

UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA



VentureSync

Curso:

Desarrollo Basado en Plataforma CS 2031

Profesor:

Bellido Angulo, Jesus Edwin

Noel Rabianes, Mateo

Integrantes:

Ayra Francisco, Gussephe Benjamin (202210033) 100%
Olivares Sarmiento, Javier Ignacio Leon (202310689) 60%
Salazar Medina, Breysi Fernanda (202310648) 100%
Luis Martin Barzola(no se sabe) 0%

Repositorio:

<https://github.com/CS2031-DBP/proyecto-backend-grupo-2.git>

Lima, 31 de mayo de 2024

Índice

Índice	1
1. Resumen ejecutivo	2
2. Introducción.....	3
3. Identificación del Problema o Necesidad	3
4. Descripción de la Solución	4
5. Proceso de Desarrollo.....	6
6. Modelo Entidad-Relación.....	7
7. Testing	9
8. Medidas de Seguridad Implementadas	10
Conclusión.....	11
Apéndices	11

1. Resumen ejecutivo

1.1 Propósito del Proyecto

El presente proyecto busca abordar una necesidad crítica en el mercado peruano, donde numerosos emprendedores con ideas prometedoras enfrentan desafíos significativos para convertir estas ideas en realidades tangibles y rentables. La principal barrera que este proyecto busca superar es la desconexión entre los emprendedores y los recursos esenciales como financiamiento y redes de contacto profesional que son fundamentales para la materialización de proyectos innovadores.

1.2 Solución Propuesta

El proyecto VentureSync propone una solución integral para abordar la desconexión entre emprendedores y los recursos esenciales en Perú mediante una plataforma digital que facilita la conexión directa entre creadores de proyectos y una red de recursos financieros y profesionales. Esta plataforma ofrece funcionalidades clave como la publicación de proyectos, la postulación a los mismos y la gestión de financiación, lo que permite una interacción eficiente y efectiva entre todos los actores involucrados.

La utilización de plataformas digitales para conectar a emprendedores con recursos y mentores es reconocida por su capacidad para superar barreras geográficas y temporales, proporcionando acceso a una red más amplia de oportunidades y conocimientos. Al integrar herramientas de colaboración y redes de apoyo dentro de una única plataforma, VentureSync se posiciona como un facilitador crucial en el ecosistema de emprendimiento, permitiendo que más proyectos pasen de la idea a la implementación.

1.3 Beneficios Clave

Desarrollo económico y social: La plataforma fomenta el movimiento e incremento económico como nación, otorgando la oportunidad de apoyar monetariamente el despegue de proyectos, desarrollados colaborativamente.

Acceso a red de contactos profesionales: Facilita la innovación colaborativa con profesionales capacitados, lo cual promueve el apoyo social e interacción entre profesionales del mismo rubro, permitiendo el surgimiento de otros nuevos proyectos.

Impulso de solución a desafíos: VentureSync actúa como un mediador para la solución de problemas locales mediante la implementación de soluciones creativas y efectivas que surgen de la colaboración entre distintos actores del ecosistema de creación. Al conectar proyectos con recursos y expertos, la plataforma asegura un impacto directo y tangible en la comunidad, impulsando la resolución de desafíos mediante enfoques revolucionarios. Fomentando una sociedad más emprendedora y resiliente.

2. Introducción

2.1. Contexto

En Perú, el contexto en el que surge la necesidad abordada por VentureSync es particularmente complejo. Existen diversos factores, por ejemplo, la financiación y apoyo a las pequeñas y medianas empresas (PYMEs) y los emprendedores han enfrentado presiones significativas, con un aumento en el costo de la financiación y una notable disminución en los préstamos a las PYMEs desde 2020. Esto ha puesto en manifiesto la necesidad crítica de diversificar los instrumentos financieros para apoyar adecuadamente a los emprendedores. Estos desafíos subrayan la relevancia de VentureSync como una solución integral para conectar a emprendedores con los recursos necesarios para fomentar el emprendimiento.

