CentralAuth: Microservicio Centralizado de Autenticación y Gestión de Usuarios

Oscar Andres Vargas Zazzali

25 de junio de 2024

Versión del Documento: 1.0

Contents

1 Resumen Ejecutivo

El proyecto **CentralAuth** consiste en desarrollar un microservicio centralizado para la autenticación y gestión de usuarios. Este microservicio permitirá la centralización de los datos de usuarios, mejorando la seguridad y eficiencia en la gestión de identidades. Además, ofrecerá funcionalidades como el registro de usuarios, inicio de sesión, recuperación de contraseñas, generación de tokens JWT y verificación de correo electrónico. El objetivo es proporcionar una solución escalable y reutilizable que pueda integrarse fácilmente en múltiples aplicaciones y servicios.

1.1 Objetivos del Proyecto

- **Objetivo General:** Desarrollar un microservicio centralizado para la autenticación y gestión de usuarios, que ofrezca seguridad, escalabilidad y eficiencia en la administración de identidades.
- **Objetivos Específicos:**
  - Centralización de Datos: Unificar la gestión de usuarios y autenticación en un único sistema centralizado para facilitar el mantenimiento y la administración.
  - Seguridad: Implementar medidas de seguridad avanzadas, incluyendo cifrado de contraseñas, autenticación de dos factores (2FA) y protección contra ataques comunes como el phishing y el brute force.
  - Escalabilidad: Diseñar el sistema de manera que pueda escalar horizontalmente para manejar un gran número de usuarios y solicitudes simultáneas.
  - Interoperabilidad: Garantizar que el microservicio pueda integrarse fácilmente con diversas aplicaciones y plataformas mediante API RESTful.
  - Experiencia de Usuario: Asegurar una experiencia de usuario intuitiva y fluida tanto para los administradores como para los usuarios finales.

1.2 Importancia del Proyecto

El proyecto **CentralAuth** es crucial por varias razones:

- **Seguridad Mejorada:** Al centralizar la autenticación y gestión de usuarios, se pueden implementar medidas de seguridad robustas de manera consistente en todas las aplicaciones y servicios que utilicen el microservicio. Esto reduce el riesgo de brechas de seguridad y garantiza una protección uniforme de los datos de los usuarios.

- **Eficiencia Operativa:** Centralizar la administración de usuarios permite un manejo más eficiente de las identidades, roles y permisos. Los administradores pueden gestionar todos los usuarios desde un único punto, reduciendo la redundancia y el esfuerzo manual asociado con la gestión de múltiples sistemas de autenticación.
- **Escalabilidad y Flexibilidad:** Al ser un microservicio independiente, **CentralAuth** puede escalarse horizontalmente para manejar un aumento en la carga de trabajo, lo que es esencial para aplicaciones que experimentan un crecimiento rápido en el número de usuarios.
- **Reutilización y Consistencia:** Este microservicio puede ser reutilizado por diferentes aplicaciones dentro de una organización, asegurando que todas ellas sigan las mismas políticas de autenticación y gestión de usuarios. Esto promueve la consistencia y facilita el cumplimiento de normativas y estándares de seguridad.
- **Mejora de la Experiencia del Usuario:** Al ofrecer una autenticación rápida y segura, **CentralAuth** mejora la experiencia del usuario final, lo que puede aumentar la satisfacción y retención de usuarios en las aplicaciones que integren este servicio.

1.3 Resumen de los Puntos Principales del Documento

- **Visión del Proyecto:** Descripción inspiradora del objetivo a largo plazo del proyecto y cómo contribuirá al éxito de la organización.
- **Introducción:** Contexto del proyecto, objetivos generales y específicos, y alcance del proyecto.
- **Descripción del Proyecto:** Detalles técnicos del proyecto, funcionalidades clave, tecnologías a utilizar y arquitectura del sistema.
- **Planificación del Proyecto:** Cronograma detallado, fases del proyecto y entregables de cada fase.
- **Recursos y Roles:** Equipo del proyecto y sus responsabilidades, y recursos necesarios.
- **Análisis de Riesgos:** Identificación de riesgos potenciales y plan de mitigación y contingencia.
- **Requisitos del Sistema:** Requisitos funcionales, no funcionales y de hardware y software.
- **Diseño del Sistema:** Diagramas de flujo, de entidad-relación, de clases y arquitectura de la base de datos.
- **Plan de Desarrollo:** Metodología de desarrollo, estrategias de implementación, plan de integración y pruebas.

