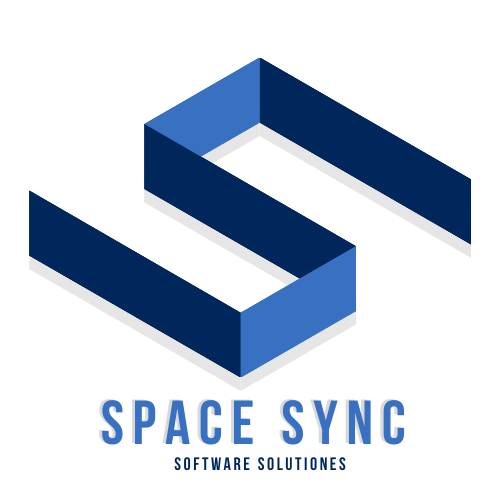
INFORME PORTAFOLIO DE TÍTULO

**PROYECTO:**



**Nombres de alumnos:** Paula Cortés Narváez

Johan Dahlbokum Novelli

**Docente:** Mariluz Rodriguez Donoso

**Carrera:** Ingeniería en Informática.

**Sede:** San Joaquín.

**Índice**

[1. Resumen 3](#_xhvrmpybcd1m)

[2. Abstract 3](#_gtg8bzi8rswp)

[3. Descripción del Proyecto 4](#_fm520owe8mn3)

[4. Competencias del Perfil de Egreso 5](#_vqa163vdfd7)

[5. Relación del Proyecto con los Intereses Profesionales 6](#_b9hfjpxyf7xk)

[6. Objetivos del Proyecto 6](#_3hibaemdmj5e)

[7. Marco de trabajo 8](#_vx0g60gmqnu2)

[Product Backlog 9](#_t68jg1py7k9y)

[Tabla de requerimientos 9](#_c09b4khog1ot)

[Tipo de arquitectura 10](#_ws2b50fnnsfy)

[Justificación de la Elección Arquitectónica 10](#_1rd054m0nz7h)

[8. Plan de Trabajo 11](#_fq257laa96xo)

[Visualización del Trabajo 12](#_tkm2g7yqz4ys)

[Hitos Clave 12](#_t3orh13131bz)

[Calendario 13](#_exqxrogx2xzq)

[9. Stack tecnológico 14](#_m2ebvx81g7q)

[Backend 14](#_hdeal6przunn)

[Base de datos 14](#_y6wep7jzlqpb)

[Frontend 14](#_6cbjk8qrshv9)

[10. Análisis de riesgos 16](#_tmupxtn0jyju)

[1. Riesgos Técnicos 16](#_2d209rviwoqf)

[2. Riesgos Organizacionales 17](#_e6slli7x9m1b)

[3. Riesgos Comerciales 17](#_2hz46lgag4y1)

[4. Riesgos Externos 17](#_6jbw6d51rktt)

[11. Viabilidad del proyecto 18](#_4isrw45zn6f2)

[12. Conclusiones Individuales 21](#_rexnp3hitub5)

[13. Reflexión 21](#_ia02oa1teyhm)

# 

# **Resumen**

Este informe detalla el desarrollo del proyecto Space Sync, una innovadora plataforma de gestión de reservas de espacios diseñada para optimizar la utilización de recursos y evitar conflictos de horarios en entornos educativos y corporativos. El sistema, basado en una arquitectura centralizada y fácil de usar, permite una gestión eficiente y precisa de las reservas mediante actualizaciones de disponibilidad en tiempo real, historial y generación de reportes detallados. La plataforma aborda los desafíos comunes en la administración de reservas, como la falta de visibilidad y la dificultad para coordinar horarios, proporcionando una solución efectiva que mejora significativamente la eficiencia operativa y la gestión de recursos en las instituciones. Al integrar características avanzadas y una interfaz intuitiva, Space Sync busca facilitar la planificación y la administración de espacios, optimizando su uso y reduciendo la posibilidad de errores y malentendidos.

**Palabras clave:** *gestión de reservas, optimización de recursos, disponibilidad en tiempo real, reportes detallados, administración de espacios, eficiencia operativa.*

# **Abstract**

This report outlines the development of the Space Sync project, an advanced space reservation management platform aimed at optimizing resource utilization and preventing scheduling conflicts in educational and corporate settings. The system is built on a centralized, user-friendly architecture that facilitates efficient and accurate reservation management through real-time availability updates, history, and comprehensive reporting. The platform addresses common challenges in reservation management, such as visibility issues and coordination difficulties, by providing an effective solution that significantly enhances operational efficiency and resource management within institutions. By incorporating advanced features and an intuitive interface, Space Sync is designed to streamline planning and space management, improving space utilization and minimizing the risk of errors and misunderstandings.