2.2. Objetivos del Proyecto

Facilitar el acceso a financiación: Aumentar el acceso a financiamiento para innovadores, integrando contacto para contribuciones económicas de proyectos, permitiendo su despegue.

Fomentar la colaboración: Establecer conexiones efectivas entre innovadores y profesionales, creando una red de apoyo para cierto rubro, permitiendo la orientación, compartiendo conocimiento y fomentando la transferencia de habilidades, abordando la necesidad de apoyo.

Impulsar mentes jóvenes: Promover el desarrollo de soluciones innovadoras y sostenibles a problemas sociales desde jóvenes es necesario. Incentivar el desarrollo colaborativo y no solo planteamiento de proyectos es una alternativa de prevención a futuras problemáticas en la sociedad.

VentureSync ofrece un espacio donde la innovación puede encontrarse con la oportunidad, no solo pretende ser una plataforma, sino también crear un ecosistema en el cual los proyectos puedan nacer, desarrollarse y alcanzar su máximo potencial, impactando positivamente en la sociedad peruana.

3. Identificación del Problema o Necesidad

3.1. Descripción del Problema

Existe una significativa brecha entre las ideas innovadoras y su ejecución. Esto se debe principalmente a la falta de acceso a recursos esenciales como financiamiento, apoyo de profesionales o contactos. Muchos emprendedores con proyectos potencialmente transformadores no encuentran el apoyo necesario para desarrollar sus conceptos a la realidad, lo que resulta en una pérdida de oportunidades tanto a nivel individual como nacional. La relevancia de solucionar este problema radica en el potencial sin explotar de estas ideas innovadoras, que podrían contribuir sustancialmente al desarrollo económico y social de toda la nación. Al proporcionar los medios para que estos proyectos se materialicen, no solo se fomenta la creación de empleo y el pensamiento grupal, sino que también se promueve la resolución de

problemas locales mediante soluciones creativas y efectivas. Además, el estímulo al emprendimiento puede generar un ciclo de progreso, beneficiando a la sociedad.

3.2. Justificación

La relevancia de solucionar los problemas de acceso a financiación y apoyo para innovadores en Perú es grande. Al proporcionar recursos y conexiones necesarias, VentureSync apunta a reconocer a los emprendedores por su papel vital en la creación de empleo al participar en revolucionarias iniciativas, fundamentales para el desarrollo económico sostenible. Además, el llevar a cabo la creación de proyectos colaborativos previene y soluciona necesidades futuras. Por lo tanto, VentureSync no solo es relevante, sino esencial para fomentar un ecosistema de emprendimiento que pueda liderar la recuperación y el futuro crecimiento económico en Perú, haciendo un uso óptimo de sus recursos locales.

4. Descripción de la Solución

4.1. Funcionalidades Implementadas

Creación de proyectos: Mediante el uso de VentureSync, los usuarios van a contar con la posibilidad de llevar a cabo proyectos, emprendimientos e iniciativas que en un pasado no se llegaban a concretar por falta de conocimiento, especialmente en los rubros de adquirir fuerza laboral, interesada no únicamente por una ganancia económica, sino por compartir ideales solucionados mediante determinada propuesta.

Participación de múltiples proyectos: Gracias a las facilidades que brindamos con la creación de proyectos, nuestra intención es que sea igual de sencillo participar de estos. Poder participar de múltiples iniciativas en simultaneo, permite que nuestros usuarios puedan cubrir todas las áreas de su interés.

Facilidad de contacto entre Usuarios: Mediante cada proyecto, los usuarios están habilitados a contactarse con los creadores de los mismos mediante correo electrónico y a través de nuestra plataforma.

4.2. Tecnologías Utilizadas

Lenguaje de programación Java: Usamos el lenguaje de programación Java, especialmente el uso de la **programación orientada a objetos (OOP)**, que resulta conveniente para evitar iterar numerosas veces en elementos ya creados, con el fin de mantener optimizado el back-end de nuestro programa.