- **Plan de Pruebas:** Tipos de pruebas a realizar, casos de prueba y plan de gestión de defectos.
- **Plan de Implementación:** Estrategias de despliegue, plan de capacitación para usuarios finales y plan de soporte post-implementación.
- **Plan de Mantenimiento:** Estrategias de mantenimiento y actualización, y plan de gestión de cambios.
- **Conclusiones:** Resumen de los puntos clave y próximos pasos.
- **Anexos:** Documentación adicional y referencias.

2 Visión del Proyecto

3 Introducción

3.1 Contexto del Proyecto

En la actualidad, muchas organizaciones enfrentan desafíos relacionados con la gestión de usuarios y la autenticación debido a la existencia de múltiples sistemas independientes. Esta fragmentación genera problemas de seguridad, inconsistencias en la experiencia del usuario y un aumento en los costos operativos. Además, la integración de nuevas aplicaciones y servicios a menudo se complica por la falta de un sistema centralizado y eficiente para la gestión de identidades. "CentralAuth" surge como una solución a estos problemas, proporcionando un microservicio centralizado que simplifica y fortalece la autenticación y gestión de usuarios.

3.2 Objetivos del Proyecto

Objetivo General:

Desarrollar un microservicio centralizado para la autenticación y gestión de usuarios, que ofrezca seguridad, escalabilidad y eficiencia en la administración de identidades.

Objetivos Específicos:

- Centralización de Datos: Unificar la gestión de usuarios y autenticación en un único sistema centralizado para facilitar el mantenimiento y la administración.
- Seguridad: Implementar medidas de seguridad avanzadas, incluyendo cifrado de contraseñas y autenticación de dos factores (2FA).
- Escalabilidad: Diseñar el sistema de manera que pueda escalar horizontalmente para manejar un gran número de usuarios y solicitudes simultáneas.
- Interoperabilidad: Garantizar que el microservicio pueda integrarse fácilmente con diversas aplicaciones y plataformas mediante API RESTful.
- Experiencia de Usuario: Asegurar una experiencia de usuario intuitiva y fluida tanto para los administradores como para los usuarios finales.

3.3 Alcance del Proyecto

El alcance del proyecto "CentralAuth" abarca las siguientes actividades:

- Diseño del Sistema: Crear la arquitectura del sistema y definir los componentes clave.
- Desarrollo del Microservicio: Implementar las funcionalidades principales.
- Pruebas: Realizar pruebas unitarias, de integración y de aceptación para asegurar la calidad del microservicio.

- Documentación: Documentar la API y proporcionar guías de integración para desarrolladores.
- Despliegue: Implementar el microservicio en un entorno de producción y asegurar su disponibilidad.

3.4 Alcance del Producto

El alcance del producto "CentralAuth" incluye las siguientes funcionalidades principales:

- Registro de Usuarios: Permitir a los usuarios crear una cuenta con su información básica.
- Inicio de Sesión: Autenticar a los usuarios mediante credenciales (correo electrónico y contraseña).
- Recuperación de Contraseñas: Permitir a los usuarios restablecer su contraseña mediante un correo electrónico de recuperación.
- Generación de Tokens JWT: Emitir tokens JWT para la autenticación segura de los usuarios en las aplicaciones conectadas.
- Verificación de Correo Electrónico: Enviar correos electrónicos de verificación para confirmar la dirección de correo electrónico de los usuarios.

3.5 Estructura del Documento

4 Descripción del Proyecto

4.1 Detalles Técnicos del Proyecto

4.1.1 Estructura Técnica General

API Gateway

- **Funciones:**
  - Autenticación inicial de solicitudes HTTP.
  - Enrutamiento de solicitudes a los microservicios correspondientes.
  - Implementación de políticas de limitación de tasa para prevenir abusos.
- **Tecnologías:**
  - Laravel Passport o Laravel Sanctum para la gestión de tokens y autenticación OAuth.
  - NGINX como servidor web.
  - Redis para almacenamiento en caché de tokens y limitación de tasa.

Servicio de Autenticación

- **Funciones:**
  - Manejo de inicio de sesión y cierre de sesión.
  - Generación, validación y renovación de tokens JWT.
  - Emisión de eventos de desautenticación global.
  - Provisión de servicios de autenticación para otros sistemas, como monolitos o aplicaciones externas.
- **Tecnologías:**
  - Laravel Passport para OAuth y JWT.
  - Base de datos PostgreSQL para almacenamiento de usuarios y tokens.
  - Redis para almacenamiento en caché y gestión de colas.
  - Laravel Event Broadcasting para emisión de eventos.
  - WebSockets y Redis para transmisión de eventos en tiempo real.

