estado del servicio

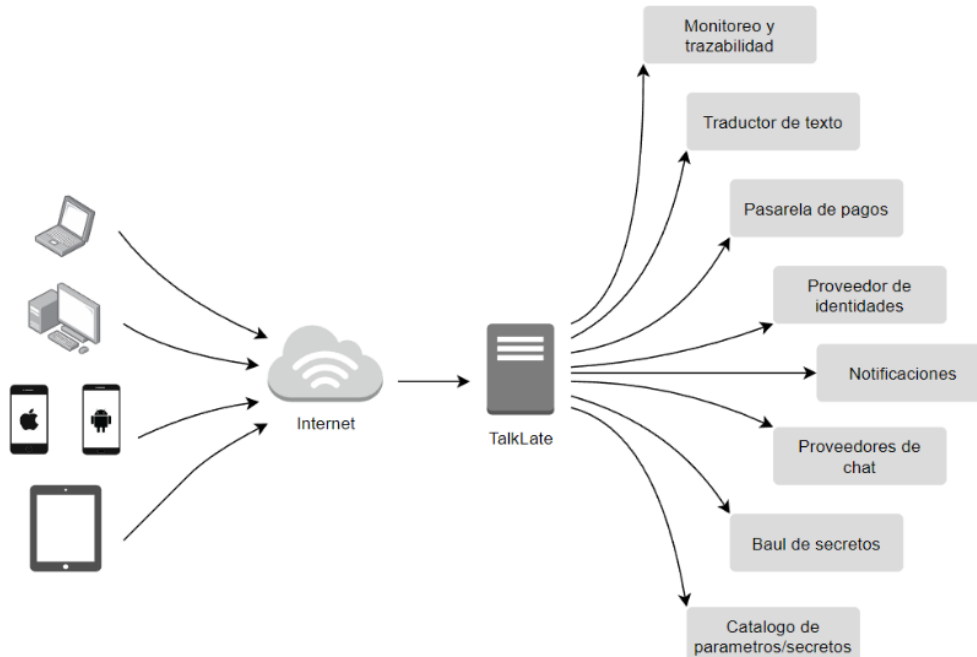
3. Tácticas y estrategias

Táctica	Ventajas	Desventajas
Uso de un video tutorial	Los videotutoriales permiten presentar información secuencial y estructurada, lo que facilita el aprendizaje paso a paso	Como la aplicación está disponible en varias regiones e idiomas se deben crear videos en distintos idiomas, lo que puede ser costoso y tomar más
Uso de herramientas para la creación de tutoriales interactivos	Los tutoriales interactivos son efectivos para facilitar el aprendizaje No es necesario crear muchos tutoriales que expliquen lo mismo en diferentes idiomas ya que se puede traducir el contenido del tutorial al idioma en el que esté la aplicación	El equipo de desarrollo no sabe cómo crear tutoriales interactivos
Diseño escalable	La capacidad de la aplicación para crecer y adaptarse a un mayor número de usuarios sin afectar negativamente su rendimiento y eficiencia, lo que resulta en un ahorro de costos al agregar recursos de manera selectiva. Los sistemas escalables pueden mantener tiempos de respuesta bajos incluso bajo cargas pesadas, lo que mejora la experiencia del usuario y la satisfacción del cliente.	La implementación de un diseño escalable puede ser más compleja y requerir una planificación adicional. El hardware y la infraestructura requeridos para admitir la escalabilidad pueden tener un costo inicial más alto. Esto puede ser un desafío para startups o empresas con recursos limitados.
Balaneo de carga	mejorar la disponibilidad de la aplicación Distribución equitativa del tráfico entre servidores para evitar la sobrecarga en uno solo	aumenta la carga de trabajo inicial del equipo de desarrollo La configuración inicial de un sistema de balanceo de carga puede requerir tiempo y recursos adicionales. Esto puede retrasar la implementación
CDN (Content Delivery Network)	reduce la carga en el servidor principal, mejorando la experiencia de usuario Acelerar la entrega de contenido estático a usuarios de todo el mundo, mejorando la velocidad de carga.	El uso de una CDN para entregar contenido estático puede generar costos adicionales según la cantidad de tráfico y datos transmitidos. Estos costos pueden acumularse, especialmente para aplicaciones con un gran volumen
Cookies de sesión en el navegador	Son fáciles de usar	Pueden generar vulnerabilidades en seguridad
Uso de tokens de autenticación	Son más seguros, además se puede configurar su duración pueden llevar información adicional del usuario como roles o permisos	el equipo de desarrollo no sabe cómo implementarlos Su implementación es más compleja
Almacenar sesiones en servidores	Es un enfoque seguro	Puede aumentar la carga del servidor y complicar la escalabilidad
Almacenar sesiones localmente	Mejora el rendimiento y la experiencia de usuario	Puede generar vulnerabilidades de seguridad
Desarrollo web nativo	Puedes tener un control completo sobre el desarrollo y diseño de tu sitio web para garantizar una experiencia de usuario personalizada al desarrollar específicamente para la web, puedes optimizar el rendimiento y la velocidad de carga, también puedes implementar estrategias de SEO de manera más efectiva para mejorar la visibilidad en motores de búsqueda.	El desarrollo web nativo puede llevar más tiempo y recursos que otras opciones requiere un mantenimiento constante para mantener la compatibilidad con navegadores y asegurar que el sitio funcione correctamente.
Frameworks de Desarrollo Web	Utilizar framework para el desarrollo web puede acelerar el proceso de desarrollo y los sitios web desarrollados con frameworks suelen ser compatibles con una variedad de navegadores puedes aprovechar la comunidad de desarrolladores y recursos disponibles para resolver problemas rápidamente.	Debes aprender el framework, lo que puede llevar tiempo si no estás familiarizado con él algunos frameworks pueden incluir características innecesarias que pueden afectar negativamente el rendimiento si no se utilizan correctamente.

Frameworks de desarrollo de aplicaciones de para dispositivos móviles	Las ventajas de los frameworks de desarrollo para dispositivos móviles radican en su capacidad para simplificar el proceso de creación de aplicaciones móviles, permitiendo a los desarrolladores escribir código una vez y ejecutarlo en múltiples plataformas, lo que ahorra tiempo y recursos	Incluyen limitaciones en la personalización y el rendimiento en comparación con el desarrollo nativo, así como la dependencia de las actualizaciones y la compatibilidad de los frameworks con las últimas versiones de sistemas operativos móviles
SDK'S para el desarrollo de aplicaciones de dispositivos móviles	Incluyen la provisión de herramientas, bibliotecas y recursos que agilizan el proceso de desarrollo facilitan el acceso a características específicas de la plataforma y permiten una mayor eficiencia en la creación de aplicaciones personalizadas	pueden abarcar la dependencia de actualizaciones de SDK la necesidad de aprender nuevas API y la posible falta de soporte para ciertas características o plataformas, lo que puede limitar la flexibilidad y requerir adaptaciones adicionales en el desarrollo.
Herramientas de envío de mensajes de texto o correos electrónicos con un enlace o código para iniciar sesión	Al ser herramientas que ya existen se ahorra el tiempo de tener que construirlas desde cero	Se debe tener en cuenta que por lo general son herramientas de pago mensual
Crear funciones que se encarguen de enviar correos electrónicos con un enlace o código para iniciar sesión	Se evitan costos adicionales	El desarrollo de estas herramientas pueden tomar tiempo de desarrollo que podría emplearse en la solución de otras funcionalidades
Implementar funciones que exijan un patrón de contraseña específico	se tendrá más seguridad al momento de crear y tener las contraseñas en la aplicación se podrá implementar el patrón que se quiera	Tomaría más tiempo de desarrollo
Roles de usuario	Los roles de usuario permiten una gestión eficiente de los permisos y responsabilidades dentro de una aplicación.	La creación y gestión de múltiples roles de usuario puede ser compleja, especialmente en aplicaciones con una gran variedad de funciones y perfiles de usuarios. Esto puede requerir un esfuerzo adicional en el desarrollo y la configuración inicial, lo que puede aumentar la complejidad del sistema.
Tokens configurados con roles	Al asignar roles específicos a los tokens, se puede mejorar la comprensión contextual y facilitar la manipulación y presentación de datos dentro de la aplicación.	pueden incluir una mayor complejidad en la implementación y la necesidad de seguir convenciones claras en la asignación de roles. el uso excesivo de tokens configurados con roles puede aumentar la longitud de las secuencias de texto, lo que podría afectar el rendimiento y la eficiencia de la aplicación, especialmente en dispositivos móviles con recursos
Bibliotecas para la autenticación de red social	Ahorran tiempo de desarrollo facilitan la autenticación de los usuarios	Las herramientas que ofrecen más calidad pueden ser de pago mensual
Enlace de Restablecimiento de Contraseña	Los usuarios pueden restablecer sus contraseñas de manera sencilla a través de un enlace enviado por correo electrónico y el enlace de restablecimiento de contraseña puede tener una fecha de vencimiento y ser único, lo que aumenta la seguridad.	Si un atacante obtiene acceso a la cuenta de correo electrónico del usuario, podría restablecer la contraseña sin permiso y los correos electrónicos de restablecimiento de contraseña pueden terminar en la carpeta de correo no deseado, lo que puede causar problemas de acceso para los usuarios.

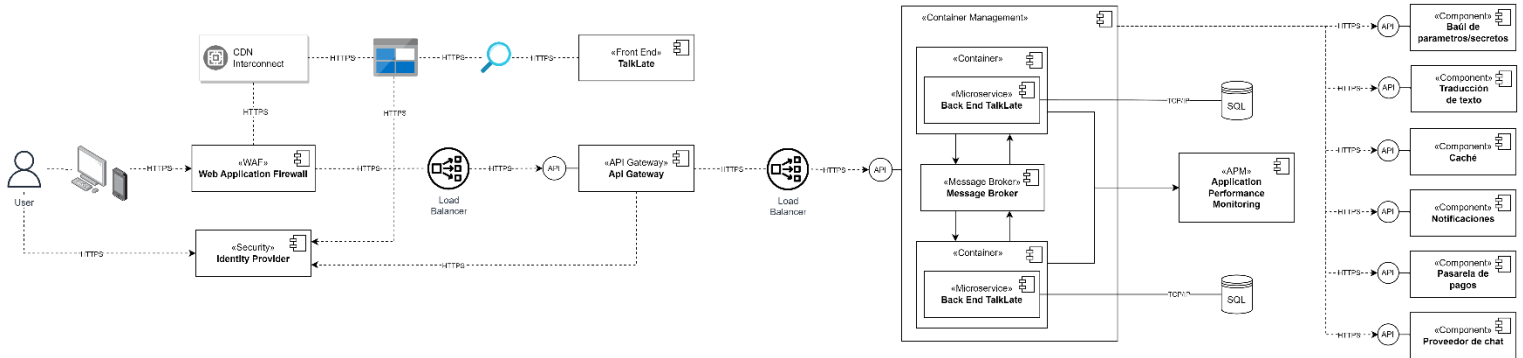
4. Modelo de contexto

es una representación gráfica que muestra las relaciones y las interacciones de un sistema con su entorno. Este tipo de diagrama se utiliza para proporcionar una vista de alto nivel de un sistema y su relación con otros sistemas, subsistemas, usuarios y componentes externos. El objetivo principal del diagrama de contexto es definir los límites del sistema y cómo se conecta con el mundo exterior.

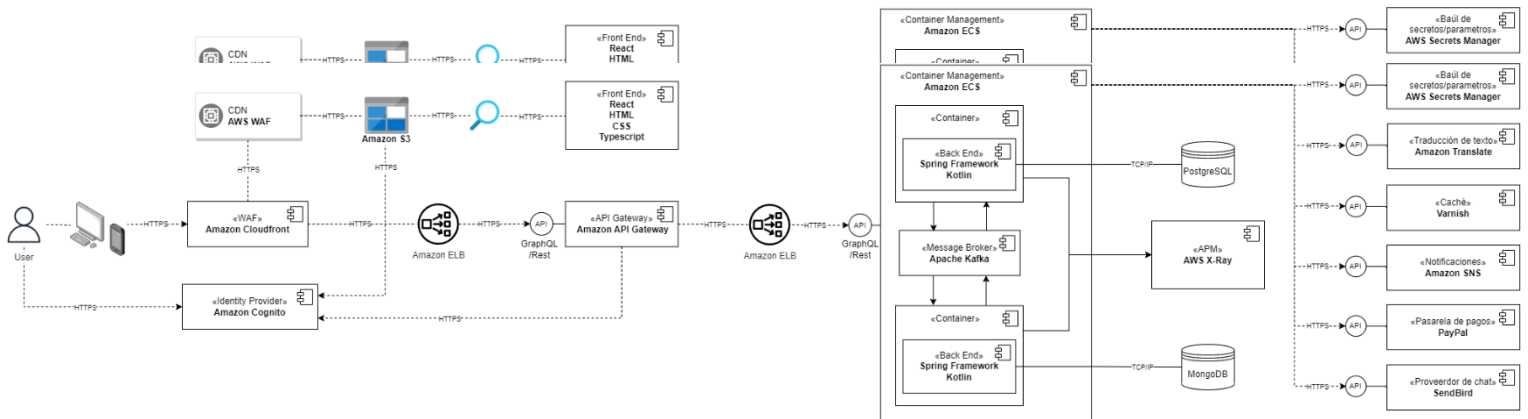


5. Arquetipo de solución/referencia

patrón o modelo reutilizable que describe una solución típica para un conjunto de problemas comunes. Los arquetipos de solución son como plantillas o ejemplos que los arquitectos y diseñadores pueden utilizar como punto de partida al abordar ciertos tipos de desafíos en el diseño de sistemas.