**Keywords*:*** *reservation management, resource optimization, real-time availability, comprehensive reporting, space administration, operational efficiency.*

# **Descripción del Proyecto**

**El proyecto Space Sync** se centra en el desarrollo de una plataforma digital para la gestión de reservas de espacios. El objetivo principal es optimizar el uso de los espacios disponibles, evitando conflictos de horarios y facilitando una administración más eficiente de las reservas.

**Justificación de su Relevancia**

En diversos entornos, la gestión eficaz de los espacios es esencial para maximizar su uso y evitar conflictos que puedan afectar la productividad. Muchas plataformas actuales tienen limitaciones en términos de integración, flexibilidad y facilidad de uso, lo que puede resultar en ineficiencias y malentendidos. Space Sync ofrece una solución integral que se distingue por varias características clave:

* **Disponibilidad en Tiempo Real:** Muestra la disponibilidad de los espacios al momento, lo que previene conflictos de horarios y facilita la planificación.
* **Historial:** Registra y almacena todas las actividades y cambios relacionados con las reservas, proporcionando una visión completa para usuarios y administradores.
* **Reportes Detallados:** Genera reportes sobre la utilización de los espacios, lo que ayuda en la optimización de recursos y en la toma de decisiones informadas.

**Comparativa con Soluciones Existentes**

1. [**Kantoor**](https://www.kantoor.cl/)

* **Ventaja:** Diseñado para escalar eficientemente con el crecimiento de la organización. Ofrece soporte robusto y una base de usuarios amplia, con actualizaciones regulares.
* **Limitaciones:** Personalización limitada en comparación con soluciones diseñadas específicamente para el usuario. Los costos pueden ser elevados, especialmente para organizaciones con presupuestos limitados.

1. [**Robin**](https://robinpowered.com/)**:**

* **Ventajas:** Buena integración con calendarios y herramientas de colaboración.
* **Limitaciones:** La configuración inicial puede ser compleja y la personalización es limitada.

1. [**Teem**](https://www.teem.com/)**:**

* **Ventajas:** Amplias funcionalidades y buena integración con herramientas de comunicación.
* **Limitaciones:** El costo puede ser alto para organizaciones pequeñas y la curva de aprendizaje puede ser pronunciada.

**Por qué Space Sync es Mejor:**

* **Facilidad de Uso:** La plataforma está diseñada para ser intuitiva, facilitando su adopción tanto por parte del personal administrativo como de los usuarios finales.
* **Costo Efectivo:** Ofrecemos una solución más accesible en comparación con las alternativas más costosas, sin sacrificar funcionalidades esenciales.

# **Competencias del Perfil de Egreso**

El proyecto Space Sync está diseñado para optimizar la gestión de reservas de espacios. Este proyecto refleja competencias clave del perfil de egreso:

* **Análisis y Diseño de Requerimientos:** Definición precisa de las necesidades para crear la plataforma Space Sync.
* **Seguridad y Calidad del Software:** Garantía de un sistema robusto y seguro para el manejo de reservas.
* **Gestión de Proyectos Informáticos:** Organización y ejecución efectiva del proyecto, cumpliendo plazos y objetivos.

# **Relación del Proyecto con los Intereses Profesionales**

El proyecto es una buena oportunidad para aplicar y desarrollar nuestras habilidades en los roles de Frontend y Backend.

Para el equipo, el desarrollo Backend es crucial para diseñar una arquitectura robusta, gestionar bases de datos y asegurar que la lógica de negocio sea eficiente y segura. Al mismo tiempo, el enfoque en Frontend nos permite crear una interfaz de usuario intuitiva y atractiva, mejorando la experiencia general del usuario con la plataforma.

Trabajar en ambas áreas permite que cada miembro del equipo aporte su experiencia para construir una solución completa y eficaz. Esta combinación asegura que el proyecto ofrezca una infraestructura sólida y una experiencia de usuario excepcional, maximizando la funcionalidad y usabilidad del sistema.

## **Objetivos del Proyecto**

El objetivo general del proyecto es desarrollar e implementar una plataforma de gestión de reservas que optimice el uso de las reservas, mejore la eficiencia operativa y permita una experiencia de usuario fluida.

1. **Diseñar una plataforma eficiente y escalable**

Descripción: Gestionar grandes volúmenes de reservas y desarrollar soluciones adaptables a diferentes tipos de entornos (corporativo, educativo). Utiliza **Django y PostgreSQL** para garantizar el rendimiento e implementa una arquitectura modular que le permite escalar fácilmente en función del crecimiento de usuarios y reservas.