Servicio de Gestión de Usuarios

- **Funciones:**
  - CRUD (Crear, Leer, Actualizar, Eliminar) de cuentas de usuario.
  - Validación y normalización de datos de usuario.
- **Tecnologías:**
  - Laravel Eloquent para ORM.
  - PostgreSQL como base de datos principal.
  - Laravel Valet para desarrollo local y pruebas.

Bus de Eventos

- **Funciones:**
  - Comunicación asincrónica entre microservicios.
  - Emisión y suscripción a eventos relevantes del sistema.
  - Manejo de eventos de desautenticación global para invalidar sesiones en todas las plataformas.
- **Tecnologías:**
  - Laravel Event Broadcasting con Redis y WebSockets para la transmisión de eventos.
  - Supervisor para la gestión de colas y trabajos en segundo plano.

Base de Datos Central

- **Funciones:**
  - Almacenamiento de datos relacionados con usuarios y autenticación.
  - Garantía de integridad y consistencia de datos.
- **Tecnologías:**
  - PostgreSQL como base de datos principal.
  - Configuración de replicación y clustering para alta disponibilidad.

4.1.2 Interacción entre Componentes

- **Flujo de Solicitudes HTTP:**
  - El **API Gateway** (NGINX) recibe las solicitudes y las autentica con el **Servicio de Autenticación** (usando Laravel Passport). Luego, enruta las solicitudes autenticadas al microservicio adecuado.
  - **Servicio de Autenticación**: Verifica credenciales y genera tokens JWT. Utiliza Laravel Passport para la gestión de tokens.
  - **Servicio de Gestión de Usuarios**: Procesa operaciones CRUD interactuando directamente con la **Base de Datos Central**.
- **Control de Acceso:**
  - **Bus de Eventos**: Facilita la emisión de eventos cuando se crean, actualizan o eliminan usuarios, permitiendo que otras aplicaciones se actualicen en tiempo real mediante Laravel Event Broadcasting.
- **Comunicación Asincrónica:**
  - **Bus de Eventos**: Utiliza Redis y WebSockets para permitir que las aplicaciones se suscriban a eventos relevantes y manejar datos de usuarios eficientemente. Los eventos incluyen, pero no se limitan a, la creación, modificación y eliminación de usuarios.

4.1.3 Integración con Otros Sistemas

- El Servicio de Autenticación no solo se utiliza para los microservicios dentro del sistema, sino que también proporciona servicios de autenticación y autorización para sistemas externos, como un monolito o aplicaciones de terceros. Esto se logra mediante la provisión de endpoints de autenticación que permiten a estos sistemas:
  - Validar tokens JWT emitidos por el Servicio de Autenticación.
  - Consultar roles y permisos de usuarios para implementar control de acceso interno.
  - Integrarse fácilmente mediante API RESTful, asegurando que cualquier sistema pueda verificar la autenticidad de los usuarios y sus permisos antes de permitir el acceso a recursos internos.

4.1.4 Ampliación de la Interacción entre Componentes

- Para asegurar una comunicación efectiva y la actualización en tiempo real de los datos, se han implementado varios patrones de diseño y tecnologías:
  - **Patrón de Saga**: Para gestionar transacciones distribuidas entre los servicios, asegurando la consistencia eventual.

  - **CQRS (Command Query Responsibility Segregation)**: Para separar las operaciones de lectura y escritura, optimizando el rendimiento.
  - **Event Sourcing**: Para registrar todos los cambios de estado como una secuencia de eventos, facilitando el monitoreo y la auditoría.

4.1.5 Escalabilidad y Disponibilidad

- Cada componente del sistema está diseñado para ser escalable de manera independiente:
  - **API Gateway** y **Servicios de Autenticación**: Pueden escalar horizontalmente mediante balanceadores de carga.
  - **Base de Datos Central**: Configurada con replicación para alta disponibilidad y recuperación ante fallos.

4.1.6 Seguridad

- Se han implementado varias capas de seguridad para proteger los datos y las comunicaciones:
  - **Cifrado de datos en tránsito y en reposo**.
  - **Autenticación multifactor (MFA)** para acceso de usuarios sensibles.
  - **Auditorías y registros de actividad** para monitorear accesos y cambios.