Spring Framework: Spring Framework es un marco de trabajo completo para el desarrollo de aplicaciones Java que proporciona una infraestructura de soporte robusta. Esta plataforma facilita la integración de diversas tecnologías como manejo de transacciones, acceso a datos, y otros servicios esenciales, haciendo que el desarrollo sea más eficiente y menos propenso a errores.

Spring Boot: Spring Boot es un proyecto dentro del ecosistema de Spring que simplifica enormemente el desarrollo de aplicaciones utilizando Spring Framework. Proporciona configuraciones por defecto óptimas y autoconfiguración que ayudan a reducir el tiempo de desarrollo. Además, permite la creación de aplicaciones

independientes con mínima configuración inicial, lo que facilita la ejecución y despliegue rápidos.

Inyección de dependencias: La inyección de dependencias, proporcionada por Spring Framework, es fundamental para crear software con componentes flexibles y desacoplados. Esta técnica permite a los desarrolladores mantener un bajo acoplamiento entre los componentes del software, facilitando la gestión de dependencias y la mejora de la mantenibilidad y escalabilidad de las aplicaciones.

Spring MVC: Utilizamos también Spring MVC, un Modelo-Vista-Controlador Sub-Framework de Spring que nos ayuda a construir aplicaciones web de forma eficiente. Este componente estructura la aplicación en capas claramente definidas, facilitando la separación de la lógica de negocio del código de la interfaz de usuario. Esto resulta en aplicaciones web más limpias y fáciles de mantener que responden bien a los cambios y a la integración de nuevas funcionalidades.

Spring Data JPA: Spring Data JPA simplifica la implementación de repositorios de datos para aplicaciones basadas en JPA (Java Persistence API), ofreciendo un modelo de programación simplificado y un manejo más eficiente de las operaciones de base de datos. Automatiza las consultas CRUD y permite definir consultas más complejas mediante nombres de métodos convencionales o anotaciones, lo que minimiza el código de acceso a datos necesario y mejora la legibilidad.

Lombok: Lombok es una biblioteca de Java que se utiliza para minimizar el código repetitivo, como “getters”, “setters”, constructores, y métodos “toString”, mediante anotaciones simples. Esto no solo hace que el código sea más limpio y fácil de leer, sino que también reduce la posibilidad de errores y mejora la mantenibilidad del código.

Spring Web: Parte del ecosistema de Spring, Spring Web proporciona herramientas robustas para desarrollar aplicaciones web, incluyendo Frameworks como Spring MVC. Ofrece un modelo de desarrollo basado en anotaciones para construir servicios “RESTful” y aplicaciones web de manera eficiente, lo que facilita la gestión de las solicitudes HTTP, la serialización de datos y la interacción con los clientes web.

Spring Security: Es un poderoso y personalizable marco de autenticación y control de acceso que protege las aplicaciones Spring. Spring Security soporta una amplia gama de estrategias de seguridad, incluyendo autenticación HTTP básica, formularios web, y OAuth2. Proporciona un mecanismo extensible para la integración de la seguridad en aplicaciones, garantizando que tanto la seguridad de la aplicación como la de los datos se mantengan de forma rigurosa.

PostgreSQL Driver: Este driver es esencial para permitir que las aplicaciones Java interactúen con la base de datos PostgreSQL. Actúa como un puente entre la aplicación y la base de datos, gestionando la conexión y permitiendo la ejecución de consultas SQL, actualizaciones y transacciones.

TestContainers: Una biblioteca de pruebas para Java que proporciona contenedores Docker ligeros y desechables para bases de datos, sistemas de Kafka, aplicaciones web, o cualquier otra dependencia que pueda ejecutarse en un

contenedor. Esto permite que las pruebas de integración sean reproducibles y aisladas del entorno de desarrollo.

H2 Database: H2 es una base de datos en memoria que es comúnmente utilizada durante el desarrollo y las pruebas debido a su configuración sencilla y rápido desempeño. Proporciona una buena plataforma para pruebas de integración y unitarias, dado que se puede configurar para funcionar en un modo que se inicia y se detiene con la aplicación.