Consistencia: seguir los principios de ingeniería de software garantiza que la plataforma sea sólida y escalable para respaldar el crecimiento de los usuarios sin sacrificar el rendimiento.

1. **Desarrollar una interfaz de usuario intuitiva**

Descripción: Cree interfaces basadas en las mejores prácticas de accesibilidad y UX/UI diseñadas tanto para administradores como para usuarios finales. Esta implementación utiliza **Astro y Tailwind CSS** para optimizar la navegación con controles y estructura visual claros.

Coherencia: La interfaz de usuario está diseñada para ser fácil de usar y eficiente, brindando una experiencia fluida y accesible que minimiza la curva de aprendizaje para los usuarios.

1. **Funcionalidad de reserva en tiempo real**

Descripción: Sistema de reservas que permite consultar disponibilidad y reservar en tiempo real, se podrá ver el historial mediante una **API RESTful** conectada a **PostgreSQL.**  
  
Consistencia: Optimiza la gestión del espacio y reduce conflictos de reservas.

1. **Informes y análisis de uso**

Descripción: Módulos que generan análisis detallados de ocupación y uso del espacio, facilitando la toma de decisiones basada en datos con gráficos claros.  
  
Consistencia: Proporciona a los administradores información precisa y oportuna para la optimización de recursos.

1. **Soporte y formación inicial**

Descripción: Plan de soporte técnico y capacitación que incluye tutoriales, documentación detallada y sesiones personalizadas para una adopción fluida de la plataforma.  
  
Consistencia: Garantiza que los usuarios aprovechen al máximo las funciones desde el primer día.

# **Marco de trabajo**

El Marco de Trabajo del Proyecto define la estructura y metodología que guiará el desarrollo del proyecto Space Sync. En esta fase, hemos decidido adoptar la metodología ágil Kanban en lugar de Scrum. Esta elección se basa en varios factores clave que hacen que Kanban sea más adecuado para nuestro equipo reducido y nuestras necesidades específicas.

**Justificación para el Uso de Kanban**

1. **Tamaño del Equipo**: Nuestro equipo está compuesto por solo dos miembros, Johan Dahlbokum y Paula Cortés, junto con un externo para pruebas de QA. Con un equipo tan pequeño, Kanban proporciona una mayor flexibilidad en la gestión de tareas, permitiendo una adaptación rápida sin la necesidad de las estructuras más rígidas que se requieren en Scrum.
2. **Fluidez en el Trabajo**: Kanban se centra en la visualización del flujo de trabajo y la gestión de tareas en tiempo real, lo que permite a nuestro equipo adaptarse rápidamente a cambios en las prioridades y necesidades del proyecto. Esto es especialmente útil en un entorno dinámico donde las demandas pueden cambiar con frecuencia.
3. **Eliminación de Sprints:** A diferencia de Scrum, donde el trabajo se divide en sprints de duración fija, Kanban permite una entrega continua. Esto significa que podemos lanzar incrementos de trabajo de manera más frecuente y eficiente, lo que se alinea mejor con nuestra necesidad de implementar funciones rápidamente y obtener retroalimentación continua.
4. **Menos Ceremonias:** Kanban minimiza la cantidad de reuniones y ceremonias requeridas en Scrum, lo que permite a nuestro equipo concentrarse más en el desarrollo y menos en la gestión del proceso. Esto es ideal para nuestro equipo pequeño, donde el tiempo es un recurso valioso.
5. **Adaptación y Mejora Continua:** La naturaleza flexible de Kanban nos permitirá realizar mejoras continuas en el proceso sin las revisiones formales al final de cada sprint que requiere Scrum. Esto se traduce en una adaptación más fluida a la retroalimentación del QA externo y de los usuarios finales

**Documentación en Kanban**

En lugar de tener un Product Backlog formal como en Scrum, utilizaremos un Tablero Kanban que representará visualmente el flujo de trabajo, con columnas que representen los diferentes estados de las tareas (Por hacer, En progreso, Completado). La documentación será más ligera, centrada en:

1. **Visualización del Progreso**: Utilizaremos un tablero Kanban (físico o digital) para visualizar el estado de las tareas y facilitar la comunicación entre los miembros del equipo. Para la visualización, ocuparemos git pro
2. **Historias de Usuario:** Crearemos y mantendremos historias de usuario que describan las funcionalidades y requisitos del sistema, alineadas con los objetivos del proyecto.