4.2 Funcionalidades Clave

- **Registro de Usuarios:** Permitir a las aplicaciones cliente registrar nuevos usuarios en el sistema.
  - Los clientes pueden enviar la información del usuario, como nombre, correo electrónico y contraseña, y recibir una confirmación del registro.
- **Inicio de Sesión:** Permitir a las aplicaciones cliente autenticar a los usuarios existentes.
  - Las aplicaciones pueden enviar las credenciales del usuario y recibir una confirmación de autenticación si las credenciales son correctas.
- **Recuperación de Contraseñas:** Proveer una función para que las aplicaciones cliente inicien el proceso de recuperación de contraseña.
  - Las aplicaciones pueden solicitar el restablecimiento de la contraseña y enviar un enlace de recuperación al correo electrónico del usuario.

- **Generación de Tokens JWT:** Proveer una función para emitir y renovar tokens JWT para la autenticación segura de los usuarios.
  - Las aplicaciones cliente pueden solicitar la emisión de tokens al iniciar sesión y la renovación de tokens cuando sea necesario.
- **Verificación de Correo Electrónico:** Proveer una función para que las aplicaciones cliente envíen correos electrónicos de verificación a los usuarios.
  - Los clientes pueden iniciar el envío de correos de verificación y confirmar la verificación de la dirección de correo electrónico del usuario.
- **Autenticación Multifactor (2FA):** Proveer una función para que las aplicaciones cliente activen y gestionen la autenticación de dos factores para los usuarios.
  - Las aplicaciones pueden habilitar el 2FA, enviar códigos de verificación y validar estos códigos para el acceso seguro.
- **Gestión de Sesiones:** Proveer una función para que las aplicaciones cliente monitoricen y gestionen las sesiones de usuario.
  - Los clientes pueden consultar sesiones activas, cerrar sesiones y gestionar la duración de las sesiones.
- **Interoperabilidad con Otras Aplicaciones:** Asegurar que el microservicio puede integrarse fácilmente con diversas aplicaciones.
  - Ofrecer endpoints estándar para que las aplicaciones puedan autenticar usuarios y consultar roles y permisos específicos de cada aplicación.
- **Desautenticación Global:** Permitir a las aplicaciones cliente cerrar la sesión de un usuario en todas las plataformas de manera coordinada.
  - Emitir eventos de desautenticación global que invaliden sesiones en todas las aplicaciones suscritas.

4.3 Tecnologías a Utilizar

4.3.1 Lenguajes de Programación

- **PHP:** Utilizado para desarrollar el backend del microservicio de autenticación.
- **JavaScript:** Utilizado para el desarrollo del frontend.

4.3.2 Frameworks

- **Laravel:** Framework de PHP utilizado para el desarrollo del backend.
- **Vue.js:** Framework de JavaScript utilizado para el desarrollo del frontend.

4.3.3 Bases de Datos

- **PostgreSQL:** Base de datos relacional utilizada para almacenar los datos de los usuarios y tokens de autenticación.

4.3.4 Servidores y Hosting

- **NGINX:** Utilizado como servidor web y balanceador de carga.
- **AWS:** Servicios de Amazon Web Services para hosting y escalabilidad.

4.3.5 Seguridad

- **Cifrado:** TLS/SSL para cifrado de datos en tránsito.
- **Autenticación Multifactor (2FA):** Implementación de 2FA para seguridad adicional.
- **OAuth y JWT:** Para la gestión de tokens y autenticación segura.

4.3.6 Herramientas de Desarrollo y CI/CD

- **GitHub:** Repositorio de código y gestión de versiones.
- **Jenkins:** Herramienta de integración continua y despliegue continuo.

4.3.7 Servicios de Mensajería

- **Redis:** Utilizado para la gestión de colas y almacenamiento en caché.
- **RabbitMQ:** Utilizado para la comunicación asincrónica entre microservicios.

4.3.8 Monitoreo y Logging

- **ELK Stack:** ElasticSearch, Logstash y Kibana para monitoreo y análisis de logs.
- **Prometheus y Grafana:** Para monitoreo del rendimiento y visualización de métricas.