6. Arquitectura de solución/referencia



6.1 Documentación

Componente/Elemento Arquitectónico	Descripción	Justificación	Marca/Fabricante	Tipo producto/Tipo uso	Tipo adquisición
Content Delivery Network	Vínculo (Link CDN)	La implementación de un CDN en TalkLate asegura que el contenido estático de la aplicación como el del sitio web, contenido multimedia y otros elementos que se consideran no cambian en el tiempo, puedan ser entregados al cliente o usuario final de forma muy rápida, eficiente, segura y comprimida	Amazon Cloudfront	Pago	Adoptado
Load Balancer	Vínculo (Link Load Balancer)	La implementación de un Load Balancer en TalkLate asegura una entrega rápida y eficiente de contenido estático al distribuir las solicitudes de los usuarios a servidores optimizados. Garantiza una experiencia de usuario más veloz mediante la compresión de datos, optimización del ancho de banda y la implementación de técnicas de caché. Además, proporciona seguridad contra ataques y alta disponibilidad al equilibrar la carga dinámicamente y redirigir las solicitudes a servidores funcionales, mejorando la confiabilidad y eficacia del servicio.	Amazon ELB	Pago/Gratuito	Adoptado
Web Application Firewall	Vínculo (Link WAF)	La implementación de un WAF en TalkLate asegura que la aplicación posea un escudo de protección como primera línea de defensa para los posibles ataques que se puedan presentar. Además, permite establecer políticas de seguridad personalizadas, monitorear el tráfico en tiempo real y proporcionar defensa proactiva contra vulnerabilidades conocidas, asegurando así la integridad, confidencialidad y disponibilidad de los datos, así como la experiencia segura del cliente o usuario final	AWS WAF	Pago	Adoptado
Identity Provider	Vínculo (Link Identity Provider)	La utilización de un Identity Provider en TalkLate ofrece una solución segura y eficiente para la autenticación y autorización de usuarios. Al integrar un Identity Provider, se permite a los usuarios acceder a la aplicación de forma rápida y segura utilizando sus credenciales existentes, simplificando el proceso de inicio de sesión y asegurando la autenticidad de las identidades.	Amazon Cognito	Pago	Adoptado
Blob Storage	Vínculo (Link Blob Storage)	La implementación de un Blob Storage en TalkLate garantiza una entrega eficiente y segura de contenido estático, como imágenes, videos y otros recursos inalterables, a los usuarios finales. Al almacenar estos elementos en un Blob Storage, se optimiza el rendimiento, asegurando una rápida carga de la aplicación, además de proporcionar una gestión eficaz de recursos multimedia, asegurando una experiencia del usuario fluida y consistente.	Amazon S3	Pago/Gratuito	Adoptado
Api Gateway	Vínculo (Link Api Gateway)	La implementación de un API Gateway en TalkLate ofrece una capa de gestión centralizada para las solicitudes de API, asegurando una comunicación segura y eficiente entre los servicios internos y externos. Al consolidar múltiples puntos de acceso en una sola interfaz, se simplifica la complejidad del sistema, permitiendo una monitorización y control detallados.	Amazon API Gateway	Pago	Adoptado

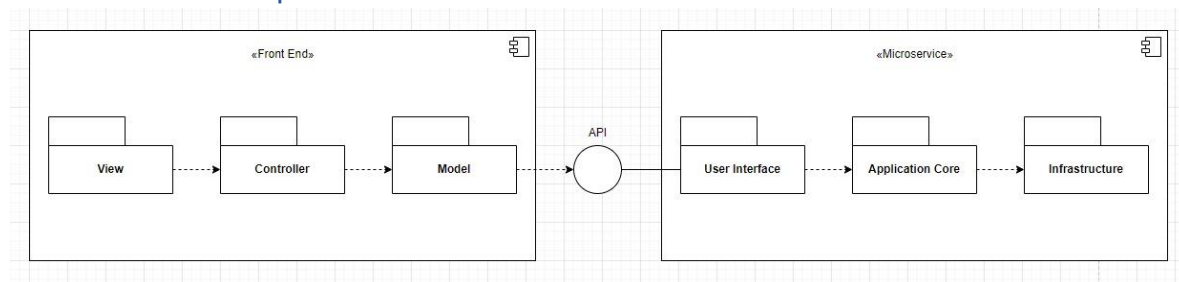
Frontend TalkLate	Vinculo (Link Frontend)	Un frontend en TalkLate es esencial para proporcionar a los usuarios una interfaz amigable y accesible que garantice una experiencia de usuario intuitiva y eficiente al acceder a los servicios ofrecidos por la aplicación.	HTML CSS Typescript React		
Container	Vinculo (Link Container)	El uso de Containers en TalkLate ofrece un ambiente encapsulado y aislado, asegurando la consistencia del entorno de ejecución y facilitando las actualizaciones sin interrupciones, optimizando así la eficiencia operativa y garantizando una experiencia de usuario sin contratiempos y confiable.	Amazon ECS	Pago	Adoptado
Backend TalkLate	Vinculo (Link Backend)	El backend en TalkLate es esencial para garantizar el cumplimiento de las reglas de negocio de la aplicación, además permite una fácil integración de tecnologías y servicios externos, lo que facilita la expansión y la adaptabilidad a medida que la aplicación crece y se desarrolla.			
Message Broker	Vinculo (Link Message Broker)	La implementación de un message broker en TalkLate garantiza una comunicación asíncrona eficiente y escalable entre los componentes de la aplicación, lo que permite una gestión de eventos y mensajes en tiempo real, un mejor rendimiento, una alta disponibilidad y una distribución de tareas más efectiva.	Apache Kafka	Gratuito/Open Source	Adoptado
Database	Vinculo (Link Database)	La implementación de una base de datos en TalkLate es necesaria ya que garantiza una gestión óptima de los datos y facilita operaciones de lectura y escritura, asegurando respuestas rápidas a las consultas de los usuarios.	PostgreSQL	Gratuito/Open Source	Adoptado
			MongoDB	Gratuito/Open Source	Adoptado
Application Performance Monitoring	Vinculo (Link APM)	La utilización de un APM en TalkLate garantiza una supervisión proactiva y detallada del rendimiento de la aplicación en tiempo real. Esto permite identificar y resolver rápidamente cuellos de botella, tiempos de respuesta lentos y otros problemas de rendimiento, asegurando así una experiencia del usuario óptima. Al monitorear continuamente la aplicación, se pueden detectar y abordar los problemas antes de que afecten negativamente a los usuarios, garantizando un funcionamiento suave y eficiente en todo momento.	AWS X-Ray	Pago/Gratuito	Adoptado
Baúl de parametros y secretos	Vinculo (Link Baúl de parametros y secretos)	La adopción de un baul de parámetros y secretos en TalkLate garantiza la gestión segura y eficiente de información confidencial, como claves de API y configuraciones específicas. Al centralizar estos datos sensibles en un repositorio seguro, se facilita la administración, se refuerza la seguridad y se asegura el acceso autorizado a recursos críticos, lo que es esencial para la integridad y confidencialidad de la aplicación, así como para la experiencia segura del usuario final.	AWS Secrets Manager	Pago	Adoptado

Notificaciones	Vinculo (Link Notificaciones)	La inclusión de un componente de notificaciones en TalkLate es esencial para garantizar una comunicación efectiva y oportuna con los usuarios. Este componente permite enviar alertas instantáneas sobre traducciones disponibles, confirmaciones de reservas u otros mensajes importantes, mejorando así la experiencia del usuario y manteniéndolos informados en tiempo real.	Amazon SNS	Pago	Adoptado
Pasarela de pagos	Vinculo (Link Pasarela de pagos)	La implementación de una pasarela de pagos en TalkLate es esencial para brindar a los usuarios una experiencia segura y eficiente al realizar transacciones financieras dentro de la aplicación. Asegura la protección de datos sensibles, la confiabilidad en las transacciones y la aceptación de diferentes métodos de pago, lo que es fundamental para la monetización exitosa de la plataforma y la satisfacción del usuario.	PayPal	Pago	Adoptado
Traducción de texto	Vinculo (Link Traducción de texto)	La integración de un componente de traducción de texto en TalkLate garantiza que el contenido de la aplicación, incluidos textos, mensajes y etiquetas, se presente de manera precisa y relevante en el idioma solicitado por el usuario. Esto no solo mejora la accesibilidad global, sino que también facilita la comprensión y la interacción del usuario con la aplicación, promoviendo una experiencia de usuario enriquecedora y personalizada en diversos contextos culturales y lingüísticos.	Amazon Translate	Pago	Adoptado
Chat Provider	Vinculo (Link Chat Provider)	La adopción de un proveedor de chat en TalkLate garantiza una comunicación en tiempo real fluida y segura entre usuarios, mejorando significativamente la experiencia del usuario.	SendBird	Pago	Adoptado
Caché	Vinculo (Link Caché)	La implementación de un sistema de caché en TalkLate garantiza una entrega ultrarrápida y eficiente de contenido estático como imágenes, videos y elementos del sitio web, optimizando la experiencia del usuario final. Al almacenar copias previamente cargadas de estos elementos, se reduce significativamente el tiempo de carga, mejora la eficiencia y seguridad en la entrega, además de minimizar la carga del servidor de origen.	Varnish	Gratuito/Open Source	Adoptado
Server	Vinculo (Link Server Provider)	La adopción de un servidor en la nube para TalkLate garantiza una entrega ultrarrápida y segura de su contenido estático, como imágenes y multimedia, a los usuarios finales. Al alojar estos elementos en servidores distribuidos globalmente, se optimiza la velocidad de carga y la experiencia del usuario, asegurando una eficiencia sin igual y protección contra amenazas, todo mientras se minimiza el consumo de ancho de banda gracias a la compresión inteligente de datos.	AWS	Pago	Adoptado

7. Línea base arquitectónica

se refiere a una versión fundamental y estable de la arquitectura de un sistema en un momento específico del ciclo de vida del proyecto. Esta línea base establece los principios y las decisiones clave de diseño arquitectónico que guiarán el desarrollo del sistema. En esencia, es un punto de referencia que define la estructura y la dirección arquitectónica del proyecto.

7.1 Línea base arquitectónica General



7.2 Estilos y patrones arquitectónicos adoptados

Estos estilos y patrones son pautas predefinidas que se han probado y utilizado en múltiples proyectos para resolver problemas arquitectónicos específicos.

7.2.1 Estilo arquitectónico 1

Arquitectura orientada a objetos

7.2.1.1 Nombre

Arquitectura Orientada a Objetos

7.2.1.2 Problema

La arquitectura orientada a objetos aborda el problema de diseñar sistemas de software de manera que los componentes del sistema se modelen como objetos, que son instancias de clases en un lenguaje de programación orientado a objetos. Este estilo arquitectónico es adecuado cuando se requiere una estructura de software que sea modular, reutilizable, y que permita modelar el mundo real de manera más natural. Se utiliza en una variedad de contextos, desde aplicaciones empresariales hasta sistemas de software embebidos.

7.2.1.3 Solución/Motivación

La solución propuesta por la arquitectura orientada a objetos consiste en diseñar el sistema en torno a objetos, que encapsulan datos y comportamiento relacionados. Estos objetos interactúan entre sí a través de mensajes, lo que promueve la modularidad y la reutilización del código. La motivación detrás de este estilo es lograr un diseño más flexible y mantenible, donde los cambios en una parte del sistema no afecten necesariamente a otras partes. También se basa en el principio de modelar el software de manera similar a cómo se modela el mundo real, lo que facilita la comprensión y la comunicación entre los miembros del equipo de desarrollo.

7.2.2 Estilo arquitectónico 2

Arquitectura en capas

7.2.2.1 Nombre

Arquitectura en capas

7.2.2.2 Problema

La arquitectura en capas aborda el problema de organizar un sistema de software de manera que los componentes estén estructurados en capas o niveles lógicos. Este estilo es útil cuando se necesita separar las preocupaciones y crear una estructura modular que facilite la escalabilidad y el mantenimiento. Se aplica en situaciones donde es importante tener una clara separación entre las partes de un sistema.

7.2.2.3 Solución/Motivación

La solución propuesta por la arquitectura en capas consiste en dividir el sistema en capas lógicas, donde cada capa tiene una responsabilidad específica. Las capas se comunican entre sí siguiendo un flujo unidireccional, generalmente desde la capa de presentación o interfaz de usuario hacia la capa de acceso a datos. Esto promueve la modularidad y la reutilización, ya que cada capa puede ser modificada o reemplazada sin afectar a las demás.

7.2.3 Estilo arquitectónico 3

arquitectura orientada a eventos

7.2.3.1 Nombre

arquitectura orientada a eventos

7.2.3.2 Problema

La arquitectura orientada a eventos aborda el problema de diseñar sistemas de software en los que los componentes se comunican y responden a eventos y cambios en lugar de depender de una secuencia lineal de instrucciones. Este estilo es adecuado cuando se necesita manejar eventos, notificaciones y flujos de datos en tiempo real, y cuando se busca una arquitectura altamente escalable y reactiva.

7.2.3.3 Solución/Motivación

La solución propuesta por la arquitectura orientada a eventos es diseñar el sistema de manera que los componentes se registren como oyentes (listeners) de eventos específicos. Cuando

ocurre un evento, se desencadenan acciones o funciones específicas en los componentes registrados para ese evento. Esto permite una arquitectura más flexible y reactiva, donde los componentes pueden responder a eventos de manera asincrónica.

8. Justificación alternativa de solución

Una justificación de alternativa de solución es donde se da explicación de por qué se tomó esa alternativa de solución

8.1 Justificación

Cloudflare WAF y Amazon Web Services (AWS) ofrecen seguridad y escalabilidad en la nube, respaldados por servicios como Amazon Cognito para la autenticación, Amazon API Gateway para la administración de API, y Amazon S3 para el almacenamiento de datos confiable. En el lado del cliente, el uso de React, HTML, CSS y TypeScript garantiza una experiencia de usuario moderna y eficiente. Amazon Elastic Load Balancer (ELB) gestiona la distribución de tráfico de manera equitativa y escalable. Spring Framework y Kotlin, junto con Apache Kafka, brindan un sólido soporte para el desarrollo de aplicaciones empresariales y la gestión de flujos de datos en tiempo real. PostgreSQL y MongoDB se adaptan a diversas necesidades de almacenamiento de datos, y AWS X-Ray proporciona visibilidad y supervisión. Esta combinación de tecnologías permite una implementación eficiente y segura de un sistema que cumple con los requisitos específicos del proyecto.

8.2 Ventajas

una combinación de seguridad y escalabilidad a través de Cloudflare WAF y Amazon Web Services (AWS), un sólido soporte para la autenticación con Amazon Cognito, una administración de API eficiente mediante Amazon API Gateway, y almacenamiento confiable con Amazon S3. Para la experiencia del usuario en el frontend, React, HTML, CSS y TypeScript ofrecen una interfaz moderna y segura. Amazon Elastic Load Balancer (ELB) garantiza una distribución equitativa y escalable del tráfico, mientras que Spring Framework, Kotlin y Apache Kafka brindan una base sólida para el desarrollo de aplicaciones empresariales y la gestión de flujos de datos en tiempo real. Las opciones de bases de datos, PostgreSQL y MongoDB, se adaptan a diversas necesidades de almacenamiento de datos, y AWS X-Ray proporciona visibilidad y supervisión. Esta selección de tecnologías permite una implementación eficiente y segura de un sistema que cumple con los requisitos específicos del proyecto.

8.3 Desventajas

costos asociados en varias de las soluciones, incluyendo Cloudflare WAF, Amazon Cognito, Amazon API Gateway, Amazon S3, Amazon ELB y AWS X-Ray, lo que puede impactar el presupuesto del proyecto. Algunas de las tecnologías, como React y TypeScript, tienen una curva de aprendizaje inicial, y requieren buenas prácticas de seguridad para evitar vulnerabilidades. Además, tecnologías como Amazon ECS requieren conocimientos específicos de Docker y AWS para su configuración adecuada. En el caso de bases de datos, PostgreSQL puede requerir una administración y configuración experta para optimizar su rendimiento, mientras que MongoDB carece de soporte para transacciones ACID en ciertos escenarios.

9. Vistas de arquitectura del sistema

Las vistas de diseño del sistema son representaciones visuales que describen cómo se estructura y se organiza un sistema de software en términos de sus componentes, módulos, relaciones y comportamientos. Estas vistas proporcionan una perspectiva detallada de la arquitectura interna del sistema, permitiendo a los desarrolladores, arquitectos y otros stakeholders comprender mejor cómo funcionan y se relacionan las partes del sistema en conjunto. Cada vista se enfoca en un aspecto específico de la arquitectura, como la vista de componentes, la vista de despliegue, la vista de datos, entre otras, para facilitar la comprensión y la comunicación efectiva durante el proceso de diseño y desarrollo del software.

9.1 Vista Funcional/Vista de Escenarios/Vista de Casos de Uso

representación visual y descriptiva de cómo el sistema de software se comportará desde la perspectiva de las funciones y acciones que realiza. Esta vista se centra en las interacciones y procesos funcionales del sistema, sin entrar en detalles técnicos de implementación. La motivación detrás de una vista funcional es comprender y comunicar de manera efectiva qué hace el sistema y cómo interactúa con los usuarios y otros componentes, sin preocuparse demasiado por cómo se construye internamente.

9.1.1 Modelo de procesos del negocio

representación visual que describe cómo una organización lleva a cabo sus actividades y operaciones. Este modelo muestra cómo se ejecutan los procesos, las interacciones entre ellos y cómo fluye la información a través de la organización. El objetivo principal de un modelo de procesos de negocio es comprender, analizar y mejorar los procesos para aumentar la eficiencia, la calidad y la productividad de la organización.

9.1.2 Glosario de términos del negocio

Traductor: Persona que ofrece servicios de traducción en la plataforma Talklate.