### **Product Backlog**

El Product Backlog es un elemento clave en la gestión del proyecto Space Sync, que actúa como un repositorio dinámico y priorizado de todos los requisitos, funcionalidades, mejoras y tareas necesarias para el desarrollo del producto.   
  
Este documento refleja la visión estratégica del proyecto, alineando los esfuerzos del equipo con los objetivos de negocio y asegurando una planificación efectiva. Se actualiza y refina de manera continua para responder a los cambios en el entorno del proyecto, optimizando la entrega de valor y garantizando que cada iteración se ejecute con la máxima eficiencia, cumpliendo con los más altos estándares de calidad y expectativas.

Para una revisión detallada, puede acceder al documento completo a través del siguiente enlace: [**Product Backlog.xlsx**](https://docs.google.com/spreadsheets/d/1XliobUirF8SJ6xNb0C7egQrhzpbhJD6p/edit?usp=drive_link&ouid=105430398268764009452&rtpof=true&sd=true)

### **Tabla de requerimientos**

La siguiente tabla detalla los requerimientos específicos identificados para el desarrollo del proyecto Space Sync. Cada requerimiento ha sido evaluado y priorizado para asegurar que se alineen con los objetivos estratégicos del proyecto y las expectativas del cliente.   
  
Esta tabla proporciona una descripción clara de cada requerimiento, incluyendo su prioridad, responsables, estado actual, y cualquier dependencia relevante. Esta estructura facilita la gestión efectiva del backlog, permitiendo al equipo de desarrollo mantenerse enfocado en la entrega de valor continuo y la adaptación a los cambios en el entorno del proyecto.

Para una revisión detallada, puede acceder al documento completo a través del siguiente enlace: [**Tabla de requerimientos**](https://docs.google.com/spreadsheets/d/1e60aTP_1KFMpitrrsThlRzOQMh5gakYtZSauQ6H9h2M/edit?usp=drive_link)

### **Tipo de arquitectura**

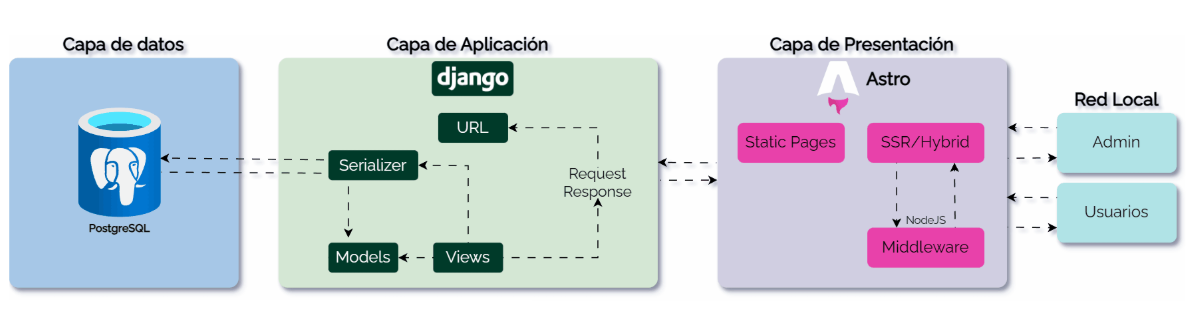
El proyecto sigue una **arquitectura cliente-servidor desacoplada**, en la que el frontend y el backend operan de forma independiente pero se comunican a través de APIs. Esta arquitectura fue elegida por su capacidad para facilitar el mantenimiento, permitir escalabilidad y proporcionar flexibilidad en futuras implementaciones.

* **Arquitectura Cliente-Servidor Desacoplada**: Este enfoque separa la interfaz de usuario (frontend) de la lógica del servidor (backend). El frontend, desarrollado con Astro y Tailwind CSS, se encarga de la experiencia visual y la interacción con el usuario. Por otro lado, el backend, gestionado con Django y PostgreSQL, maneja las operaciones del servidor y la base de datos, proporcionando datos y servicios a través de APIs RESTful utilizando Django REST Framework.

### **Justificación de la Elección Arquitectónica**

1. **Independencia de Componentes**: Al desacoplar el frontend y el backend, se facilita el trabajo en ambos de manera simultánea sin dependencias rígidas, lo que permite mayor flexibilidad para evolucionar la solución.
2. **Escalabilidad**: Cada parte de la arquitectura (frontend y backend) puede ser escalada de manera independiente según la demanda, lo que optimiza el uso de recursos y mejora la capacidad de respuesta del sistema.
3. **Mantenimiento y Actualización**: La separación entre frontend y backend permite realizar cambios en una parte sin afectar directamente a la otra, lo que facilita el mantenimiento y la introducción de nuevas características sin riesgos significativos.
4. **Integración con Tecnologías**:

* **Frontend (Astro y Tailwind CSS)**: Ofrecen rapidez en la generación de contenido estático, lo que mejora la experiencia del usuario, junto con un diseño flexible y responsivo.
* **Backend (Django y PostgreSQL)**: Django proporciona una estructura robusta y segura para manejar las solicitudes y la lógica del sistema, mientras que PostgreSQL asegura un manejo eficiente de los datos relacionales.