JDBC API: JDBC (Java Database Connectivity) es una API de Java que permite la ejecución de operaciones sobre bases de datos desde el lenguaje Java, independientemente de la base de datos específica que se utilice. Proporciona los métodos para consultar y actualizar datos en una base de datos, y es una parte fundamental del acceso a datos en aplicaciones Java.

5. Proceso de Desarrollo

5.1. Desarrollo Histórico:

Fase de planificación: Al comienzo del proyecto, el grupo se enfocó en definir claramente los objetivos y las funcionalidades de VentureSync. Durante esta fase, se realizaron múltiples sesiones de lluvia de ideas para acordar las tecnologías a utilizar y estructurar el proyecto. Se establecieron roles dentro del equipo para asegurar una distribución equitativa de tareas y responsabilidades.

Primera Iteración (Diseño y desarrollo Inicial): Se comenzó con el diseño del modelo entidad-relación y la configuración inicial del entorno de desarrollo utilizando Spring Boot y otras tecnologías seleccionadas. El equipo desarrolló las primeras versiones de los endpoints de la API, asegurando que cumplieran con los requisitos funcionales básicos.

Desafío: Uno de los principales desafíos encontrados durante esta fase fue la falta de pruebas en los endpoints después de su implementación, continuando con demás aspectos. Sin embargo, posteriormente usamos la herramienta Postman para validar la funcionalidad y seguridad de estos endpoints, lo que llevó a errores no detectados en la primera iteración.

Segunda Iteración (Refinamiento y escalabilidad): En esta etapa, se revisaron y mejoraron las funcionalidades existentes. Se implementaron mejoras en la estructura para apoyar una mayor escalabilidad y se optimizaron las interacciones con la base de datos para mejorar el rendimiento.

Implementación de asincronía y eventos:

En esta sección, se agregaron eventos para el envío de correos electrónicos, cuando: Se crea una publicación, se asigna una publicación a un comentario, se crea una postulación, se crea un comentario perteneciente a un usuario a un startup. De esta forma se mejora la interacción con la plataforma.

Tercera Iteración (Seguridad y despliegue): El foco de la última iteración estuvo en fortalecer las medidas de seguridad y preparar el proyecto para su despliegue. Se implementaron mecanismos de autenticación y autorización usando JWT y Spring Security.

Deployment:

Se logró exitosamente el deployment a AWS, permitiendo así que nuestro servidor backend se despliegue en la nube y de esta manera ejecutar acciones desde el servidor. La última IP generada fue:

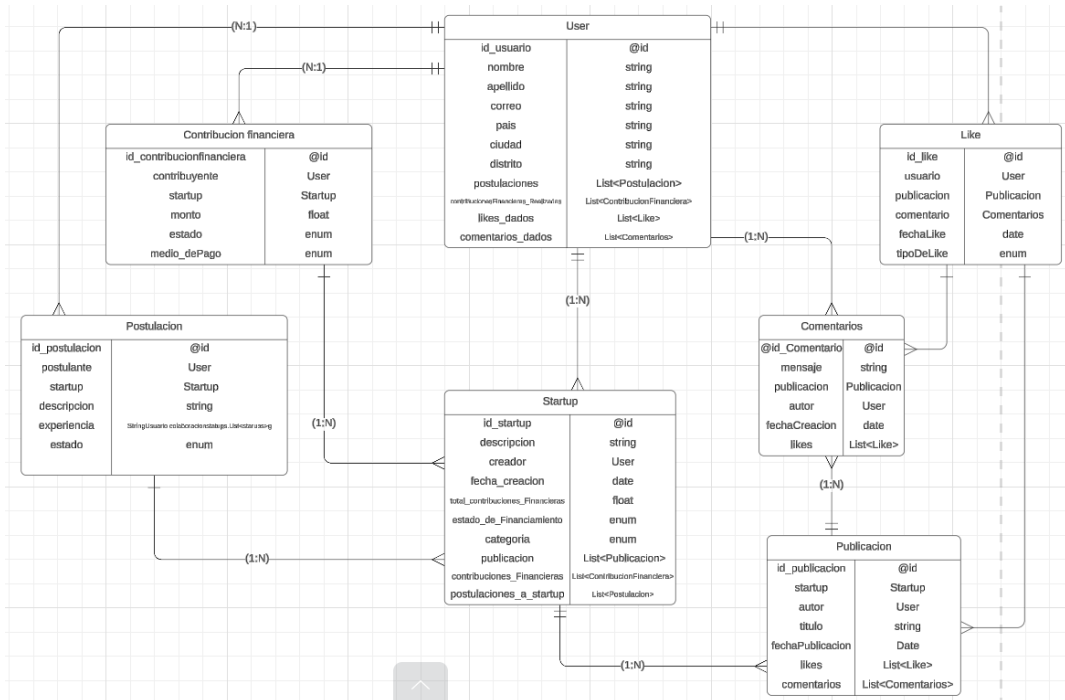
35.173.242.119

Colocando el siguiente protocolo, dominio y puerto, además de los endpoints definidos podremos testear el servidor en la nube.

<http://35.173.242.119:8080/>

6. Modelo Entidad-Relación

6.1 Diagrama ER



6.2 Descripción de Entidades

Entidades y sus atributos:

1. User

`id_usuario`: Identificador único del usuario.

`nombre`: Nombre del usuario.

`apellido`: Apellido del usuario.

`correo`: Dirección de correo electrónico.

`pais`: País de residencia.

`ciudad`: Ciudad de residencia.

`distrito`: Distrito de residencia.

Relaciones:

Posee múltiples Postulaciones.

Posee múltiples Contribuciones Financieras.

Posee múltiples Likes.

Posee múltiples Comentarios.

2. Startup
 - id_startup: Identificador único de la startup.
 - descripcion: Descripción de la startup.
 - creador: Usuario que creó la startup.
 - fecha_creacion: Fecha en que se creó la startup.
 - total_contribuciones_financieras: Suma total de las contribuciones financieras.
 - estado_de_financiamiento: Estado actual del financiamiento (enumerado).
 - categoria: Categoría de la startup.
 - Relaciones:
 - Tiene múltiples Postulaciones.
 - Tiene múltiples Publicaciones.
 - Recibe múltiples Contribuciones Financieras.
3. Postulación
 - id_postulacion: Identificador único de la postulación.
 - postulante: Usuario que realiza la postulación.
 - startup: Startup a la cual se postula.
 - descripcion: Descripción de la postulación.
 - experiencia: Experiencia relevante del postulante.
 - estado: Estado de la postulación (enumerado).
4. Publicación
 - id_publicacion: Identificador único de la publicación.
 - startup: Startup relacionada a la publicación.
 - autor: Usuario que realiza la publicación.
 - título: Título de la publicación.
 - fecha_publicacion: Fecha de la publicación.
 - Relaciones:
 - Tiene múltiples Likes.
 - Tiene múltiples Comentarios.
5. Contribución Financiera
 - id_contribucion_financiera: Identificador único de la contribución.
 - contribuyente: Usuario que contribuye financieramente.
 - startup: Startup beneficiada por la contribución.
 - monto: Cantidad de dinero contribuido.
 - estado: Estado de la contribución (enumerado).
 - medio_de_pago: Método utilizado para la contribución (enumerado).
6. Comentario
 - id_comentario: Identificador único del comentario.
 - mensaje: Mensaje del comentario.
 - publicacion: Publicación asociada al comentario.
 - autor: Usuario que realiza el comentario.
 - fecha_creacion: Fecha en que se creó el comentario.
 - Relaciones:
 - Tiene múltiples Likes.
7. Like

id_like: Identificador único del 'like'.
 usuario: Usuario que da el 'like'.
 publicacion: Publicación a la que se da 'like'.
 comentario: Comentario al que se da 'like'.
 fecha_like: Fecha en que se dio el 'like'.
 tipo_de_like: Tipo de 'like' (enumerado).

Relaciones entre las Entidades:

Usuario y Postulación: Relación de uno a muchos (1:N): Un Usuario puede realizar múltiples Postulaciones a diferentes Startups.