Usuario: Persona que utiliza la aplicación Talklate para solicitar servicios de traducción.

Solicitud de Traducción: Petición realizada por un usuario para obtener servicios de traducción.

Conversación: Interacción entre un usuario y un traductor dentro de la aplicación.

Calificación: Evaluación dada por un usuario o traductor para medir la calidad de los servicios prestados.

Chat Integrado: Funcionalidad que permite la comunicación en tiempo real entre usuario y traductor.

Historial de Conversación: Registro de las conversaciones pasadas entre usuarios y traductores.

Planes de Pago: Diferentes opciones de suscripción o tarifas que ofrecen características adicionales en la aplicación.

Publicidad: Anuncios mostrados en la aplicación como fuente de ingresos.

Dispositivo: Equipo o hardware desde el cual se accede a la aplicación, como teléfonos, tabletas y computadoras.

Notificaciones: Mensajes automáticos que alertan a los usuarios sobre actividades importantes en la aplicación.

Perfil de Traductor: Página personalizada de un traductor que muestra su experiencia y habilidades.

Soporte al Cliente: Servicio de atención al cliente que responde a las preguntas y preocupaciones de los usuarios.

Base de Datos: Almacenamiento de información sobre usuarios, traductores y conversaciones.

Pagos en Línea: Proceso para realizar transacciones financieras dentro de la aplicación.

Verificación de Habilidades: Proceso para confirmar las competencias lingüísticas de un traductor.

Clasificación de Traductores: Evaluación que categoriza a los traductores según su experiencia y habilidades.

Restricciones de Negocio: Limitaciones o reglas específicas que rigen el funcionamiento de la aplicación.

Términos y Condiciones: Reglas y políticas que los usuarios y traductores deben aceptar al utilizar la aplicación.

9.1.3 Especificación de requisitos de software

Una especificación de requisitos de software es un documento detallado que describe de manera precisa y completa las necesidades y expectativas de un sistema de software. Este documento identifica y documenta todos los requisitos funcionales y no funcionales del software, como las características que debe tener, el comportamiento esperado, las restricciones, el rendimiento y otros aspectos relevantes. La especificación de requisitos de software sirve como guía esencial para el desarrollo del proyecto, ya que proporciona a los desarrolladores, arquitectos y otros stakeholders una comprensión clara de lo que se espera que el software haga y cómo debe

comportarse. Además, esta especificación es crucial para la definición del diseño, ya que establece la base sobre la cual se construirá y probará el software, garantizando que cumpla con las necesidades y expectativas del usuario final.

9.1.3.1 Requisitos de usuario

Registro de Cuenta: Los usuarios deben poder registrarse en la plataforma proporcionando información básica, como nombre, dirección de correo electrónico y contraseña.

Autenticación Segura: Se requiere un proceso de autenticación seguro para proteger las cuentas de usuario y garantizar la privacidad.

Perfiles de Usuario: Cada usuario debe tener un perfil personalizado donde pueda agregar información adicional, como una foto de perfil, idiomas que habla, especializaciones, etc.

Solicitudes de Traducción: Los usuarios deben poder enviar solicitudes de traducción especificando el idioma de origen y el idioma de destino, así como la cantidad de palabras o el texto a traducir.

Selección de Traductor: Los usuarios deben poder elegir entre varios traductores disponibles y revisar sus perfiles antes de seleccionar uno.

Comunicación en Tiempo Real: La plataforma debe admitir un sistema de chat en tiempo real que permita a los usuarios y traductores comunicarse de manera eficiente durante el proceso de traducción.

Calificaciones y Comentarios: Los usuarios deben tener la opción de calificar a los traductores y dejar comentarios sobre la calidad de los servicios de traducción proporcionados.

Historial de Conversaciones: Los usuarios deben poder acceder a un historial de sus conversaciones anteriores con traductores para referencia futura.

Opciones de Pago: Deben ofrecerse diversas opciones de pago para las tarifas de traducción, lo que incluye la posibilidad de elegir entre planes de pago y métodos de pago seguros.

Notificaciones: Los usuarios deben recibir notificaciones para mantenerlos informados sobre el estado de sus solicitudes, mensajes importantes y actualizaciones de la plataforma.

Soporte al Cliente: La plataforma debe proporcionar un canal de soporte al cliente para que los usuarios puedan hacer preguntas, informar problemas o buscar asistencia.

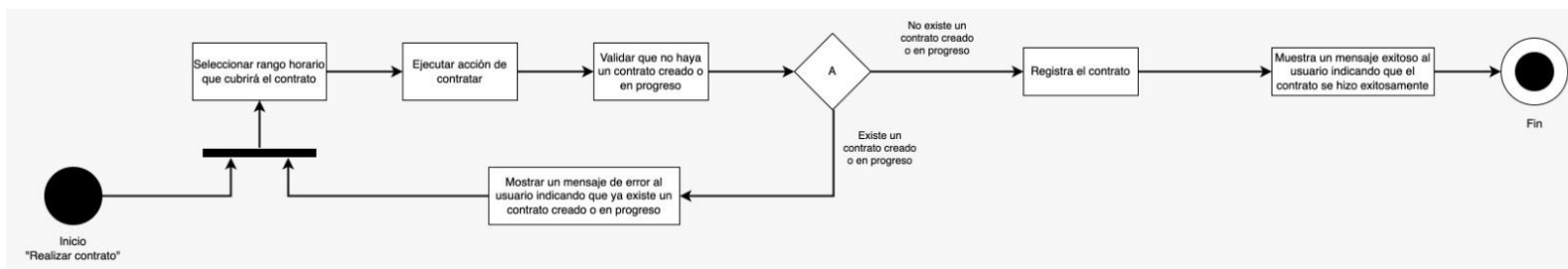
Privacidad y Seguridad: Se debe garantizar la privacidad y seguridad de los datos del usuario, implementando medidas de seguridad efectivas.

9.1.4 Diagrama de actividades

Un diagrama de actividades es una representación visual que se utiliza en la modelización de procesos y sistemas para mostrar la secuencia de actividades, tareas y decisiones que ocurren en un proceso o flujo de trabajo. Este tipo de diagrama se utiliza para describir el comportamiento y la lógica operativa de un sistema o proceso de una manera clara y estructurada.

En un diagrama de actividades, las actividades se representan como cajas o formas, y las flechas indican la secuencia de ejecución de estas actividades. Además, se pueden utilizar símbolos como rombos para representar decisiones y bifurcaciones en el flujo de trabajo. En general, el diagrama de actividades se centra en la representación de cómo se realizan las actividades, las condiciones que afectan a las decisiones y cómo fluye la información entre las actividades.

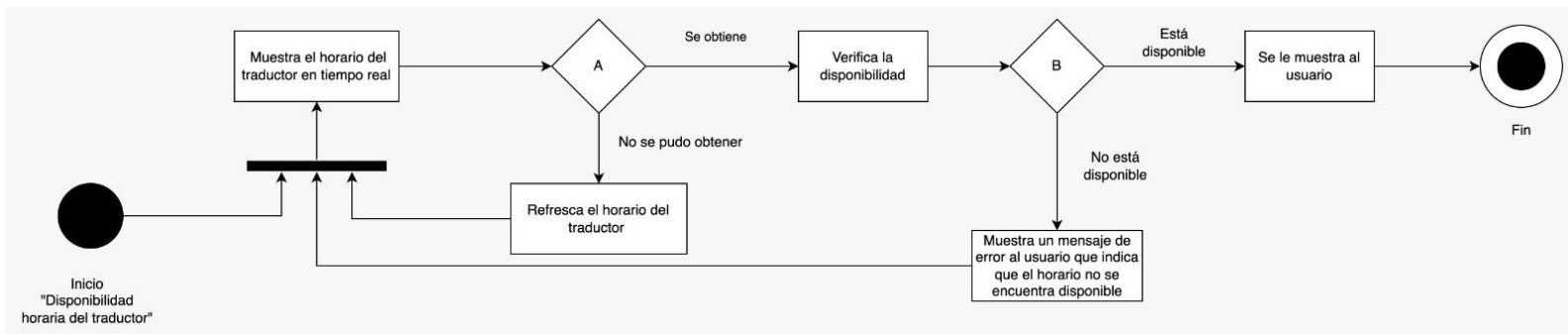
9.1.4.1 Diagrama 1



9.1.4.1.1 Documentación

Diagrama:	Actividades	
Transición:	Realizar contrato	
Descripción:	Serie de pasos de cómo realizar un contrato de un traductor	
Componente	Tipo	Descripción
Seleccionar rango horario que cubrirá el contrato	Actividad	Corresponde a la actividad donde un usuario selecciona un rango horario que comprende la duración de un contrato o servicio de traducción que desea solicitar
Ejecutar acción de contratar	Actividad	Corresponde a la actividad donde un usuario despues de seleccionar el rango horario para el cual desea contratar el servicio de traducción confirma la selección al dar click en el boton "Confirmar" y la solicitud es enviada al servicio de contratación para ser procesada
Validar que no haya un contrato creado o en proceso	Actividad	Corresponde a la actividad donde el sistema valida que el rango horario seleccionado para el contrato esté disponible o que no haga parte de otro contrato creado o en proceso
A	Decisión	Punto de desición que permite saber que flujo se debe tomar dependiendo del resultado de la validación del rango horario seleccionado
Registrar el contrato	Actividad	Corresponde a la actividad donde el sistema registra el contrato que el usuario desea realizar
Muestra un mensaje de error al usuario, indicando que ya existe un contrato creado o en proceso	Actividad	Corresponde a la actividad donde el sistema muestra al usuario un mensaje informando del error o problema registrando el contrato con el rango horario seleccionado
Muestra un mensaje exitoso al usuario indicando que el contrato se realizó exitosamente	Actividad	Corresponde a la actividad donde el sistema muestra al usuario un mensaje que le informa qe el registro del contrato se realizó correcta y exitosamente

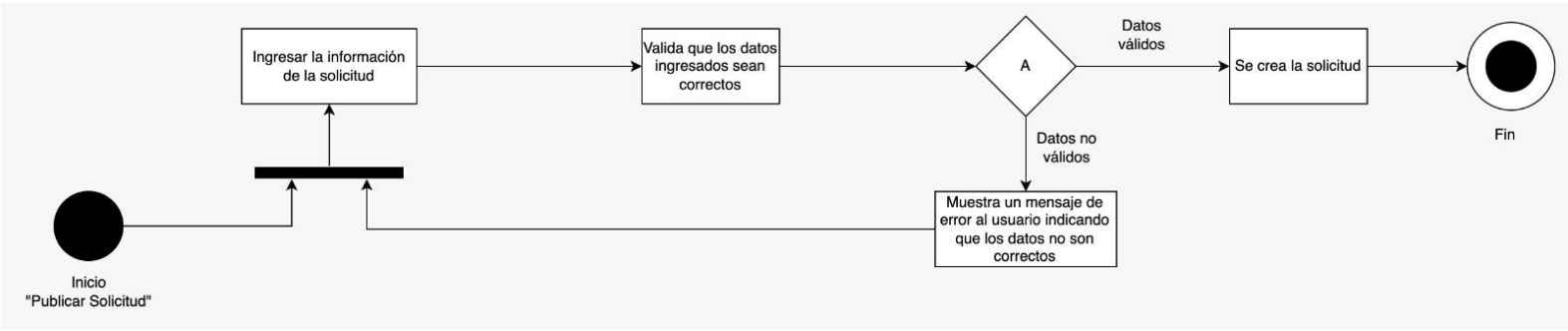
9.1.4.2 Diagrama 2



9.1.4.2.1 Documentación

Diagrama:	Actividades	
Transición:	Mostrar disponibilidad horario del traductor	
Descripción:	Modela el flujo de actividades/operaciones que se deben llevar a cabo para mostrar la disponibilidad del traductor	
Componente	Tipo	Descripción
Muestra el horario del traductor en tiempo real	Actividad	Corresponde a la actividad donde el sistema muestra al usuario el horario o disponibilidad de un traductor para realizar contratos o servicios de traducción
A	Decisión	Punto de decisión que permite saber que flujo se debe tomar dependiendo de si se puede obtener la información nueva
Verifica la disponibilidad	Actividad	Corresponde a la actividad donde el sistema valida que la información entrante sea verídica y coherente a la disponibilidad real del traductor
Refresca el horario del traductor	Decisión	Corresponde a la actividad donde el sistema refresca o actualiza la información del horario o disponibilidad del traductor, reemplazando la información previamente mostrada por la nueva
B	Decisión	Punto de decisión que permite saber que flujo se debe tomar dependiendo de la llegada de una actualización del horario de un traductor
Se le muestra al usuario el horario del traductor	Actividad	Corresponde a la actividad donde el sistema le muestra al usuario el nuevo horario del traductor
Muestra un mensaje de error al usuario que indica que el horario no se encuentra disponible	Actividad	Corresponde a la actividad donde el sistema le muestra al usuario un mensaje indicando que hay un problema con la actualización de la disponibilidad de un traductor por lo que es posible que la información mostrada sea errónea

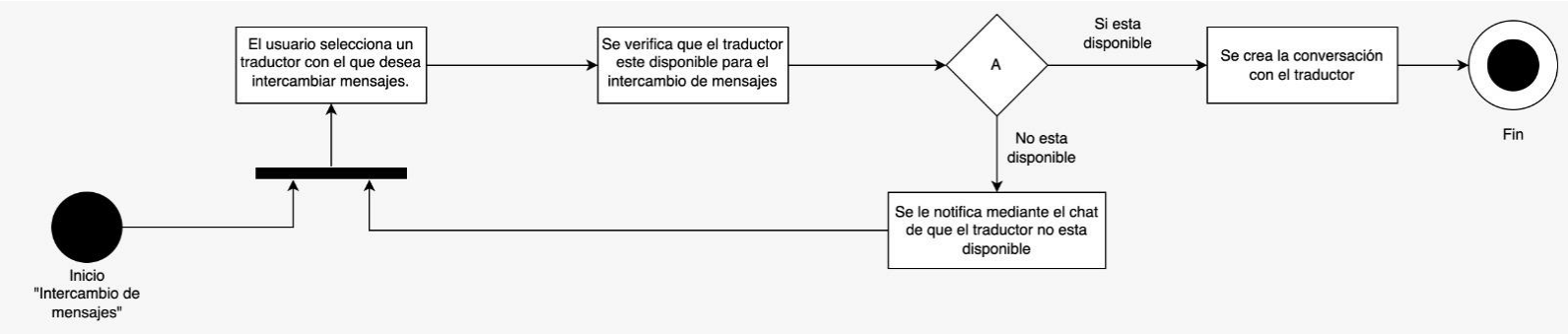
9.1.4.3 Diagrama 3



9.1.4.3.1 Documentación

Diagrama:	Actividades	
Transición:	Publicar solicitud	
Descripción:	Modela el flujo de actividades/operaciones que se deben llevar a cabo para Publicar una solicitud	
Componente	Tipo	Descripción
Ingresar la información de la solicitud de traducción	Actividad	Corresponde a la actividad donde el usuario ingresa la información de una solicitud o necesidad de traducción
Valida que los datos requeridos sean correctos	Actividad	Corresponde a la actividad donde el sistema verifica que los datos ingresados por el usuario sean válidos
A	Decisión	Punto de decisión que permite saber que flujo se debe tomar dependiendo de si los datos son válidos o no
Se crea la solicitud	Actividad	Corresponde a la actividad donde el sistema procesa y registra la solicitud creada por el usuario
Muestra un mensaje de error al usuario, indicando que los datos ingresados no son válidos	Actividad	Corresponde a la actividad donde el usuario ingresa la información relacionada con el nuevo país. En esta actividad a nivel de experiencia de usuario, se garantiza que los datos se validen en tiempo real, asegurando que sea muy amigable e intuitiva la interacción con el usuario.