# **Plan de Trabajo**

**Flujo de Trabajo Kanban:**

1. **Definición del Proyecto (23 de agosto - 3 de septiembre):**

* **Establecer Visión:** Definir la visión, objetivos y alcance del proyecto.
* **Recopilación de requisitos:** Reunir y documentar los requisitos de las partes interesadas.
* **Crear Backlog Inicial:** Listar y priorizar las funcionalidades y tareas que se abordarán.

1. **Diseño y Desarrollo Inicial (4 de septiembre - 10 de octubre):**

* **Diseño de la Arquitectura:** Definir la estructura del sistema y la interfaz de usuario.
* **Desarrollo de Funcionalidades Básicas:** Implementación inicial de las características clave y diseño preliminar.
* **Pruebas Iniciales:** Realizar pruebas básicas para verificar la funcionalidad desarrollada.

1. **Desarrollo y Refinamiento (11 de octubre - 19 de noviembre):**

* **Implementación de Características Adicionales:** Añadir funcionalidades y mejoras en la interfaz.
* **Retroalimentación Continua:** Recopilar y analizar retroalimentación de los usuarios para ajustes.
* **Pruebas exhaustivas:** Realizar pruebas más completas para asegurar la calidad del sistema.

1. **Preparación para el Lanzamiento (20 de noviembre - 10 de diciembre):**

* **Finalización del Desarrollo:** Completar cualquier funcionalidad pendiente y corregir errores.
* **Capacitación a Usuarios:** Preparar material de capacitación y entrenar a los usuarios finales.
* **Pruebas Finales:** Realizar pruebas en el entorno de producción para asegurar el correcto funcionamiento.

1. **Lanzamiento y Post-Lanzamiento (11 de diciembre en adelante):**

* **Despliegue del Sistema:** Implementar el sistema en el entorno en vivo.
* **Soporte Técnico:** Ofrecer soporte técnico y resolver problemas iniciales.
* **Evaluación y Mejora Continua:** Monitorear el sistema y aplicar mejoras basadas en la retroalimentación de usuarios.

### **Visualización del Trabajo**

El flujo de trabajo se puede representar en un tablero Kanban con las siguientes columnas:

* **Por Hacer:** Tareas que están planificadas pero aún no comenzadas.
* **En Progreso:** Tareas que están siendo trabajadas actualmente.
* **En Revisión:** Tareas que han sido completadas y están siendo revisadas o probadas.
* **Completado:** Tareas que han sido finalizadas y aprobadas.

### **Hitos Clave**

* **3 de septiembre:** Completar la recopilación de requisitos y crear el backlog inicial.
* **10 de octubre:** Finalizar el diseño preliminar y desarrollo básico.
* **19 de noviembre:** Concluir la implementación de funcionalidades y prepararse para el lanzamiento.
* **11 de diciembre en adelante:** Realizar el lanzamiento y comenzar soporte post-lanzamiento.

### **Calendario**

# **Stack tecnológico**

El stack tecnológico para el proyecto **Space Sync** se compone de las siguientes tecnologías, adaptadas para optimizar el desarrollo tanto del frontend como del backend, además de la gestión de datos:

### **Backend**

* **Framework:** **Django**

1. **Descripción:** Django es un framework de alto nivel para Python que facilita el desarrollo rápido y limpio de aplicaciones web. Ofrece una estructura robusta y muchas funcionalidades integradas para la creación de aplicaciones seguras y escalables.
2. **Ventajas:** Proporciona una arquitectura bien definida, incluye herramientas de administración integradas, y tiene una gran comunidad de soporte y una amplia documentación.

* **Lenguaje de Programación:** **Python**

1. **Descripción:** Python es un lenguaje de programación de alto nivel, conocido por su sintaxis clara y su enfoque en la legibilidad del código. Es ampliamente utilizado en el desarrollo web y en la ciencia de datos.
2. **Ventajas:** Facilita el desarrollo rápido y la integración con otras herramientas y tecnologías, ofreciendo gran flexibilidad y eficiencia.