Usuario y Contribución Financiera: Relación de uno a muchos (1:N): Un Usuario puede realizar múltiples Contribuciones Financieras a diferentes Startups.

Usuario y Like: Relación de uno a muchos (1:N): Un Usuario puede dar múltiples Likes a diferentes Publicaciones y Comentarios.

Usuario y Comentario: Relación de uno a muchos (1:N): Un Usuario puede hacer múltiples Comentarios en diversas Publicaciones.

Startup y Postulación: Relación de uno a muchos (1:N): Una Startup puede recibir múltiples Postulaciones de diferentes Usuarios.

Startup y Contribución Financiera: Relación de uno a muchos (1:N): Una Startup puede recibir múltiples Contribuciones Financieras de diferentes Usuarios.

Startup y Publicación: Relación de uno a muchos (1:N): Una Startup puede estar relacionada con múltiples Publicaciones hechas por diversos Usuarios.

Publicación y Like: Relación de uno a muchos (1:N): Una Publicación puede recibir múltiples Likes de diferentes Usuarios.

Publicación y Comentario: Relación de uno a muchos (1:N): Una Publicación puede recibir múltiples Comentarios de diferentes Usuarios.

Comentario y Like: Relación de uno a muchos (1:N): Un Comentario puede recibir múltiples Likes de diferentes Usuarios.

8. Testing

8.5 Niveles de Testing Realizados:

Pruebas Unitarias: Para estas pruebas, utilizamos Mockito. Cada método en nuestros archivos service y repository fue probado exhaustivamente para asegurar su correcto comportamiento. Las pruebas unitarias se centraron en validar la lógica de negocio en los servicios, asegurar el correcto funcionamiento de los métodos CRUD en los repositorios y verificar el comportamiento de los métodos.

Mockito se utilizó específicamente para simular (mock) las dependencias de los componentes, lo que permitió aislar y probar dos entidades “Publicación” y “Like”, de manera independiente sin la necesidad de un entorno de datos real.

Prueba de Integración: La prueba de integración fue realizada para asegurar que los diferentes componentes del sistema funcionaran bien juntos. Para ello, utilizamos Spring Test y TestContainers. Esta prueba se centro en verificar la integración entre el service y la base de datos, asegurar que las llamadas a la API funcionaran correctamente a través de los controllers REST y probar los flujos completos de negocio que involucraban múltiples servicios y repositorios. Usamos testcontainers para el testeo de un repositorio.

8.6 Resultados:

Entre los principales problemas identificados y resueltos encontramos:

Errores en la lógica de negocio: Se encontraron y corrigieron errores en la validación de datos. Por ejemplo, hubo problemas con la lógica de cálculo del estado de financiamiento de los startups.

Problemas de integración: Durante la prueba de integración, se descubrió problemas de mapeo entre las entidades y la base de datos. Estos problemas fueron solucionados ajustando las anotaciones de JPA, revisando las configuraciones de las relaciones entre las entidades.

Fallas en endpoints REST: Algunos endpoints no respondían correctamente debido a configuraciones incorrectas. Estos problemas se detectaron con las pruebas automatizadas de Postman, realizadas posterior a intentar implementar testing. Después de asegurar el correcto funcionamiento de los endpoints, agregamos demás aspectos.

9. Medidas de Seguridad Implementadas

9.1 Seguridad de Datos

Sistema de Autenticación y Autorización Basado en Roles con JWT En nuestro proyecto, implementamos un sistema de autenticación y autorización avanzado utilizando Spring Security y tokens JWT (JSON Web Tokens). Este sistema permite que los usuarios se autenticuen mediante credenciales válidas (como nombre de usuario y contraseña), tras lo cual el sistema genera un token JWT. Este token actúa como una credencial de acceso que encapsula los permisos del usuario (roles) y su identidad, permitiéndole interactuar con la aplicación según los permisos asignados a su rol. Los roles facilitan la implementación de un control de acceso granular, asegurando que los usuarios solo puedan acceder a los recursos y ejecutar acciones que son pertinentes a sus roles en la aplicación. Además, el token JWT tiene un tiempo de expiración, lo que añade una capa adicional de seguridad: el usuario debe reautenticarse una vez que el token expire. Esta configuración no solo refuerza la seguridad, sino que también mejora la experiencia del usuario, manteniendo un equilibrio entre accesibilidad y protección de recursos críticos.