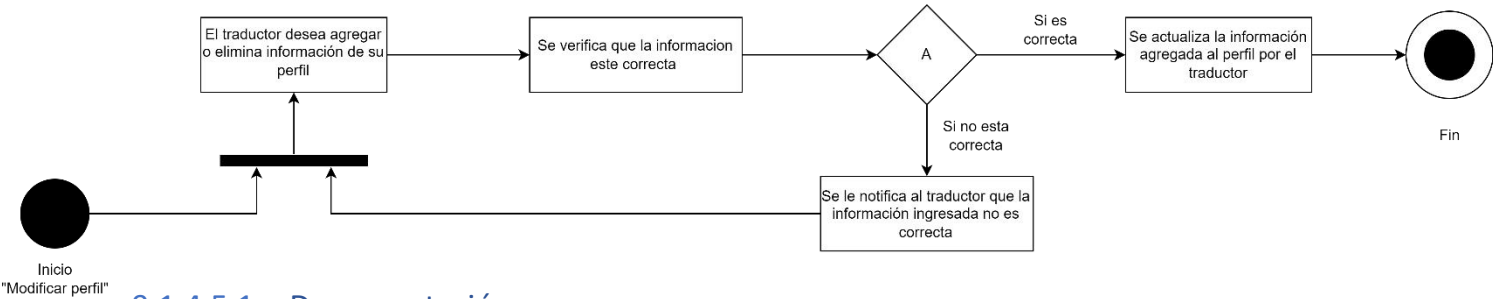
9.1.4.4 Diagrama 4



9.1.4.4.1 Documentación

Diagrama:	Actividades	
Transición:	Intercambio de mensajes	
Descripción:	Modela el flujo de actividades/operaciones que se deben llevar a cabo para registrar información de un nuevo país	
Componente	Tipo	Descripción
El usuario Selecciona un traductor con el que desea intercambiar mensajes	Actividad	Corresponde a la actividad donde el usuario selecciona un traductor. En esta actividad a nivel de experiencia de usuario, se garantiza que los datos se validen en tiempo real, asegurando que sea muy amigable e intuitiva la interacción con el usuario.
Se verifica que el traductor esté disponible para el intercambio de mensajes	Actividad	Corresponde a la actividad donde el Sistema valida que el traductor esté disponible. En esta actividad a nivel de experiencia de usuario, se garantiza que los datos se validen en tiempo real, asegurando que no se pierda el tiempo del usuario.
Se crea la conversación con el traductor	Actividad	Corresponde a la actividad donde se crea la conversación del usuario y el trauctor seleccionado
A	Decisión	Punto de desición que permite saber que flujo se debe tomar dependiendo de si se puede obtener la información nueva
Se le notifica mediante el chat de que el traductor no está disponible	Actividad	Corresponde a la actividad donde el sistema mediante el chat le muestra al usuario que el traductor con quien trata de comunicarse no está disponible

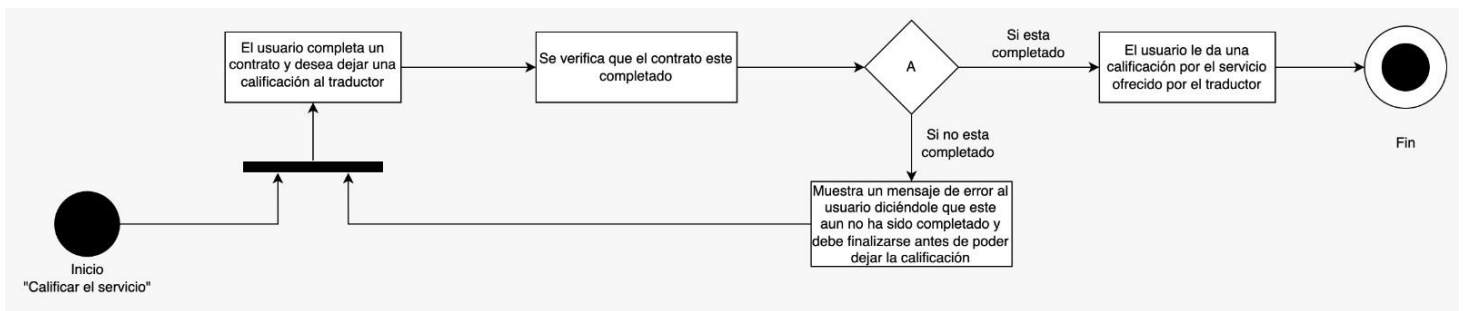
9.1.4.5 Diagrama 5



9.1.4.5.1 Documentación

Diagrama:	Actividades	
Transición:	Modificar perfil	
Descripción:	Modela el flujo de actividades/operaciones que se deben llevar a cabo para modificar el perfil	
Componente	Tipo	Descripción
El traductor desea agregar o eliminar información de su	Actividad	Corresponde a la actividad donden el traductor desea modificar el perfil el perfil ya sea añadiendo o eliminando información
Se verifica que la información este correcta	Actividad	Corresponde a la actividad donde el sistema realiza una verificación para confirmar que la información ingresada o eliminada haya sido correcto haciendo referencia al tipo de dato
A	Decision	Punto de decision que permite saber qué flujo se debe tomar dependiendo de si la información agregada o eliminada sea
Se le notifica al traductor que la información ingresada no es correcta	Actividad	Corresponde a la actividad donde el sistema el muestra un mensaje de error al usuario porque el contrato no ha sido terminado y que debe finaizarlo primero para que pueda dejar
Se actualiza la información agregada al perfil por el	Actividad	Corresponde a la actividad donde el usuario le deja una calificacion al traductor luego de haber finalizado el contrato

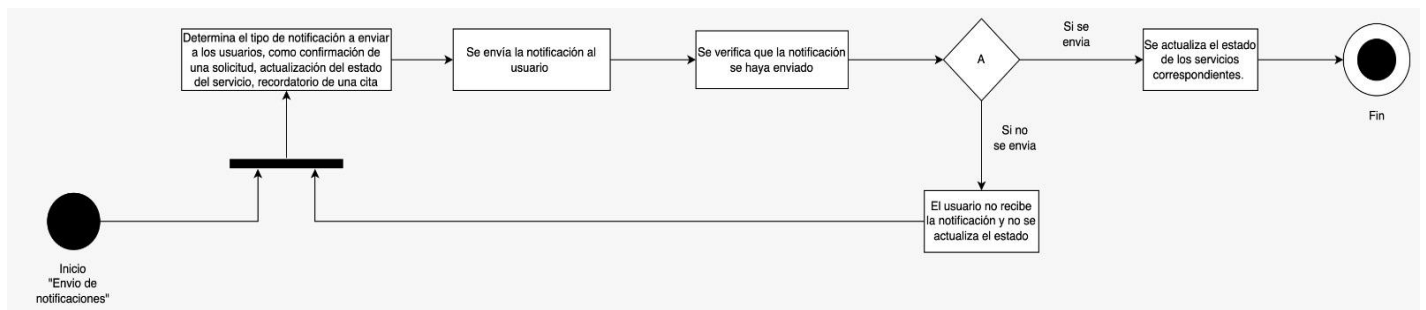
9.1.4.6 Diagrama 6



9.1.4.6.1 Documentación

Diagrama:	Actividades	
Transición:	Calificar el servicio	
Descripción:	Modela el flujo de actividades/operaciones que se deben llevar a cabo para calificar el servicio contratado	
Componente	Tipo	Descripción
El usuario completa un contrato y desea dejar una calificación al traductor	Actividad	Corresponde a la actividad donden usuario ha concluido satisfactoriamente un contrato de servicios de traducción y, como parte de su interacción con la plataforma o el servicio, tiene la oportunidad de expresar su opinión sobre la calidad y la satisfacción general de la traducción realizada.
Se verifica que el contrato este completado	Actividad	Se realiza una revisión o verificación para confirmar que un contrato específico ha sido cumplido de acuerdo con los términos y condiciones acordados entre las partes involucradas.
A	Decision	Punto de decision que permite saber qué flujo se debe tomar dependiendo de la verificacion del contrato ya sea que este finalizado o no
Muestra un mensaje de error al usuario diciéndole que este aun no ha sido completado y debe finalizarse antes de poder dejar la calificación	Actividad	Corresponde a la actividad donde el sistema el muestra un mensaje de error al usuario porque el contrato no ha sido terminado y que debe finaizarlo primero para que pueda dejar una calificación
El usuario le da una calificación por el servicio ofrecido por el traductor	Actividad	Corresponde a la actividad donde el usuario le deja una calificacion al traductor luego de haber finalizado el contrato

9.1.4.7 Diagrama 7



9.1.4.7.1 Documentación

Diagrama:	Actividades	
Transición:	Envío de notificaciones	
Descripción:	Modela el flujo de actividades/operaciones que se deben llevar a cabo para enviar una notificación	
Componente	Tipo	Descripción
Determina el tipo de notificación a enviar a los usuarios, como confirmación de una solicitud, actualización del estado del servicio,	Actividad	Corresponde a la actividad donde el sistema determina que tipo de notificación se le va a enviar al usuario. En esta actividad a nivel de experiencia de usuario, se garantiza que las notificaciones se realicen en el momento que surja alguna novedad.
Se envía la notificación al usuario	Actividad	Corresponde a la actividad donde el sistema envía la notificación al usuario.
Se verifica que la notificación se haya enviado	Actividad	Corresponde a la actividad donde el sistema verifica que la notificación sea correcta y haya sido enviada al usuario haciendo referencia al tipo de notificación para que esta pueda
A	Decision	Punto de decisión que permite saber qué flujo se debe tomar de habersido enviada o no la notificación al usuario
El usuario no recibe la notificación y no se actualiza el estado	Actividad	Corresponde a la actividad donde el sistema verifica que la notificación no fue enviada al usuario y que esta no procedió a actualizar su estado.
Se actualiza el estado de los servicios correspondientes.	Actividad	Corresponde a la actividad donde sistema actualiza el estado de los servicios correspondientes al tipo de notificación enviada al

9.1.5 Incepción Ágil

Un Agile Inception, también conocido como "inception phase" o "fase de inicio ágil", es un conjunto de actividades y reuniones iniciales que tienen como objetivo establecer una comprensión compartida del proyecto entre los equipos de desarrollo y los stakeholders, así como definir las bases y los principios para guiar el desarrollo del proyecto.

El propósito principal de un Agile Inception es alinear a todas las partes interesadas en torno a la visión, los objetivos y los requisitos del proyecto, y proporcionar una dirección clara para el equipo de desarrollo

9.1.5.1 Por qué estamos aquí

Estamos aquí para abordar una necesidad en el mercado de servicios de traducción. La aplicación Talklate nace para proporcionar una solución eficiente y accesible que facilite la comunicación presencial entre usuarios que hablan diferentes idiomas y traductores calificados, además estamos aquí para crear una aplicación que genere valor para los usuarios. Talklate permitirá a las personas acceder a servicios de traducción de alta calidad de manera conveniente y asequible, mejorando así sus viajes y Talklate también está diseñado para crear oportunidades para personas que sepan diferentes idiomas. Les permite ofrecer sus servicios de manera independiente y llegar a una audiencia más amplia.

9.1.5.2 Visión/Elevator Pitch

La visión de un proyecto es una declaración que describe el propósito y el resultado deseado del proyecto de manera clara y concisa. Es una imagen atractiva y convincente del futuro en el que el

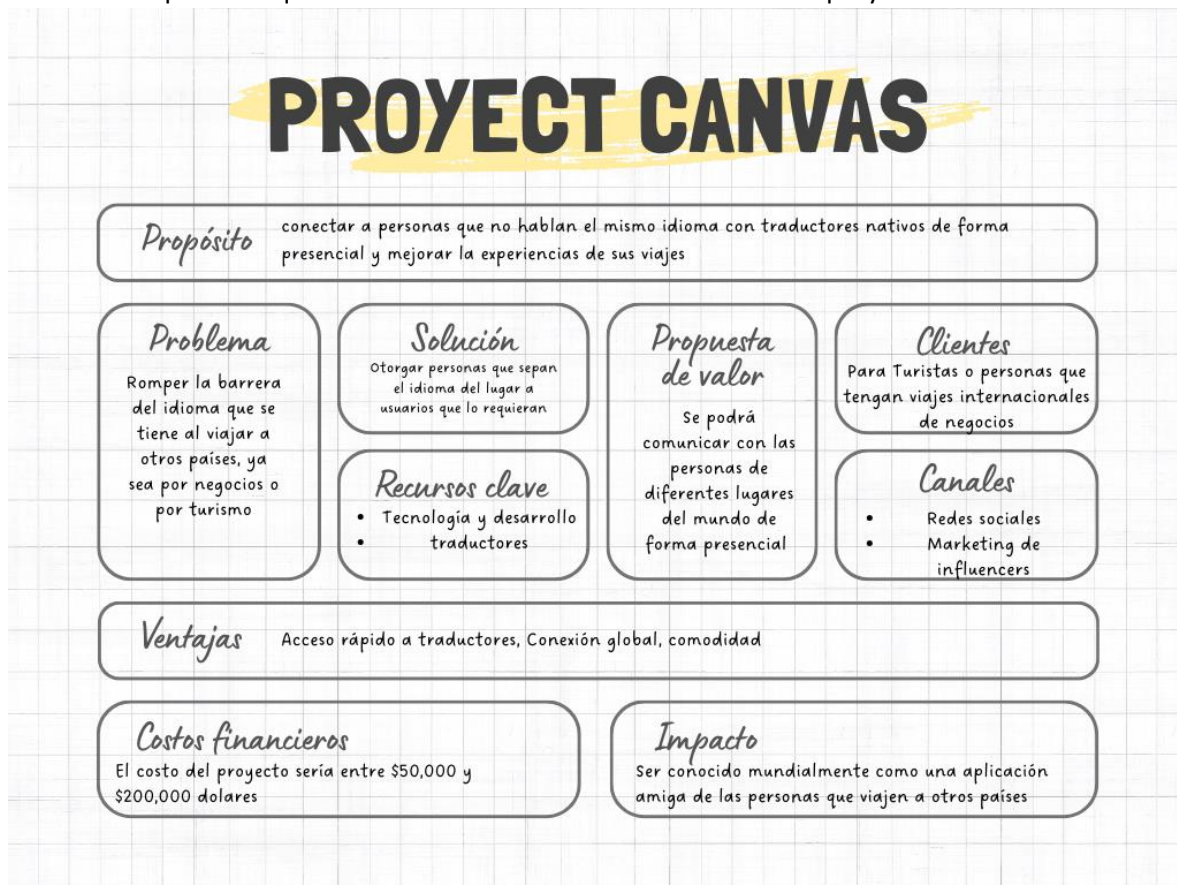
proyecto se materializa con éxito. La visión proporciona una dirección unificada y una guía para el desarrollo del proyecto y la definición del diseño.

9.1.5.2.1 Visión

TalkLate es una aplicación web/móvil que conecta a personas que no hablan el mismo idioma con traductores nativos de forma presencial. Ya sea que estés de viaje, trabajando en el extranjero o simplemente necesites comunicarte con alguien en otro idioma, TalkLate te brinda acceso a traductores locales para una comunicación más auténtica y precisa. A diferencia de las herramientas de traducción automática, nuestro enfoque se basa en el toque humano y la riqueza cultural de los traductores nativos

9.1.5.2.2 Project Canvas

El Project Canvas es una herramienta visual y estructurada que se utiliza para resumir y organizar la información clave de un proyecto en un formato claro y conciso. Se compone de diferentes secciones o "paneles" que contienen información relevante sobre el proyecto.