5 Planificación del Proyecto

5.1 Cronograma Detallado (Timeline)

- **Fase de Diseño y Planificación**
  - **Inicio**: 25 de junio de 2024
  - **Duración**: 2 semanas
  - **Fin**: 9 de julio de 2024
  - **Actividades Principales**:
    - Definición de requisitos (25 de junio - 27 de junio)
    - Diseño de la arquitectura del sistema (28 de junio - 3 de julio)
    - Diseño de la base de datos y planificación de microservicios (4 de julio - 9 de julio)
    - Planificación del cronograma detallado y asignación de recursos y roles (4 de julio - 9 de julio)
  - **Entregables**:
    - Documentación de requisitos
    - Diagramas de arquitectura del sistema
    - Diseño de la base de datos
- **Fase de Desarrollo (Backend)**
  - **Iteración 1: Registro de Usuarios (Backend)**
    - **Inicio**: 10 de julio de 2024
    - **Duración**: 1 semana
    - **Fin**: 16 de julio de 2024
    - **Actividades Principales**:
      - Desarrollo del backend para el registro de usuarios
      - Pruebas unitarias y de integración para el registro de usuarios
    - **Entregables**:
      - API funcional para el registro de usuarios
      - Pruebas unitarias para el registro de usuarios
  - **Iteración 2: Gestión de Usuarios (Backend)**
    - **Inicio**: 17 de julio de 2024
    - **Duración**: 1 semana
    - **Fin**: 23 de julio de 2024
    - **Actividades Principales**:
      - Desarrollo del backend para la gestión de usuarios (CRUD)
      - Pruebas unitarias y de integración para la gestión de usuarios
    - **Entregables**:

        - API funcional para la gestión de usuarios
        - Pruebas unitarias para la gestión de usuarios
- **Iteración 3: Inicio de Sesión (Backend)**
    - **Inicio**: 24 de julio de 2024
    - **Duración**: 1 semana
    - **Fin**: 30 de julio de 2024
    - **Actividades Principales**:
        - Desarrollo del backend para el inicio de sesión
        - Pruebas unitarias y de integración para el inicio de sesión
    - **Entregables**:
        - API funcional para el inicio de sesión
        - Pruebas unitarias para el inicio de sesión
- **Iteración 4: Recuperación de Contraseñas (Backend)**
    - **Inicio**: 31 de julio de 2024
    - **Duración**: 1 semana
    - **Fin**: 6 de agosto de 2024
    - **Actividades Principales**:
        - Desarrollo del backend para la recuperación de contraseñas
        - Pruebas unitarias y de integración para la recuperación de contraseñas
    - **Entregables**:
        - API funcional para la recuperación de contraseñas
        - Pruebas unitarias para la recuperación de contraseñas
- **Iteración 5: Generación de Tokens JWT (Backend)**
    - **Inicio**: 7 de agosto de 2024
    - **Duración**: 1 semana
    - **Fin**: 13 de agosto de 2024
    - **Actividades Principales**:
        - Desarrollo del backend para la generación de tokens JWT
        - Pruebas unitarias y de integración para la generación de tokens JWT
    - **Entregables**:
        - API funcional para la generación de tokens JWT
        - Pruebas unitarias para la generación de tokens JWT
- **Iteración 6: Verificación de Correo Electrónico (Backend)**
    - **Inicio**: 14 de agosto de 2024
    - **Duración**: 1 semana
    - **Fin**: 20 de agosto de 2024
    - **Actividades Principales**:

        - Desarrollo del backend para la verificación de correo electrónico
        - Pruebas unitarias y de integración para la verificación de correo electrónico
      - **Entregables**:
        - API funcional para la verificación de correo electrónico
        - Pruebas unitarias para la verificación de correo electrónico

- **Fase de Desarrollo (Frontend)**
  - **Iteración 7: Registro de Usuarios (Frontend)**
    - **Inicio**: 21 de agosto de 2024
    - **Duración**: 1 semana
    - **Fin**: 27 de agosto de 2024
    - **Actividades Principales**:
      - Desarrollo del frontend para el registro de usuarios
      - Pruebas unitarias y de integración para el frontend de registro de usuarios
    - **Entregables**:
      - Interfaz de usuario funcional para el registro de usuarios
      - Pruebas unitarias para el frontend
  - **Iteración 8: Inicio de Sesión (Frontend)**
    - **Inicio**: 28 de agosto de 2024
    - **Duración**: 1 semana
    - **Fin**: 3 de septiembre de 2024
    - **Actividades Principales**:
      - Desarrollo del frontend para el inicio de sesión
      - Pruebas unitarias y de integración para el frontend de inicio de sesión
    - **Entregables**:
      - Interfaz de usuario funcional para el inicio de sesión
      - Pruebas unitarias para el frontend
  - **Iteración 9: Recuperación de Contraseñas (Frontend)**
    - **Inicio**: 4 de septiembre de 2024
    - **Duración**: 1 semana
    - **Fin**: 10 de septiembre de 2024
    - **Actividades Principales**:
      - Desarrollo del frontend para la recuperación de contraseñas
      - Pruebas unitarias y de integración para el frontend de recuperación de contraseñas
    - **Entregables**:

        - Interfaz de usuario funcional para la recuperación de contraseñas
        - Pruebas unitarias para el frontend

- **Fase de Despliegue**
  - **Inicio**: 11 de septiembre de 2024
  - **Duración**: 1 semana
  - **Fin**: 17 de septiembre de 2024
  - **Actividades Principales**:
    - Despliegue en el entorno de producción (11 de septiembre - 13 de septiembre)
    - Verificación de despliegue (14 de septiembre - 17 de septiembre)
  - **Entregables**:
    - Microservicio desplegado en producción
    - Verificación de despliegue exitosa

6 Recursos y Roles

6.1 Equipo del Proyecto y Responsabilidades

- **Líder del Proyecto**
  - **Responsabilidades**:
    - Supervisar el progreso del proyecto.
    - Coordinar las actividades del equipo.
    - Asegurar que los objetivos del proyecto se cumplan en tiempo y forma.
    - Comunicarse con las partes interesadas.
- **Desarrollador Backend**
  - **Responsabilidades**:
    - Desarrollar las APIs del microservicio de autenticación.
    - Implementar la lógica de negocio para el registro, login y recuperación de contraseñas.
    - Escribir pruebas unitarias y de integración.
    - Garantizar la seguridad y eficiencia del backend.
- **Desarrollador Frontend**
  - **Responsabilidades**:
    - Desarrollar la interfaz de usuario para el registro, login y recuperación de contraseñas.
    - Integrar el frontend con las APIs del backend.
    - Escribir pruebas unitarias y de integración para el frontend.
    - Asegurar una experiencia de usuario fluida y eficiente.
- **Ingeniero DevOps**
  - **Responsabilidades**:
    - Configurar y mantener el entorno de desarrollo y producción.
    - Implementar pipelines de CI/CD.
    - Monitorear el desempeño del sistema y gestionar la infraestructura.
    - Asegurar la disponibilidad y escalabilidad del sistema.
- **Tester/QA**
  - **Responsabilidades**:
    - Realizar pruebas funcionales y no funcionales.
    - Ejecutar pruebas de aceptación y reportar defectos.
    - Validar que las funcionalidades cumplan con los requisitos especificados.
    - Colaborar con los desarrolladores para resolver problemas encontrados durante las pruebas.

6.2 Recursos Necesarios

- **Herramientas de Desarrollo**
  - **IDE**: Visual Studio Code, PHPStorm.
  - **Lenguajes de Programación**: PHP para backend, JavaScript para frontend.
  - **Frameworks**: Laravel para backend, Vue.js para frontend.
  - **Repositorios**: GitHub para control de versiones.
- **Servicios y Hosting**
  - **Servidor Web**: NGINX para balanceo de carga y servidor web.
  - **Base de Datos**: PostgreSQL para almacenamiento de datos.
  - **Cloud Hosting**: AWS para infraestructura escalable.
- **Seguridad**
  - **Cifrado**: Certificados TLS/SSL para seguridad en tránsito.
  - **Autenticación**: OAuth y JWT para gestión de tokens, implementación de 2FA.
- **Herramientas de CI/CD**
  - **Integración Continua**: Jenkins para automatización de builds y despliegues.
  - **Despliegue Continuo**: Pipelines de CI/CD para despliegue automático en entornos de desarrollo y producción.
- **Monitoreo y Logging**
  - **Monitoreo de Rendimiento**: Prometheus y Grafana para métricas y visualización.
  - **Logging**: ELK Stack (ElasticSearch, Logstash, Kibana) para análisis y monitoreo de logs.

7 Análisis de Riesgos

7.1 Escala de Gravedad del Riesgo

7.1.1 Escala de Probabilidad

- **Baja (1)**: El riesgo es poco probable que ocurra.
- **Media (2)**: El riesgo tiene una posibilidad moderada de ocurrir.
- **Alta (3)**: El riesgo es muy probable que ocurra.

7.1.2 Escala de Impacto

- **Bajo (1)**: El impacto del riesgo en el proyecto es mínimo y fácil de manejar.
- **Moderado (2)**: El impacto del riesgo puede causar problemas significativos pero manejables.
- **Alto (3)**: El impacto del riesgo puede ser devastador para el proyecto, causando grandes retrasos o fallos críticos.