9.2 Prevención de Vulnerabilidades

Inyección SQL: Utilizamos Spring Data JPA, que emplea ORM (Object-Relational Mapping) con consultas parametrizadas, lo que asegura que los datos ingresados por los usuarios se manejen de manera segura y eviten la ejecución de código malicioso en la base de datos.

XSS: Nuestras configuraciones de Spring Web limpian automáticamente las entradas de usuario, asegurando que cualquier script potencialmente dañino sea desactivado antes de que se muestre en las interfaces de usuario. Además, implementamos políticas de Content Security Policy (CSP) que proporcionan un nivel adicional de seguridad al restringir los recursos que el navegador puede cargar.

10. Conclusión

10.1 Logros del Proyecto:

El proyecto VentureSync ha logrado abordar de manera efectiva la desconexión entre emprendedores y recursos esenciales en el mercado. Los logros más destacados incluyen:

- **Conexión eficiente:** La plataforma permite la conexión directa entre emprendedores y una red de recursos financieros y profesionales, facilitando la colaboración y el apoyo necesarios para materializar proyectos innovadores.
- **Estructura:** La plataforma ha logrado realizar funcionalidades las cuales soportan el desarrollo necesario de la interfaz, cumpliendo los objetivos planteados.
- **Seguridad mejorada:** Con la integración de Spring Security, se ha asegurado que la plataforma cumpla con estándares de seguridad, protegiendo los datos y privacidad de los usuarios.

10.2 Aprendizajes Clave:

El desarrollo de VentureSync ha dejado varios aprendizajes significativos para el equipo:

- **Importancia de las Pruebas:** La experiencia resaltó la importancia de realizar pruebas exhaustivas, tanto unitarias como de integración, para garantizar la calidad del software. La falta inicial de pruebas con herramientas como Postman subrayó la necesidad de integrar pruebas automatizadas desde el principio.
- **Colaboración y Comunicación:** El proyecto demostró que una comunicación efectiva y una colaboración estrecha dentro del equipo son cruciales para superar desafíos y cumplir con los plazos establecidos.
- **Adaptabilidad:** La capacidad de adaptarse rápidamente a los problemas y buscar soluciones fue vital para el proyecto. La integración de nuevas herramientas y técnicas, como TestContainers para pruebas de integración, mostró la importancia de adoptar nuevas tecnologías.

10.3 Trabajo Futuro:

Aunque VentureSync ha logrado cumplir con sus objetivos iniciales, existen áreas para futuras mejoras y extensiones:

- **Ampliación de funcionalidades:** Se podrían añadir nuevas funcionalidades, como un sistema de reuniones en línea y foros de discusión, para enriquecer aún más la experiencia de los usuarios.
- **Mejoras en la UX/UI:** Continuar optimizando la interfaz de usuario para hacerla más intuitiva y atractiva, basándose en el feedback de los usuarios.
- **Análisis de datos:** Integrar herramientas de análisis de datos para proporcionar a los usuarios rangos valiosos sobre tendencias y oportunidades en el ecosistema de emprendimiento.

11. Apéndices

11.1 Licencia

Nuestro proyecto se distribuirá de forma gratuita y estará disponible para usuarios de todo el mundo. Al acceder a nuestra plataforma será necesario el iniciar sesión para poder acceder y navegar a través de ella.

11.2 Referencias

Gómez Sánchez, P., & Fernández López, C. (2022). El rol de las plataformas digitales en el fomento del emprendimiento. *Revista de Tecnología y Sociedad*, 40-55.

OECD.org. (2024). *OECD*.

Echevarria Bonet, C., & Álvarez Alonso, P. (2024). Gamificación en tiempos de pandemia: rediseño de una experiencia en educación superior. *Revista Eureka sobre Enseñanza y Divulgación de las Ciencias*.