9.1.5.2.3 Mapa de impacto

Un mapa de impacto es una herramienta visual que se utiliza para identificar y representar de manera clara y concisa cómo ciertas acciones, eventos o cambios afectarán a un proyecto o a un conjunto de elementos relacionados. Este mapa muestra las relaciones causa-efecto entre

diversos factores y eventos, ayudando a comprender cómo una acción o un cambio puede influir en otros aspectos del proyecto.

9.1.5.3 Caja de producto

La caja de producto, en el contexto del desarrollo de software, se refiere a un conjunto de características y funcionalidades que se planea incluir en una versión específica de una aplicación o software. Es como una "caja" que contiene todas las características que se entregarán en una actualización o versión de la aplicación. La caja de producto define lo que estará disponible para los usuarios en esa versión en particular.

Esta caja de producto está compuesta por:

Registro de Usuarios: La capacidad para que los usuarios se registren en la plataforma, creen perfiles y gestionen sus preferencias.

Chat en Vivo: Una interfaz de chat en vivo que permita a los usuarios interactuar con los traductores.

Selección de Idioma: Una forma sencilla para que los usuarios elijan los idiomas en los que desean comunicarse.

Calificación y Comentarios: Un sistema que permita a los usuarios calificar a los traductores y dejar comentarios sobre sus servicios.

Sistema de Pago y Tarifas: La capacidad de realizar pagos de manera segura y permitir a los traductores establecer sus tarifas.

Geolocalización: Funcionalidad de geolocalización para encontrar traductores cercanos.

Gestión de Perfil de Traductor: Herramientas para que los traductores gestionen sus perfiles, experiencia y disponibilidad.

Notificaciones en Tiempo Real: Notificaciones instantáneas sobre nuevas solicitudes de traducción y mensajes de los usuarios.

Mensajería Interna: Un sistema de mensajería interna para la comunicación fluida entre usuarios y traductores.

Accesibilidad: Características de accesibilidad, como lectores de pantalla, para usuarios con discapacidades visuales.

Historial de Conversaciones: La posibilidad de acceder y revisar conversaciones pasadas con traductores.

Búsqueda Avanzada de Traductores: Herramientas de búsqueda que permiten a los usuarios encontrar traductores específicos.

Gestión de Proyectos: Funcionalidades que ayudan a los traductores a gestionar múltiples proyectos de manera eficiente.

Sistema de Valoración de Usuarios: Permite a los traductores calificar a los usuarios según su comportamiento y puntualidad en los pagos.

Seguridad y Privacidad de Datos: Garantiza la seguridad de los datos personales y las conversaciones de los usuarios.

Integración de Pasarela de Pagos: Facilita pagos seguros a través de múltiples métodos de pago.

Noticias y Anuncios: Sección que proporciona información relevante y actualizaciones a los usuarios.

Gestión de Contenido y Reportes: Herramientas para que los administradores gestionen el contenido generado por los usuarios y generen informes de actividad.

9.1.5.4 Lo que sí, lo que no

se refiere a una lista de características o funcionalidades que se incluirán en una aplicación o proyecto (lo que sí) y una lista de características que se omitirán o no se incluirán (lo que no). Esta lista ayuda a definir claramente lo que se planea desarrollar y lo que se considera fuera del alcance del proyecto.

El proceso de identificar "lo que sí y lo que no" es importante en la gestión de proyectos y en el desarrollo de software, ya que ayuda a establecer expectativas claras y a definir los límites del proyecto. También ayuda a evitar el alcance excesivo del proyecto, lo que puede llevar a retrasos y costos adicionales.

Y sería:

Lo que sí:

Traducción en tiempo real: Los usuarios podrán comunicarse en diferentes idiomas en tiempo real.

Registro de Usuarios: Los usuarios podrán registrarse en la plataforma y crear perfiles.

Chat en Vivo: Interfaz de chat en vivo para la comunicación entre usuarios y traductores.

Calificación y Comentarios: Los usuarios podrán calificar a los traductores y dejar comentarios sobre sus servicios.

Sistema de Pago y Tarifas: Capacidad para realizar pagos seguros y permitir que los traductores establezcan sus tarifas.

Geolocalización: Funcionalidad de geolocalización para encontrar traductores cercanos.

Mensajería Interna: Sistema de mensajería interna para una comunicación fluida entre usuarios y traductores.

Accesibilidad: Características de accesibilidad, como lectores de pantalla, para usuarios con discapacidades visuales.

Historial de Conversaciones: La posibilidad de acceder y revisar conversaciones pasadas con traductores.

Lo que no:

Lector de Texto para Discapacidad Visual: Esta característica se omitirá inicialmente, pero podría considerarse en futuras actualizaciones.

Búsqueda Avanzada de Traductores: Una función de búsqueda avanzada puede ser más compleja y se podría posponer para futuras versiones.

Noticias y Anuncios: La gestión de contenido adicional, como noticias y anuncios, podría no estar disponible en la versión inicial.

Gestión de Contenido y Reportes: Herramientas para la gestión de contenido y generación de informes podría ser una adición futura.

Integración de Pasarela de Pagos: La integración con múltiples métodos de pago puede requerir más tiempo y recursos.

9.1.5.5 Mapa de historias de usuario

El mapa de historias de usuario es una herramienta visual que se utiliza en la gestión de proyectos y desarrollo de software para organizar y representar las historias de usuario de una manera coherente y comprensible. Las historias de usuario son descripciones breves de las funcionalidades o características que los usuarios desean en una aplicación o proyecto.

9.1.5.6 Product Backlog Item

El Product Backlog es una lista dinámica y priorizada de todas las características, funcionalidades, mejoras y requisitos que se desean implementar en un proyecto. Cada elemento en el Product Backlog se denomina "Product Backlog Item" y representa un trabajo específico que debe realizarse. Los PBIs se mantienen priorizados de acuerdo con su importancia y valor para el producto.

Título: Configuración del servidor de base de datos

Descripción: Configurar el servidor de base de datos que respaldará la funcionalidad de almacenamiento de conversaciones y datos de usuarios en el back-end de Talklate.

Criterios de Aceptación:

- A. Se ha seleccionado y configurado un sistema de gestión de base de datos compatible con los requisitos del proyecto.
- B. Se ha implementado una estructura de base de datos que incluye tablas para almacenar conversaciones, usuarios y otros datos relevantes.
- C. La configuración de la base de datos garantiza un rendimiento óptimo y una alta disponibilidad.
- D. Se han establecido políticas de seguridad y copias de seguridad para proteger la integridad de los datos.
- E. Se ha realizado una prueba de conexión exitosa entre la aplicación y la base de datos configurada.

9.1.5.7 La comunidad

La "comunidad del proyecto" se refiere a un grupo de personas interesadas o involucradas en el proyecto que comparten un interés común en su éxito y desarrollo. Esta comunidad puede incluir a los miembros del equipo de desarrollo, patrocinadores, usuarios, colaboradores voluntarios, expertos en la materia y otros interesados en el proyecto. La comunidad del proyecto desempeña un papel importante al aportar diferentes perspectivas, conocimientos y recursos que pueden beneficiar al proyecto en diversas formas.

Comunidad del Proyecto Talklate:

Equipo de Desarrollo: Incluye a desarrolladores, diseñadores, administradores de sistemas y otros profesionales involucrados en la creación y gestión de la aplicación.

Usuarios y Colaboradores: Personas que utilizan la aplicación Talklate, proporcionan retroalimentación, contribuyen al desarrollo y participan en la comunidad.

Expertos en Lenguas y Cultura: Hablantes nativos y expertos en lenguas y culturas que contribuyen a la precisión de las traducciones y al entendimiento cultural.

Inversores o Patrocinadores: Personas o entidades que brindan apoyo financiero o recursos para el proyecto.

Comunidad de Desarrollo Ágil: Si el proyecto sigue una metodología ágil, la comunidad de desarrollo ágil incluirá Scrum Masters, Product Owners y otros roles relacionados.

Usuarios Potenciales: Personas que aún no utilizan la aplicación pero pueden beneficiarse de ella una vez que esté en funcionamiento.

9.1.5.8 La solución

La "solución" se refiere a la respuesta o enfoque que se implementará para abordar un problema o necesidad específica en un proyecto.

Solución (Aplicación Talklate): La aplicación web y móvil que permite a los usuarios comunicarse en varios idiomas con la ayuda de traductores nativos. La solución incluye características como chat, traducción en tiempo real, perfiles de usuarios y traductores, gestión de conversaciones, etc.

Usuarios Finales: Las personas que utilizan la aplicación Talklate para la comunicación multilingüe, ya sea por motivos de turismo, trabajo o cualquier otra razón. Esto incluye a los usuarios que viajan a lugares donde no hablan el idioma local y desean comunicarse con hablantes nativos.

Traductores Nativos: Personas que ofrecen servicios de traducción en la aplicación Talklate. Estos traductores son hablantes nativos del idioma local y brindan traducciones precisas y auténticas.

Base de Datos: La base de datos en la infraestructura del servidor que almacena información de usuarios, conversaciones, perfiles de traductores y otros datos relevantes.

Servidor de Aplicación: El servidor que aloja la aplicación Talklate y gestiona las solicitudes de usuarios, la autenticación, la traducción y la gestión de datos.

APIs Externas: Servicios externos que Talklate puede utilizar, como servicios de mapas para ubicación, servicios de pago para transacciones y servicios de autenticación.

Dispositivos Móviles y Navegadores Web: Los dispositivos utilizados por los usuarios para acceder a la aplicación Talklate, ya sea a través de una aplicación móvil o un navegador web.

9.2 Vista Lógica

Una vista lógica es una representación abstracta de un sistema de software que se centra en la lógica y la estructura de cómo funciona el sistema sin preocuparse por detalles de implementación o tecnología específica. La motivación detrás de una vista lógica es proporcionar una visión clara y coherente de la funcionalidad del sistema, independiente de los detalles técnicos, lo que facilita la comprensión y la comunicación sobre cómo opera el sistema.

9.2.1 Diagrama de clases

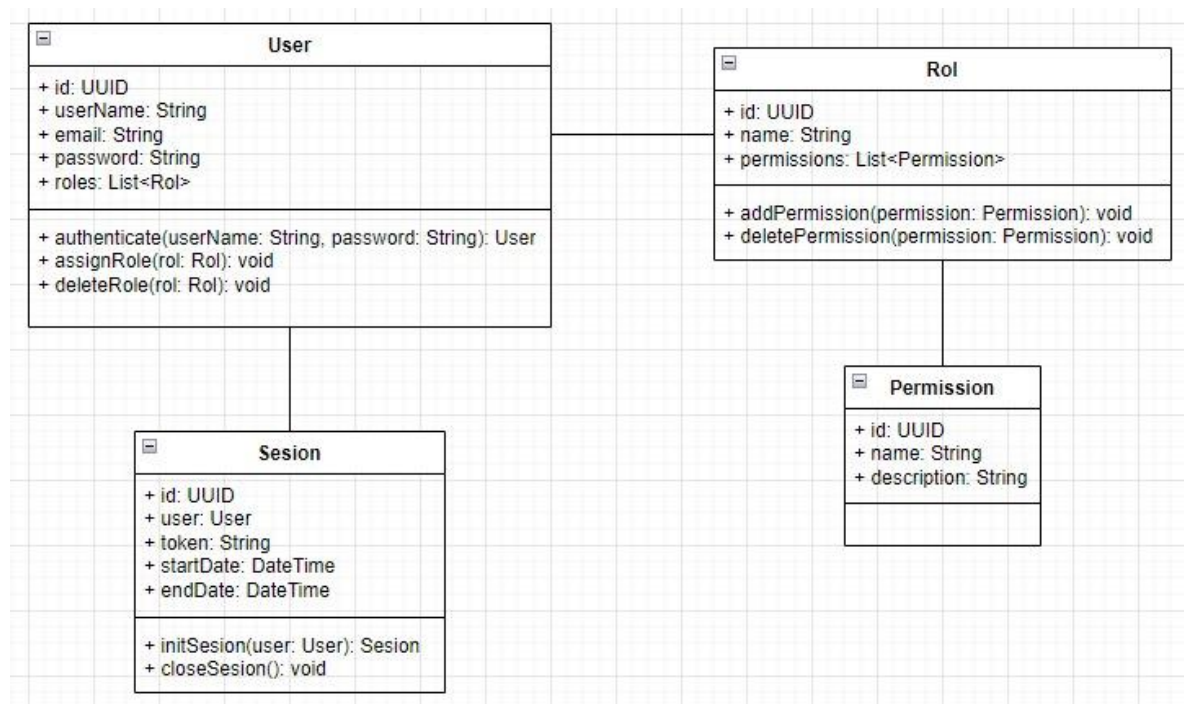
Un diagrama de clases es una representación visual que muestra la estructura de clases y objetos en un sistema de software, así como las relaciones entre ellos. En esencia, es una herramienta gráfica que representa la organización de las clases y sus interacciones en el sistema.

La motivación detrás de un diagrama de clases es proporcionar una vista clara y organizada de la estructura de clases en el sistema. Esto es esencial para comprender cómo se organizan las partes del software, cómo se relacionan entre sí y cómo contribuyen a la funcionalidad del sistema.