### **Base de datos**

* **Sistema de Gestión de Bases de Datos: PostgreSQL**

1. **Descripción:** PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto, conocido por su robustez y soporte avanzado de características como transacciones ACID y replicación. Utiliza SQL para la gestión y manipulación de datos.
2. **Ventajas:** Ofrece una gran escalabilidad y es altamente personalizable, permitiendo una gestión eficiente de datos estructurados. Su capacidad para manejar consultas complejas y grandes volúmenes de datos lo convierte en una opción ideal para aplicaciones de alto rendimiento, como Space Sync, facilitando la integración con Astro y otras tecnologías.

### **Frontend**

* **Framework: Astro**

1. **Descripción:** Astro es un generador de sitios estáticos que permite construir aplicaciones web rápidas y optimizadas, utilizando componentes de diferentes frameworks y librerías.
2. **Ventajas:** Optimiza el rendimiento al cargar solo el JavaScript necesario, facilita la creación de páginas estáticas y permite la integración de componentes de otros frameworks como React o Vue.

* **Lenguaje de Programación: JavaScript**

1. **Descripción:** JavaScript es un lenguaje de programación de alto nivel que permite la creación de contenido dinámico y la interactividad en las aplicaciones web.
2. **Ventajas:** Amplia compatibilidad en navegadores, soporte para programación orientada a objetos y funcional, y una gran comunidad con abundantes recursos y bibliotecas.

Este stack tecnológico no sólo es adecuado para el desarrollo del proyecto **Space Sync**, sino también para las capacidades y preferencias del equipo:

* **Simplicidad y Colaboración**: Python y Django son conocidos por su facilidad de uso y legibilidad, lo que facilita la colaboración dentro del equipo. Todos los desarrolladores pueden trabajar de manera eficiente con código limpio y organizado.
* **Escalabilidad y Rendimiento**: PostgreSQL y Django proporcionan una solución robusta y escalable para manejar el backend y las bases de datos, asegurando que el sistema pueda crecer según las necesidades del cliente.
* **Rendimiento en el Frontend**: Astro y Tailwind CSS permiten crear interfaces rápidas y eficientes, lo que asegura que los usuarios finales tengan una experiencia fluida y optimizada.
* **Flexibilidad y Evolución**: La arquitectura desacoplada entre el frontend y el backend permite que el equipo trabaje en diferentes áreas del proyecto sin interferencias, facilitando el mantenimiento y la evolución de la plataforma.

# **Análisis de riesgos**

El proyecto Space Sync, diseñado para la gestión eficiente de espacios en instituciones educativas y empresariales, enfrenta una serie de riesgos que deben ser identificados y gestionados para asegurar su éxito. A continuación, se presenta un análisis detallado de los riesgos, junto con las estrategias de mitigación y las acciones de contingencia para cada uno de ellos.

### **1. Riesgos Técnicos**

**Diseño de interfaz**

* **Riesgo**: Fallos en el diseño o funcionalidades poco intuitivas que afecten la experiencia del usuario.
* **Mitigación (Prevención)**: Realizar pruebas de usabilidad frecuentes y recopilar feedback de los usuarios.
* **Contingencia (Acción)**: Rediseñar la interfaz o realizar ajustes basados en los resultados de las pruebas y el feedback recibido.

**Rendimiento**

* **Riesgo**: Problemas de rendimiento, como tiempos de carga elevados o comportamientos erráticos en dispositivos móviles.
* **Mitigación (Prevención)**: Optimización del frontend mediante técnicas como el lazy loading y la minificación de archivos.
* **Contingencia (Acción)**: Ajustar el rendimiento con medidas adicionales de optimización o realizar una refactorización del código si es necesario.

**Seguridad**

* **Riesgo**: Vulnerabilidades en el sistema que expongan datos sensibles.
* **Mitigación (Prevención)**: Implementar buenas prácticas de seguridad, como autenticación robusta y encriptación de datos.
* **Contingencia (Acción)**: Corregir los fallos de seguridad de forma inmediata y realizar un análisis continuo de posibles vulnerabilidades futuras.

### **2. Riesgos Organizacionales**

**Dependencia de equipo**

* **Riesgo**: Falta de comunicación o retrasos debido a la dependencia de un solo miembro del equipo.
* **Mitigación (Prevención)**: Mantener reuniones de sincronización regulares y promover el trabajo en equipo para evitar cuellos de botella.
* **Contingencia (Acción)**: Redistribuir las tareas entre los miembros del equipo y considerar la contratación de apoyo temporal en caso de ser necesario.