Matriz de riesgo

7.2 Identificación y Evaluación de Riesgos Potenciales

7.2.1 Riesgos del Proyecto

- **Cambio de Requisitos: Seguridad**
  - **Descripción**: Los requisitos de seguridad del proyecto pueden cambiar debido a nuevas necesidades o descubrimientos de vulnerabilidades.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)
- **Cambio de Requisitos: Escalabilidad**
  - **Descripción**: Los requisitos de escalabilidad del proyecto pueden cambiar debido a un aumento inesperado en la cantidad de usuarios.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)
- **Cambio de Requisitos: Integración**
  - **Descripción**: Los requisitos de integración del proyecto pueden cambiar debido a la necesidad de soportar nuevas aplicaciones cliente.
  - **Probabilidad**: 2 (Media)

  - **Impacto**: 2 (Moderado)

- **Falta de Comunicación con Partes Interesadas**
  - **Descripción**: Problemas de comunicación pueden surgir con las partes interesadas, causando malentendidos y retrasos.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 2 (Moderado)

- **Desgaste del Equipo por Alta Carga de Trabajo**
  - **Descripción**: La alta carga de trabajo puede afectar la moral y la productividad de la persona a cargo.
  - **Probabilidad**: 3 (Alta)
  - **Impacto**: 3 (Alto)

- **Desgaste del Equipo por Plazos Ajustados**
  - **Descripción**: La presión para cumplir con plazos ajustados puede afectar la moral y la productividad de la persona a cargo.
  - **Probabilidad**: 3 (Alta)
  - **Impacto**: 2 (Moderado)

- **Falta de Redundancia: Indisponibilidad del Personal**
  - **Descripción**: La falta de personal adicional significa que si la persona a cargo está indisponible, el proyecto puede verse seriamente afectado.
  - **Probabilidad**: 3 (Alta)
  - **Impacto**: 3 (Alto)

7.2.2 Riesgos Técnicos

- **Problemas de Calidad: Pruebas Inadecuadas**
  - **Descripción**: La falta de pruebas adecuadas puede resultar en la entrega de un microservicio con defectos.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)

- **Problemas de Calidad: Defectos en Autenticación**
  - **Descripción**: Defectos en el manejo de la autenticación pueden comprometer la seguridad del sistema.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)

- **Problemas de Calidad: Defectos en Gestión de Usuarios**
  - **Descripción**: Defectos en la gestión de usuarios pueden llevar a la pérdida o corrupción de datos.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 2 (Moderado)
- **Problemas de Calidad: Defectos en API**
  - **Descripción**: Defectos en las APIs pueden resultar en fallos de integración con las aplicaciones cliente.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 2 (Moderado)
- **Problemas de Calidad: Defectos en Seguridad**
  - **Descripción**: Defectos en la implementación de medidas de seguridad pueden dejar el sistema vulnerable a ataques.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)
- **Problemas de Compatibilidad de Software**
  - **Descripción**: Problemas de compatibilidad entre diferentes versiones de software y bibliotecas utilizadas pueden causar fallos en el sistema.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 2 (Moderado)
- **Complejidad del Código**
  - **Descripción**: La programación orientada a eventos puede llevar a un código más complejo y difícil de mantener debido a la naturaleza asíncrona y las múltiples fuentes de eventos.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 2 (Moderado)
- **Problemas de Debugging**
  - **Descripción**: La depuración de aplicaciones orientadas a eventos puede ser más difícil debido a la falta de una secuencia clara de ejecución.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 2 (Moderado)
- **Condiciones de Carrera**

- **Descripción**: La naturaleza asíncrona de la EDP puede introducir condiciones de carrera, donde dos o más eventos intentan acceder o modificar el mismo recurso al mismo tiempo.
- **Probabilidad**: 2 (Media)
- **Impacto**: 3 (Alto)

- **Bloqueos y Errores Difíciles de Reproducir**
  - **Descripción**: Los errores en aplicaciones orientadas a eventos pueden ser difíciles de reproducir y diagnosticar, especialmente cuando dependen de la interacción de múltiples eventos.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)

7.2.3 Riesgos de Seguridad

- **Exposición de Datos Sensibles**
  - **Descripción**: Fallos en la seguridad pueden resultar en la exposición de datos sensibles de los usuarios.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)

- **Ataques de Fuerza Bruta**
  - **Descripción**: El sistema puede ser vulnerable a ataques de fuerza bruta en las contraseñas.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)

- **Phishing**
  - **Descripción**: Usuarios pueden ser víctimas de ataques de phishing, comprometiendo sus credenciales.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)