9.2.1.1 Componente 1

Diagrama de clases para microservicio talklate authentication

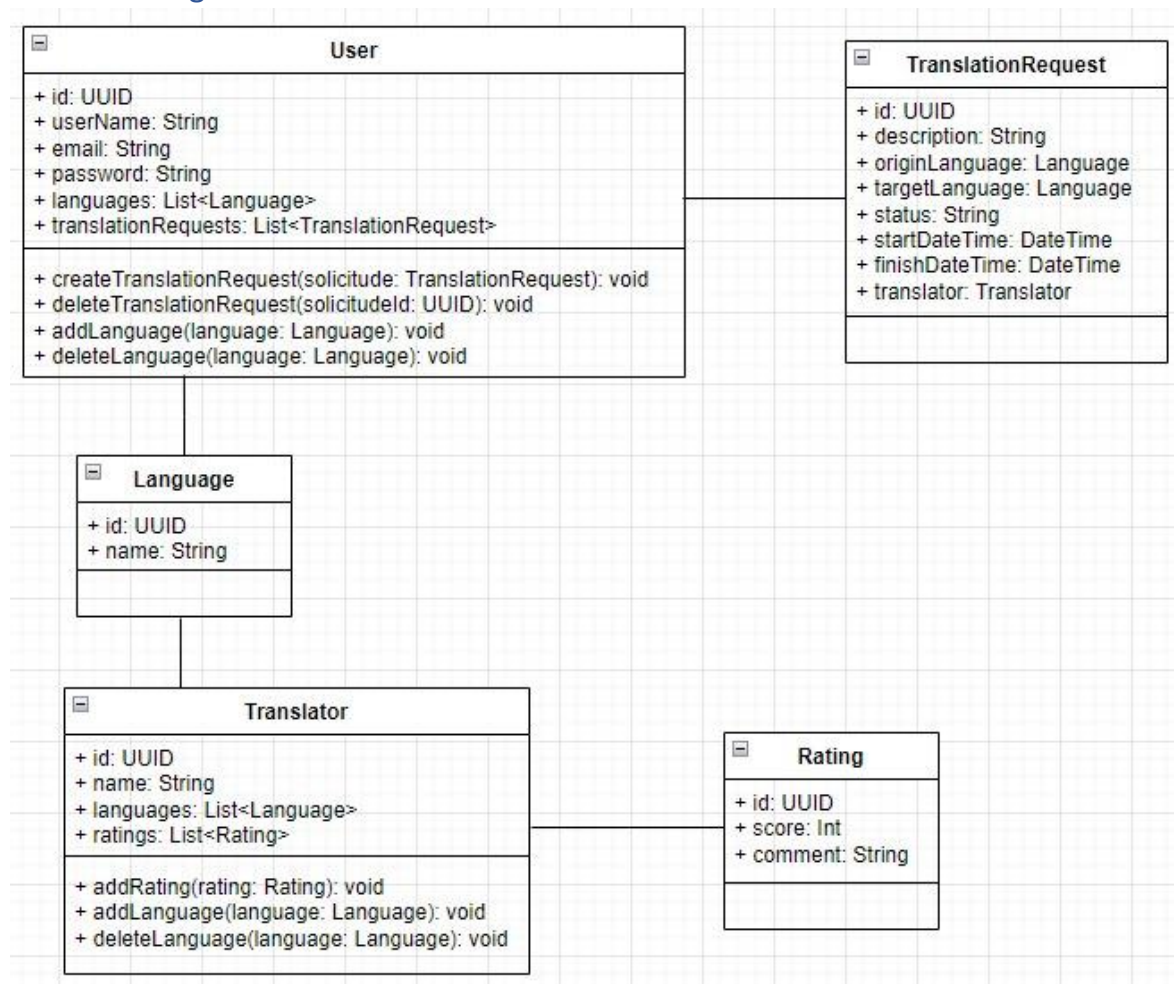
9.2.1.1.1 Diagrama



9.2.1.2 Componente 2

Diagrama de clases para microservicio talklate_management

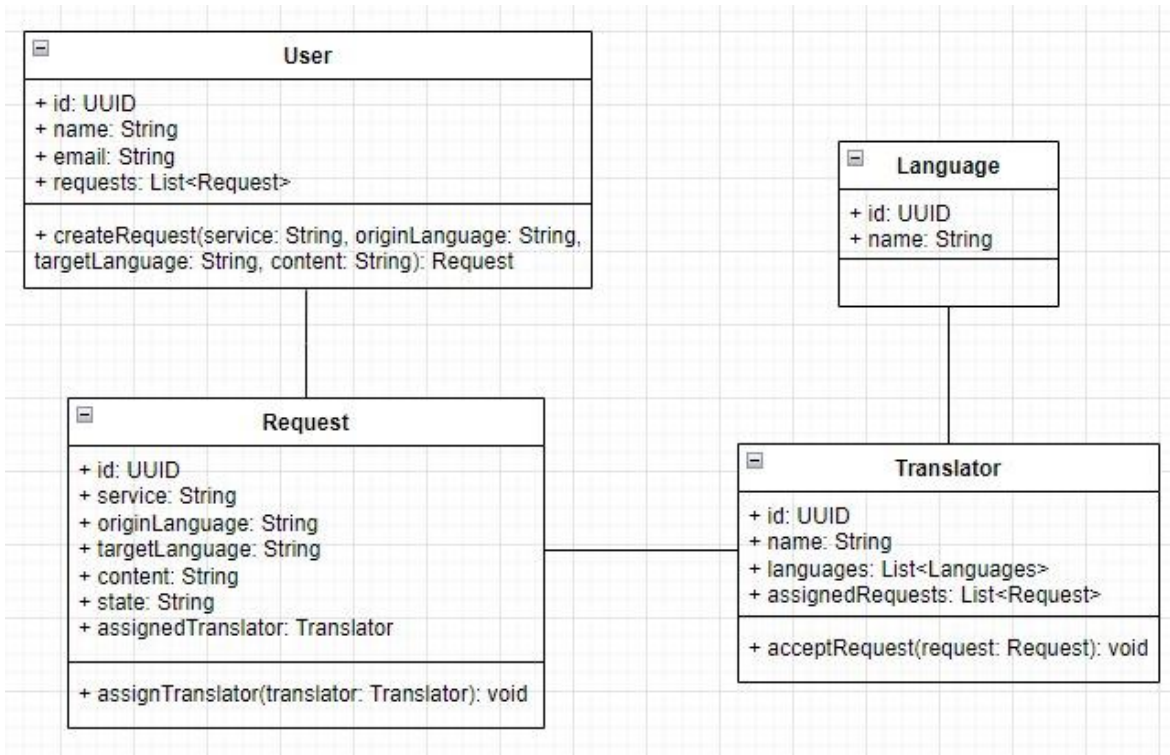
9.2.1.2.1 Diagrama



9.2.1.3 Componente 3

Diagrama de clases para el microservicio talklate_contract

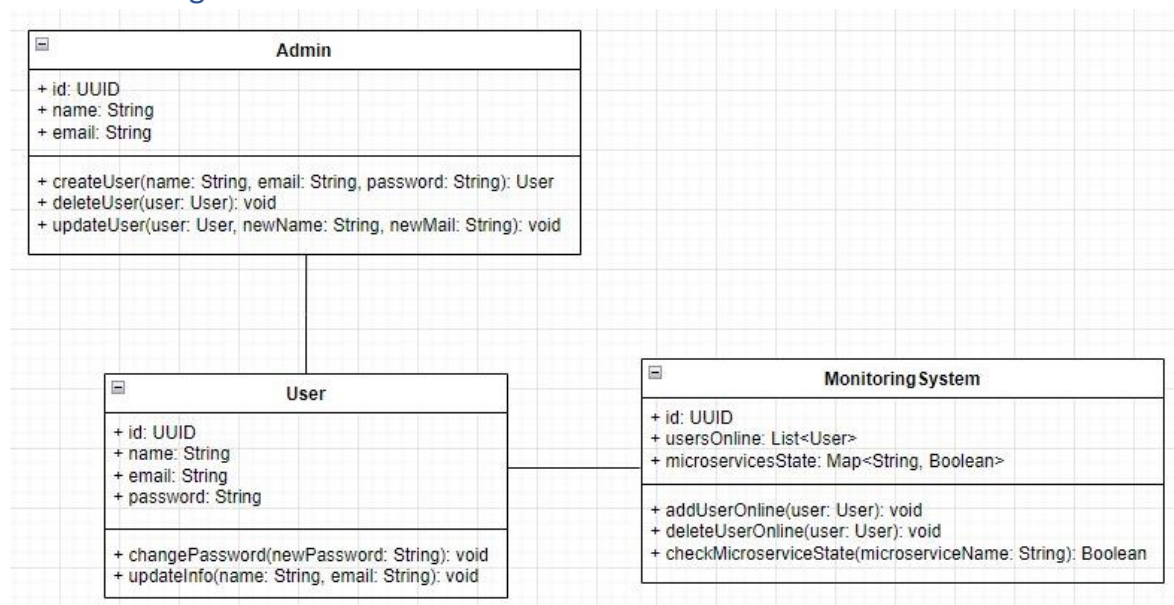
9.2.1.3.1 Diagrama



9.2.1.4 Componente 4

Diagrama de clases para el microservicio talklate_monitoring

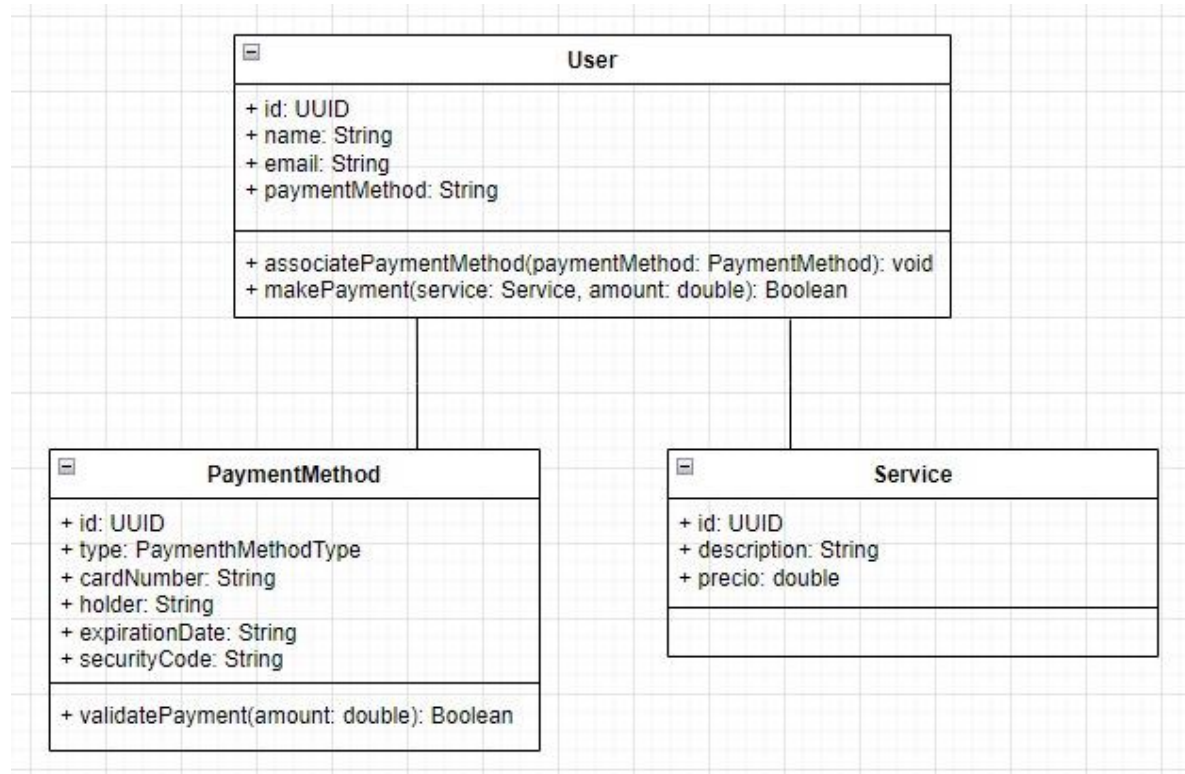
9.2.1.4.1 Diagrama



9.2.1.5 Componente 5

Diagrama de clases para microservicio talklate_payment

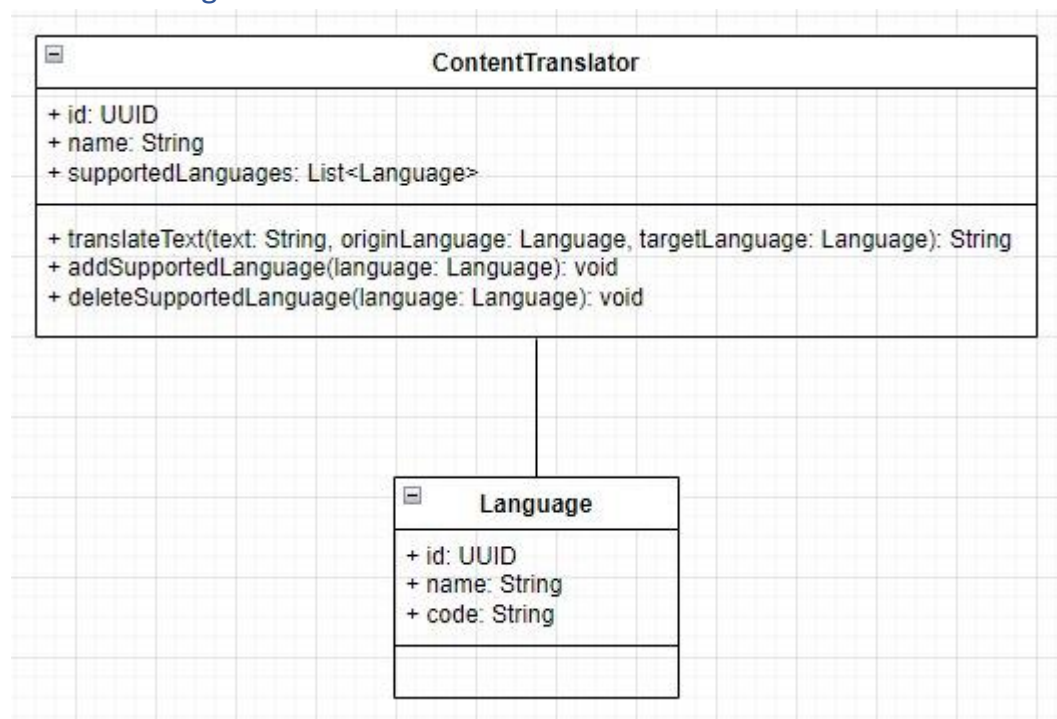
9.2.1.5.1 Diagrama



9.2.1.6 Componente 6

Diagrama de clases para el microservicio talklate_content_translator

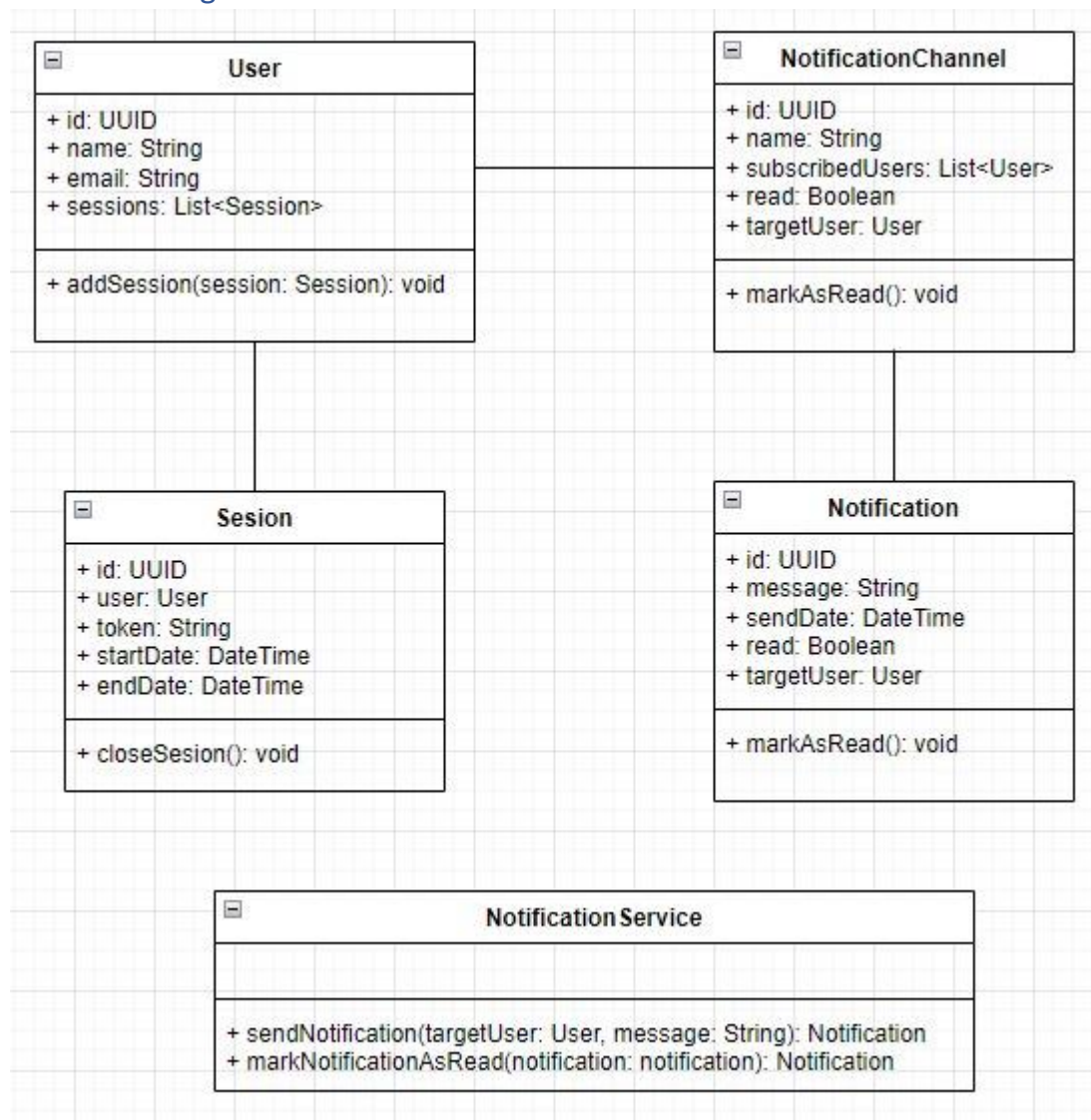
9.2.1.6.1 Diagrama



9.2.1.7 Componente 7

Diagrama de clases para microservicio talklate_notifications

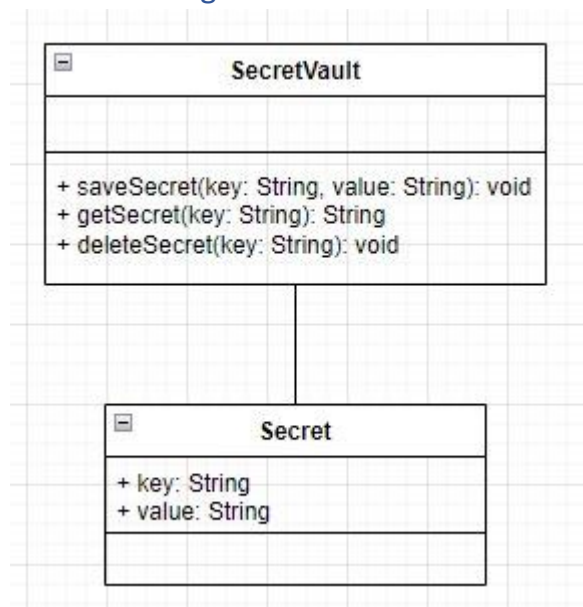
9.2.1.7.1 Diagrama



9.2.1.8 Componente 8

Diagrama de clases para el microservicio `talklate_vault`

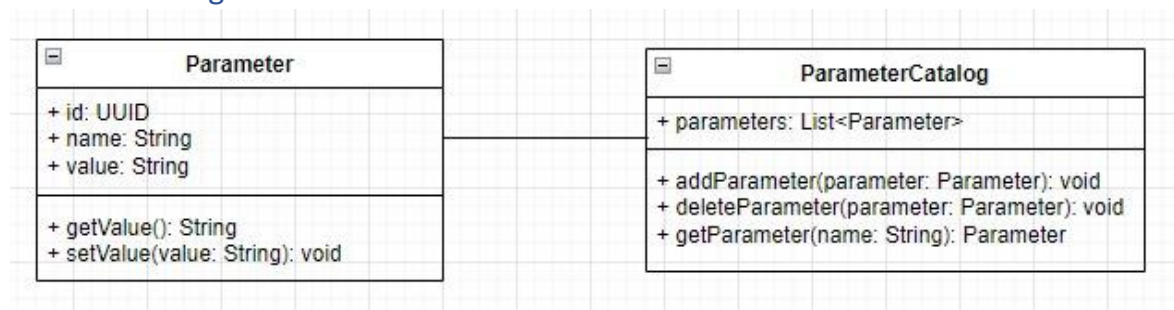
9.2.1.8.1 Diagrama



9.2.1.9 Componente 9

Diagrama de clases para el microservicio talklate_config

9.2.1.9.1 Diagrama



9.3 Vista de Despliegue/Vista de Desarrollo/Vista de Implementación

Una vista de despliegue es una representación visual que muestra cómo los componentes y artefactos de un sistema de software se despliegan o implementan en la infraestructura de hardware, incluyendo servidores, dispositivos y redes. Esta vista proporciona información detallada sobre la ubicación y la disposición física de los componentes del sistema en el entorno de implementación.

La motivación detrás de una vista de despliegue es brindar una vista clara y específica de cómo el sistema se ejecutará en el entorno físico. Esto es esencial para comprender cómo los componentes de software interactúan con la infraestructura de hardware y cómo se asegura la disponibilidad, el rendimiento y la escalabilidad del sistema.

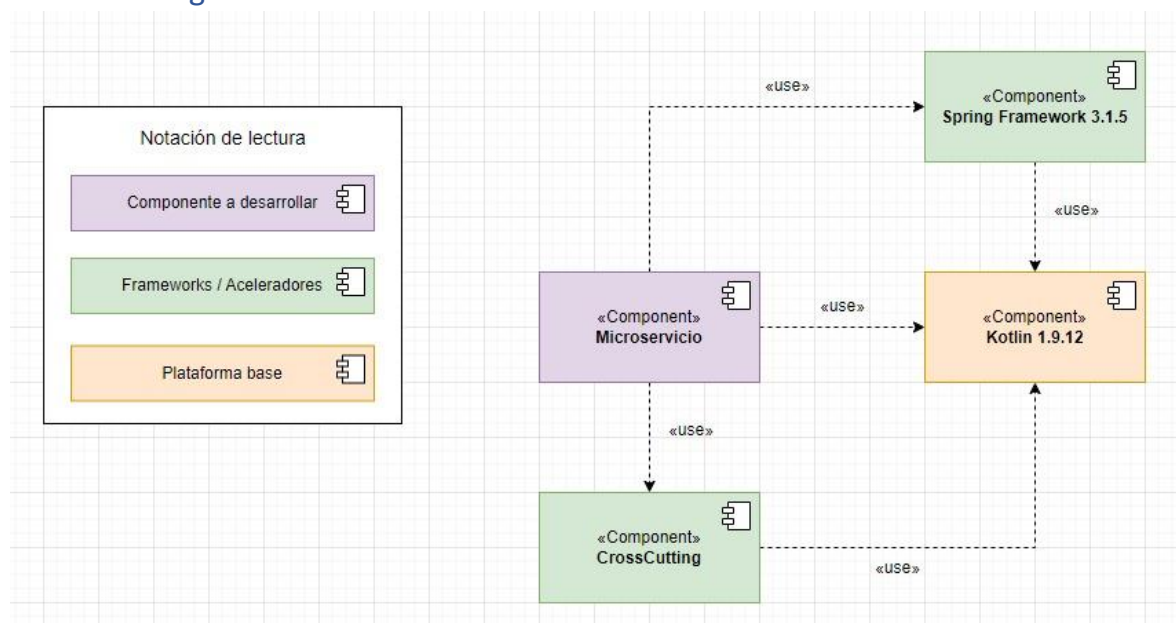
9.3.1 Diagrama de componentes

es una representación visual que muestra las partes fundamentales de un sistema de software y cómo se relacionan entre sí. En otras palabras, es un esquema gráfico que identifica los componentes clave del sistema y las interacciones o conexiones entre ellos.

9.3.1.1 Componente 1

Componente del Backend

9.3.1.1.1 Diagrama



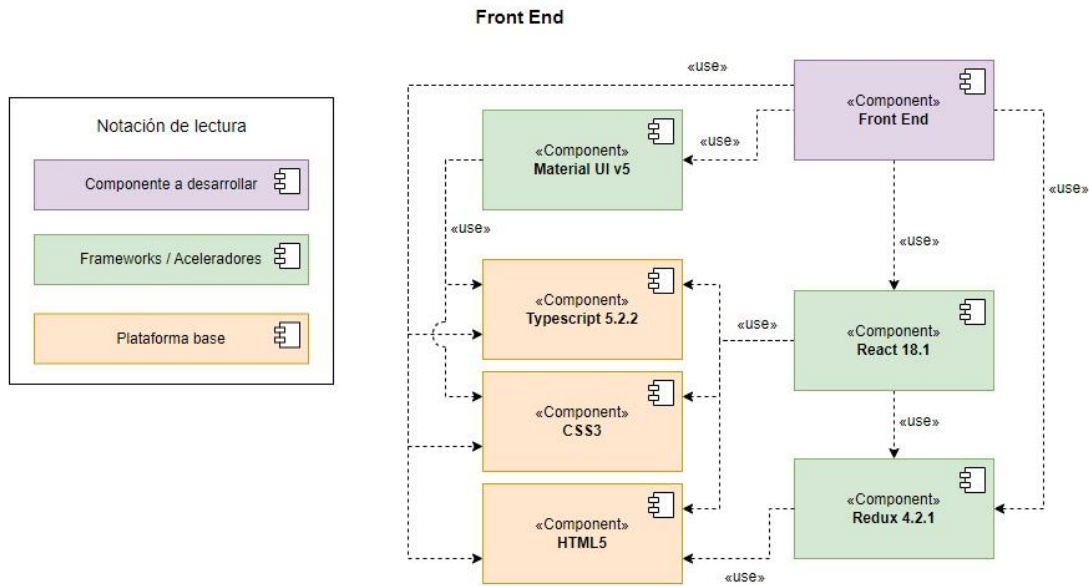
9.3.1.1.2 Documentación

Diagrama	Componentes			
Componente	Backend			
Descripción	Diagrama que muestra el 100% de los componentes que serán construidos o adoptados para proceder con la construcción del componente indicado. Adicionalmente indica como estos componentes están relacionados			
Componente:	Descripción	Motivación	Depende /Usa	Tipo Componente
Microservice	Componente que representa la línea base de cada microservicio que se construirá dentro de la aplicación de forma genérica	Apoya todo el dominio de negocio que internamente asegura que contenga todas las reglas y lógica del contexto que apalanca.	Crosscutting Spring Framework 3.1.5 Kotlin 1.9.12	Componente a desarrollar
CrossCutting	Componente que ofrece las funciones transversales que pueden ser utilizadas por cualquier componente microservicio de forma general	Ofrece una catidg de funciones, operaciones o aspectos que pueden ser utilizados de forma general, transversal, o común por cualquier otro componente desarrollado bajo la misma plataforma base (Kotlin 1.9.12), con el objetivo de fomentar la reutilización	Kotlin 1.9.12	Frameworks/Aceleradores