**Capacitación del equipo**

* **Riesgo:** Dificultades para adaptarse a nuevas tecnologías, como Astro, debido a la falta de experiencia del equipo.
* **Mitigación (Prevención)**: Proporcionar capacitación adecuada al equipo y asignar roles en función de las competencias de cada miembro.
* **Contingencia (Acción)**: Realizar capacitaciones adicionales o contratar consultores externos para apoyar en la formación y desarrollo de habilidades específicas.

### **3. Riesgos Comerciales**

**Competencia**

* **Riesgo**: Aparición de competidores con soluciones similares, lo que podría afectar la cuota de mercado del proyecto.
* **Mitigación (Prevención)**: Innovar con nuevas funcionalidades y mantener un enfoque centrado en las necesidades del usuario para diferenciarse en el mercado.
* **Contingencia (Acción)**: Adaptar la estrategia comercial y ofrecer diferenciadores clave para mantener la competitividad.

**Modelo de negocio**

* **Riesgo**: No alcanzar las proyecciones de monetización inicialmente planteadas.
* **Mitigación (Prevención)**: Evaluar el modelo de negocio de manera periódica y ajustarlo según las condiciones del mercado.
* **Contingencia (Acción)**: Modificar o diversificar las formas de monetización basándose en el análisis del mercado para asegurar ingresos sostenibles.

### **4. Riesgos Externos**

**Cambios regulatorios**

* **Riesgo**: Cambios en normativas o leyes, como GDPR o normativas locales, que afecten el uso del sistema.
* **Mitigación (Prevención)**: Mantenerse actualizado sobre las regulaciones vigentes y posibles cambios que puedan afectar el sistema.
* **Contingencia (Acción)**: Ajustar el sistema y las políticas internas para cumplir con las nuevas regulaciones en caso de ser necesario.

**Fallos en infraestructura**

* **Riesgo**: Caídas de servidores o servicios de terceros que afecten la disponibilidad del sistema.
* **Mitigación (Prevención):** Implementar redundancias y utilizar servicios en la nube con alta disponibilidad para reducir riesgos de caídas.
* **Contingencia (Acción):** Migrar temporalmente a otros servicios o servidores hasta que el problema sea resuelto.

Para una revisión detallada, puede acceder al documento completo a través del siguiente enlace: [**Tabla de riesgos**](https://docs.google.com/spreadsheets/d/14QXhqns7B0MKLu8FlbskYBryZqcXU7K4peXQ1tmwFB0/edit?usp=drive_link)

# **Viabilidad del proyecto**

Este análisis evalúa la viabilidad económica y financiera del proyecto centrándose en la estrategia de precios, el flujo de caja, los costos operativos y las métricas de rentabilidad. El objetivo es proporcionar una visión clara de la sostenibilidad y el potencial retorno de la inversión a largo plazo.

Los aspectos importantes de la viabilidad del proyecto se enumeran a continuación.

**Estrategia de precios:**

1. **Fase inicial (años 1-2):** se ofrece un precio asequible de **USD $1250** para atraer clientes y generar ventas. Esto le permite establecer relaciones y generar su primer flujo de ingresos que impulsará su crecimiento.
2. **Fase de integración (años 3 a 6):** una vez que el producto esté en funcionamiento y se establezca una reputación, se espera que las tarifas aumenten a **USD $4000** a medida que ajustamos los precios al valor real del servicio. Resumen del flujo de caja
3. **Inversión inicial (año 0):** **USD $8.000**. Cubre los costos de desarrollo, infraestructura y comercialización.

**Beneficio de ventas:**

| **Año** | **Valor Total / Ingresos** |
| --- | --- |
| 1er | USD $2,500 |
| 2do | USD $3,750 |
| 3er | USD $5,000 |
| 4to | USD $16,000 |
| 5to | USD $20,000 |
| 6to | USD $20,000 |

**Gastos operativos**

Los costos operativos se mantienen en **USD $592** por año, lo que reduce la inversión en marketing y soporte.

**Beneficio antes de impuestos**

Los ingresos antes de impuestos aumentaron significativamente de **USD $1,907** en el primer año a **USD $19,989** en el sexto año, lo que refleja el impacto positivo de los precios.

**Impuestos e ingresos después de impuestos**

Con el impuesto del 25% aplicado, su ganancia después de impuestos aumenta de **USD $1,430** en el primer año a **USD $14,991**

**Indicadores de rentabilidad**

1. **VPN (valor actual neto): USD $29,194.** Demuestra que el proyecto generará un valor significativo superior a la inversión original. **TIR (Tasa Interna de Retorno): 58,67%**, que es significativamente mayor que el costo de capital, lo que indica que el proyecto es altamente rentable.
2. Recuperación de la inversión: Se estima que la recuperación de la inversión se producirá en **2 años, 6 meses y 15 días**, lo cual es razonable y atractivo para los inversores.