- **Vulnerabilidades en Bibliotecas Externas**
  - **Descripción**: Las bibliotecas externas pueden tener vulnerabilidades que pueden ser explotadas por atacantes.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)

7.2.4 Riesgos de Escalabilidad y Rendimiento

- **Degradación del Rendimiento Bajo Alta Carga**
  - **Descripción**: El sistema puede no manejar adecuadamente un aumento repentino en la carga de usuarios y solicitudes.
  - **Probabilidad**: 3 (Alta)
  - **Impacto**: 3 (Alto)
- **Limitaciones de Escalabilidad Horizontal**
  - **Descripción**: El diseño del sistema puede no soportar escalabilidad horizontal efectiva.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)
- **Problemas de Tiempo de Respuesta**
  - **Descripción**: Aumento en el tiempo de respuesta bajo alta carga de usuarios puede afectar la experiencia del usuario.
  - **Probabilidad**: 3 (Alta)
  - **Impacto**: 3 (Alto)
- **Sobrecarga de Eventos**
  - **Descripción**: Un exceso de eventos puede sobrecargar el sistema, provocando una disminución del rendimiento o incluso fallos en el sistema.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 2 (Moderado)
- **Latencia en el Procesamiento de Eventos**
  - **Descripción**: La latencia en el procesamiento de eventos puede afectar la experiencia del usuario, especialmente en sistemas que requieren respuestas en tiempo real.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 2 (Moderado)

7.2.5 Riesgos de Integración

- **Incompatibilidad con APIs Existentes**
  - **Descripción**: Incompatibilidades con las APIs existentes pueden causar problemas en la integración con aplicaciones cliente.
  - **Probabilidad**: 2 (Media)

- **Impacto**: 2 (Moderado)

- **Errores en la Comunicación entre Servicios**
  - **Descripción**: Errores en la comunicación entre microservicios pueden llevar a fallos en el sistema.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)

- **Compatibilidad de Eventos entre Sistemas**
  - **Descripción**: Problemas de compatibilidad entre los eventos generados por diferentes sistemas pueden causar errores en la integración.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 2 (Moderado)

- **Manejo de Eventos No Previstos**
  - **Descripción**: La aparición de eventos no previstos puede llevar a comportamientos inesperados o errores en el sistema.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 2 (Moderado)

7.2.6 Riesgos de Disponibilidad

- **Interrupciones en Servicios Externos**
  - **Descripción**: Caídas o interrupciones en los servicios externos (por ejemplo, AWS, Redis) pueden afectar la disponibilidad del sistema.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)

- **Limitaciones de Capacidad en Servicios Externos**
  - **Descripción**: Limitaciones en los servicios de nube utilizados pueden afectar la disponibilidad y escalabilidad del sistema.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)

- **Dependencia de Herramientas de CI/CD**
  - **Descripción**: Problemas con herramientas de CI/CD pueden retrasar el despliegue y la integración continua.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 2 (Moderado)

- **Dependencia de Laravel**
  - **Descripción**: Problemas o actualizaciones incompatibles en Laravel pueden afectar el desarrollo y mantenimiento del backend.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)
- **Dependencia de Vue.js**
  - **Descripción**: Problemas o actualizaciones incompatibles en Vue.js pueden afectar el desarrollo y mantenimiento del frontend.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)
- **Interrupciones de Servicio por Mantenimiento**
  - **Descripción**: Mantenimiento no planificado puede causar interrupciones en el servicio.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 2 (Moderado)
- **Caídas del Servidor**
  - **Descripción**: Problemas de infraestructura pueden resultar en caídas del servidor y pérdida de disponibilidad del servicio.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)
- **Limitaciones de Capacidad en Servicios de Nube**
  - **Descripción**: Limitaciones en los servicios de nube utilizados pueden afectar la disponibilidad y escalabilidad del sistema.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)

7.2.7 Riesgos Legales y Regulatorios

- **Incumplimiento de Normativas de Privacidad de Datos**
  - **Descripción**: Fallos en el cumplimiento de las normativas de privacidad de datos pueden resultar en sanciones.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)
- **Problemas de Propiedad Intelectual**

  - **Descripción**: Uso no autorizado de bibliotecas o herramientas de software puede llevar a problemas de propiedad intelectual.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)

- **Incumplimiento de Licencias de Software**
  - **Descripción**: Uso incorrecto o no autorizado de bibliotecas o herramientas de software puede llevar a problemas legales.
  - **Probabilidad**: 2 (Media)
  - **Impacto**: 3 (Alto)