Spring Framework 3.1.5	Componente usado para acelerar el desarrollo de una aplicación, respecto a aquellos aspectos que no hacen parte del core de negocio	Acelerar el desarrollo y concetrar los esfuerzos dedl equipo en tareas y aspectos netamente del negocio.	Kotlin 1.9.12	Frameworks/Aceleradores
Kotlin 1.9.12	Plataforma base de desarrollo bajo el lenguaje Kotlin 1.9.12	Es un lenguaje multiplataforma que por su gran capacidad ha demostrado que puede ser utilizado para construir aplicaciones tanto livianas como empresariales con excelentes prestaciones de rendimiento y seguridad.		Plataforma base

9.3.1.2 Componente 2

Componente del Frontend

9.3.1.2.1 Diagrama



9.3.1.2.2 Documentación

Diagrama	Componentes			
Componente	Frontend			
Descripción	Diagrama que muestra el 100% de los componentes que serán construidos o adoptados para proceder con la construcción del componente indicado. Adicionalmente indica como estos componentes están relacionados			
Componente:	Descripción	Motivación	Depende /Usa	Tipo Componente
Frontend	Componente que representa la línea base del proyecto a nivel de contenido estatico y dinámico con el que interactuarán los usuarios finales y que se construirá dentro de la aplicación de forma genérica	Brinda a los usuarios la oportunidad de interactuar con la aplicación de manera simple e intuitiva mejorando la accesibilidad, usabilidad y experiencia general.	Typescript 5.2.2 CSS3 HTML5 React 18.1 Redux 4.2.1 Material UI v5	Componente a desarrollar
Material UI v5	Componente usado para acelerar el desarrollo de la parte interactiva de una aplicación	Ofrece la facilidad de crear interfaces de usuario amigables y atractivas de manera rapida y sencilla acelerando el desarrollo de la aplicación	Typescript 5.2.2 CSS3 HTML5	Frameworks/Aceleradores
React 18.1	Componente que permite crear interfaces de usuario interactivas y reutilizables acelerando el proceso de desarrollo al dejar de lado las complejidades del negocio.	Simplifica el desarrollo, mejora el rendimiento y ofrece una estructura modular para construir una aplicación interactiva y eficiente	Typescript 5.2.2 CSS3 HTML5 Redux 4.2.1	Frameworks/Aceleradores
Redux 4.2.1	Componente usado para la gestión del estado para aplicaciones Javascript/Typescript, especialmente aquellas basadas en React. Permite gestionar el flujo de datos de forma mas predecible y eficiente mediante un único almacén centralizado	Permite gestionar el flujo de datos de la aplicación proporcionando un enfoque estructurado y predecible, lo que permitirá entregar información actualizada y en tiempo real a los usuarios	Typescript 5.2.2	Frameworks/Aceleradores
Typescript 5.2.2	Plataforma base de desarrollo bajo el lenguaje Typescript 5.2.2	TypeScript mejora la seguridad, la claridad y la colaboración en el desarrollo de software, lo que lo convierte en una opción atractiva para muchos desarrolladores y equipos de desarrollo.		Plataforma base
CSS3	Plataforma base de estilos y diseño grafico bajo el lenguaje CSS3	Permite crear una experiencia de usuario visualmente agradable, facil de usar y accesible en diferentes dispositivos mejorando significativamente la calidad y la usabilidad de la aplicación		Plataforma base
HTML5	Plataforma base de desarrollo bajo el lenguaje HTML5.	Permite la creación de contenido web accesible y bien estructurado		Plataforma base

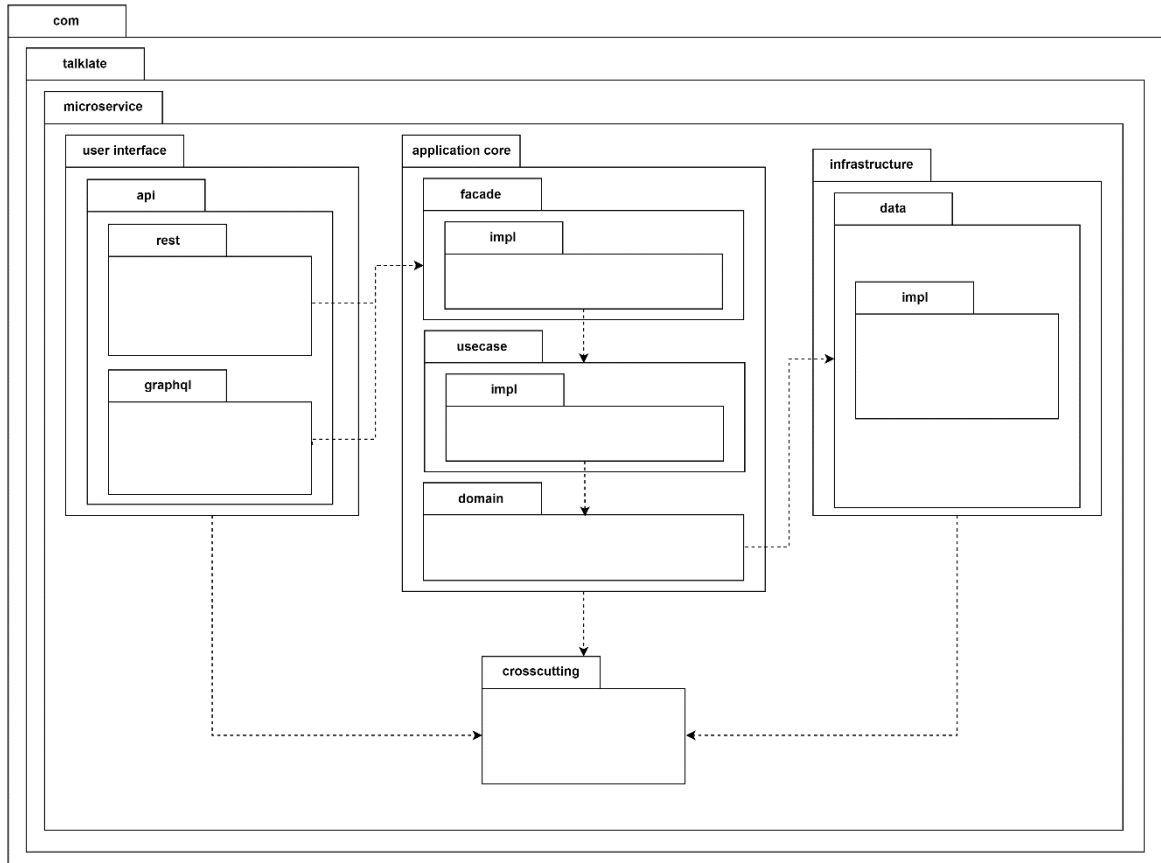
9.3.2 Diagrama de paquetes

representación visual que organiza y agrupa los elementos del sistema de software en paquetes o conjuntos lógicos. Cada paquete contiene elementos relacionados, como clases, subpaquetes u otros artefactos del sistema. La motivación detrás de un diagrama de paquetes es proporcionar una estructura organizativa que mejora la claridad y la gestión del diseño del sistema.

9.3.2.1 Componente 1

Diagrama que muestra la distribución de paquetes y la forma en que estos paquetes pueden usar la estructura de lo demás

9.3.2.1.1 Diagrama



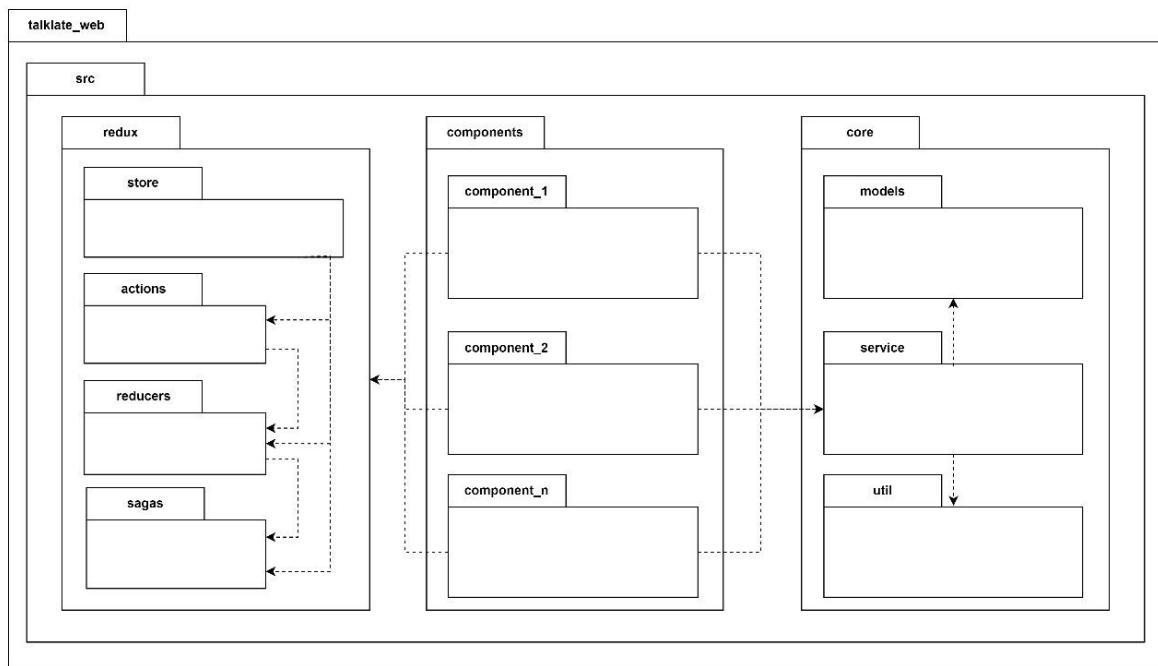
9.3.2.1.2 Documentación

Diagrama	Paquetes backend		
Componente:	Arquetipo general		
Descripción	Diagrama que muestra la distribución de paquetes orientada a una estrategia por capas de una arquitectura limpia y la forma que estos paquetes pueden usar las estructuras de los demás		
Paquete	Paquete padre	Descripción	Usa/importa
com		Paquete raíz que indica o representa que la solución se desarrolla de forma global	
talklate	com	Paquete raíz que indica o representa que la solución es para el proyecto TalkLate	
microservice	talklate	Paquete raíz que indica que todos los paquetes que se encuentran dentro de él conforman parte de la solución	
userinterface	microservice	Paquete que agrupa todos los elementos de la capa userinterface de una arquitectura limpia para la aplicación microservice	crosscutting applicationcore
applicationcore	microservice	Paquete que agrupa todos los elementos de la capa applicationcore de una arquitectura limpia para la aplicación microservice	crosscutting infrastructure
infrastructure	microservice	Paquete que agrupa todos los elementos de la capa infrastructure de una arquitectura limpia para la aplicación microservice	crosscutting
crosscutting	microservice	Paquete que agrupa todos los elementos de la capa de crosscutting o transversal de una arquitectura limpia para la aplicación microservice	
api	applicationcore	Paquete que agrupa todas interfaces de programación de aplicación (API) de la aplicación microservice	
rest	api	Paquete que agrupa todas las implementaciones de tipo rest de la aplicación microservice	facade
graphql	api	Paquete que agrupa todas las implementaciones de tipo graphql de la aplicación microservice	facade
facade	applicationcore	Paquete que agrupa los contratos de acceso a la capa applicationcore de la aplicación microservice	
usecase	applicationcore	Paquete que agrupa los contratos de acceso a usecase de la capa applicationcore de la aplicación microservice	
domain	applicationcore	Paquete que agrupa los metodos de domain de la capa applicationcore de la aplicación microservice	data
impl	facade	Paquete que agrupa todas las implementaciones de los contratos de acceso a la capa applicationcore de la aplicación microservice	usecase
impl	usecase	Paquete que agrupa todas las implementaciones de los contratos de acceso a usecase de la capa applicationcore de la aplicación microservice	domain
data	infrastructure	Paquete que agrupa todos los contratos de acceso a la capa infrastructure de la aplicación microservice	
impl	data	Paquete que agrupa todas las implementaciones de los contratos de acceso a datos de la aplicación microservice	

9.3.2.2 Componente 2

Diagrama que muestra la distribución de paquetes orientada a una estrategia por capas de una arquitectura limpia y la forma que estos paquetes pueden usar las estructuras de los demás

9.3.2.2.1 Diagrama



9.3.2.2.2 Documentación

Diagrama	Paquetes frontend		
Componente:	Arquetipo general		
Descripción	Diagrama que muestra la distribución de paquetes y la forma en que estos paquetes pueden usar la estructura de lo demás		
Paquete	Paquete padre	Descripción	Usa/importa
talklate_front		Paquete raíz que indica o representa que la solución es para el proyecto TalkLate	
src	talklate_front	Paquete donde se agrupa todo el código fuente de la aplicación	
redux	src	Paquete que agrupa componentes y métodos que permiten gestionar el estado global de la aplicación	
components	src	Paquete que agrupa todos los componentes de presentación de React	
core	src	Paquete que contiene la lógica esencial de la aplicación TalkLate	
store	redux	Paquete donde se gestiona el almacén de estados de la aplicación	actions reducers sagas
actions	redux	Paquete que agrupa las acciones de Redux para modificar el estado	
reducers	redux	Paquete que contiene los Reducers para manejar acciones de la aplicación	
sagas	redux	Paquete que agrupa los métodos que permiten actualizar el estado de la aplicación	
component_n	components	Paquete componente de presentación de React	service
models	core	Paquete que contiene las clases que definen la estructura de los datos de la aplicación, como modelos de usuarios y traductores	
service	core	Paquete que contiene las clases responsables de la lógica de negocio y la comunicación con el backend	models util
util	core	Paquete que agrupa las utilidades generales que pueden ser compartidas en toda la aplicación, como validadores, formateadores y funciones de registro	

9.4 Vista de Procesos

Una vista de procesos es una representación visual que muestra cómo los diferentes procesos y tareas dentro de un sistema de software interactúan y se comunican entre sí. Esta vista se centra en la secuencia de actividades y operaciones que ocurren en el sistema, lo que incluye cómo los datos y la información fluyen a través de estos procesos.

La motivación detrás de una vista de procesos es proporcionar una comprensión clara y detallada de cómo funciona el sistema en términos de las actividades y tareas que se realizan. Esta vista es esencial para entender el flujo de trabajo, la lógica operativa y la interacción entre los diferentes componentes del sistema.

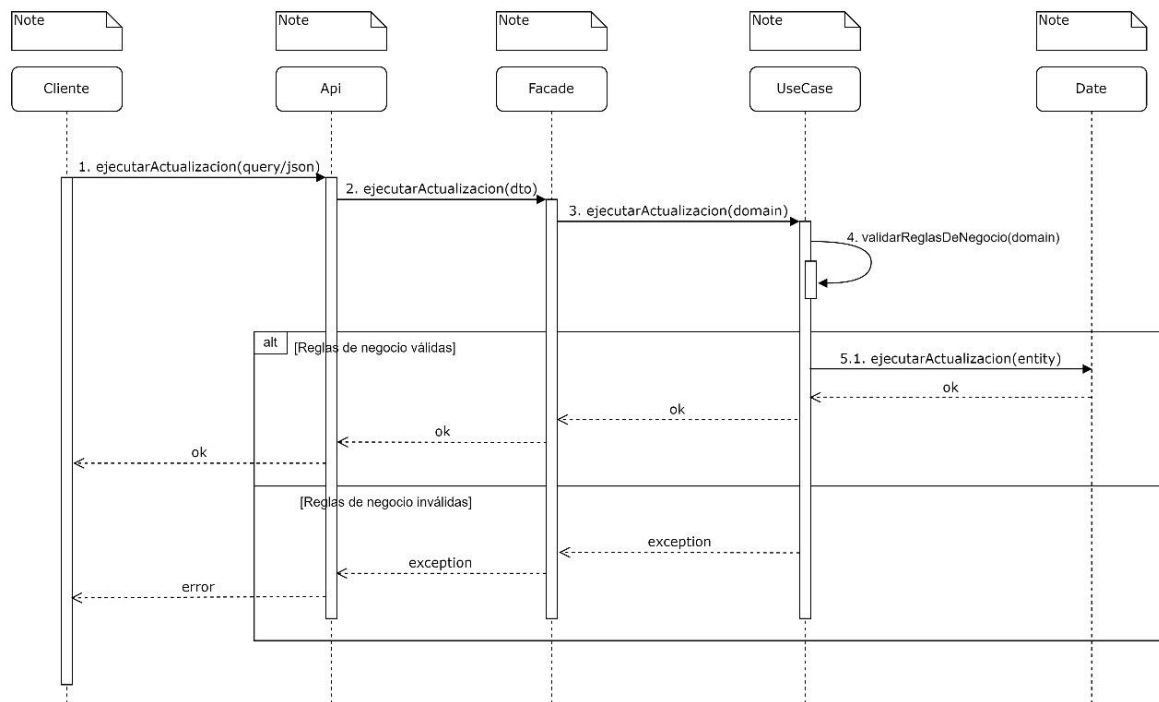
9.4.1 Diagrama de secuencia

Un diagrama de secuencia es una representación visual que muestra cómo interactúan distintos objetos o componentes en un sistema de software a lo largo del tiempo. En otras palabras, es como una película que presenta la secuencia de eventos y acciones que ocurren en el sistema en respuesta a ciertas acciones o eventos de los usuarios u otros componentes.

9.4.1.1 Componente general

Diagrama de secuencia general

9.4.1.1.1 Diagrama



9.4.1.1.2 Documentación

Diagrama	Secuencia		
Transacción	Secuencia de una transacción general		
Descripción	Modela una secuencia de una transacción general con el fin de poder mostrar cómo funciona la interacción entre las diferentes capas de la arquitectura definida		
Líneas de tiempo			
Nombre	Descripción		
Cliente	Es quien hace uso del aplicativo		
Api	Conjunto de reglas que permiten que diferentes aplicaciones se comuniquen entre sí		
Facade	Patrón de diseño que proporciona una interfaz simple y unificada para un conjunto más complejo de subsistemas o componentes		
UseCase	Describe cómo un sistema interactúa con sus usuarios o actores		
Data	Información que se almacena en el aplicativo		
Interacción			
Acción	Origen	Destino	Descripción
1. ejecutarActualización(graphql/json)	Cliente	Api	El cliente solicita a la api llevar a cabo una acción determinada, para lo cual le entrega los datos en formato
2. EjectutarActualización(dto)	Api	Facade	la api delega la ejección de la acción a la fachada correspondiente, entregandole la información requerida en un objeto de tipo DTO
3. EjecutarActualización(domain)	Facade	Usecase	La fachada delega la ejecución de la acción al caso e uso respectivo, iniciando la transacción y entregandole la información requerida en un objeto ded tipo dominio
4. ValidarReglasNegocio(domain)	Usecase	Usecase	Asegura el cumplimiento de toas las políticas o reglas de negocio, de tal manera que la transacción sea consistente
5.1 EjecutarActualización(entity)	Usecase	Data	Se ejecuta la actualización requerida contra la fuente de información a la cual se entrega en un objeto de tipo entidad. La acción solo se ejecuta cuando se ha garantizado el cumplimiento de todas las reglas de negocio. En caso contrario, se detiene el proceso a raíz de las excepciones

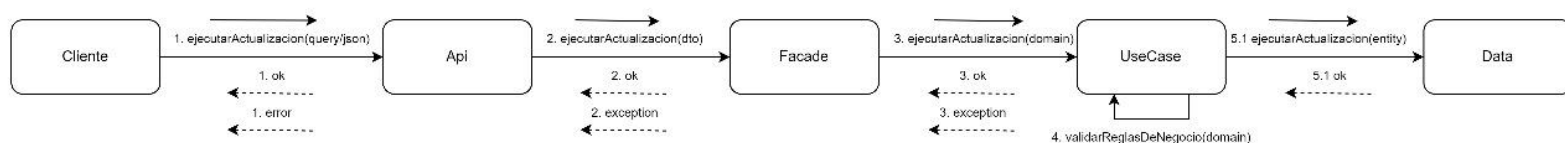
9.4.2 Diagrama de colaboración

Un diagrama de colaboración es una representación visual que muestra cómo los objetos o componentes de un sistema de software se comunican y colaboran entre sí para lograr un objetivo específico. En esencia, es una representación gráfica que destaca las interacciones y relaciones entre los elementos del sistema, lo que ayuda a comprender cómo trabajan juntos en un escenario particular.

9.4.2.1 Componente 1

Diagrama que muestra el 100% de los componentes que serán construidos o adoptados para proceder con la construcción del componente indicado. Adicionalmente indica como estos componentes están relacionados

9.4.2.1.1 Diagrama



9.4.2.1.2 Documentación

Referencia Documentación del diagrama de secuencias

9.5 Vista Física/Vista de Implantación

Una vista física es una representación del diseño arquitectónico de un sistema de software que se centra en los aspectos concretos de implementación, como la estructura física de hardware, la disposición de servidores, dispositivos, redes y otros elementos físicos. La motivación detrás de una vista física es proporcionar una visión detallada y concreta de cómo se despliega y ejecuta el sistema en el entorno físico.

9.5.1 Diagrama de despliegue

Un diagrama de despliegue es una representación visual que muestra cómo los componentes de un sistema de software se despliegan o implementan en la infraestructura de hardware y la disposición física de servidores, dispositivos y redes. En otras palabras, es un esquema gráfico que ilustra cómo los componentes del sistema se ubican y se conectan en el entorno de hardware real.

La motivación detrás de un diagrama de despliegue es proporcionar una vista clara y detallada de cómo el sistema se ejecutará en la infraestructura física. Esto es esencial para comprender cómo los componentes del software interactúan con los recursos de hardware y cómo se distribuye el sistema en un entorno de producción.

Referencia Arquitectura de solución y su Documentación