**Ventajas de la estrategia de precios.**

1. Adquisición de clientes: los precios iniciales más bajos pueden atraer clientes que dudan en invertir en nuevos servicios.
2. Crecimiento de ventas sostenible: una vez establecida su base de clientes, aumente sus precios a un nivel más sostenible para garantizar un flujo de cajas constante y creciente.

Para un análisis más detallado, las plantillas brindan una visión clara de los flujos de efectivo, los costos y las proyecciones financieras del proyecto. : [**Plantilla de costos**](https://docs.google.com/spreadsheets/d/1vlTzFPmr7QIsQZ76qUJ9fAWTWIg0MSIAICFd-4bnWzQ/edit?usp=drive_link)

# **Pruebas QA**

En esta sección se describe el proceso de **Pruebas de Calidad (QA)** realizado en el proyecto **Space Sync** para garantizar que el sistema funcione de manera óptima y cumpla con los requisitos definidos. Estas pruebas fueron llevadas a cabo por **Kevin Aguirre**, utilizando la herramienta **Selenium** para la automatización de escenarios clave que simulan interacciones reales de los usuarios.

El objetivo principal de estas pruebas es validar aspectos críticos de la aplicación, incluyendo la funcionalidad de los formularios, la navegación entre vistas, el manejo de sesiones, y la adaptabilidad de la interfaz en diferentes dispositivos. Además, se busca identificar posibles errores o inconsistencias que puedan afectar la experiencia del usuario y el rendimiento del sistema.

A continuación, se detallan los tipos de pruebas realizadas y los resultados obtenidos.

# **Early Adopter**

En esta sección se destaca la participación de **Francisco Iturra Piceros**, **Ingeniero en Computación e Informática**, como **early adopter** del proyecto **Space Sync**. Su enfoque se centró en evaluar el sistema desde la perspectiva de un usuario que busca gestionar la **reserva de espacios** de manera eficiente, brindando retroalimentación clave para mejorar la experiencia del usuario final.

Francisco probó las funcionalidades principales del sistema, como:

* **Inicio de sesión y autenticación segura.**
* **Búsqueda y reserva de espacios disponibles.**
* **Acceso al dashboard y visualización de estadísticas.**

Gracias a su experiencia técnica, Francisco identificó mejoras importantes en la usabilidad del sistema, incluyendo:

* **Mayor claridad en los flujos de reserva.**
* **Optimización de la visibilidad de opciones en dispositivos móviles.**
* **Sugerencias para mejorar la respuesta del sistema en situaciones de alta demanda.**

Su rol como early adopter permitió validar el cumplimiento de los objetivos del sistema en un entorno real, garantizando que el proyecto **Space Sync** responda adecuadamente a las necesidades de profesionales que dependen de una solución confiable para la reserva y gestión de espacios. Su contribución fue fundamental para alinear la plataforma con los estándares esperados por los usuarios.

# **Conclusiones Individuales**

**Paula Cortés' Conclusion (Frontend):**

Working on the Space Sync project has been an enriching experience that allowed me to apply advanced techniques in interface design using Astro. I successfully implemented key features such as real-time visualization and accessibility, which were exciting challenges. Prioritizing responsive design and performance optimization were critical to delivering a smooth and efficient user experience. Collaboration with Johan on integrating the frontend with the backend was pivotal in resolving technical issues and achieving a functional system. I am proud of the final results and the quality we achieved in this project.

**Johan Dahlbokum's Conclusion (Backend and Project Leader):**

In the Space Sync project, my role as a backend developer and project leader was essential for designing and building a robust architecture using Django. Implementing business logic and integrating PostgreSQL ensured efficient reservation management and system reliability. The Scrum methodology facilitated agile project management, enabling quick iterations and adjustments to emerging needs. Collaboration with Paula to synchronize the backend with the frontend was key to consolidating the system. I am very pleased with the technical direction and the final outcomes, which fully meet the project's objectives.

# **Reflexión**

The Space Sync project has been successfully completed, achieving results that reflect a cohesive team effort and solid technical planning. From the outset, we focused on establishing a strong foundation for comprehensive space reservation management, meeting all the outlined objectives.

The strategic selection of technologies, such as PostgreSQL for the backend and Astro for the frontend, proved effective in ensuring seamless integration and scalability. The Scrum methodology enabled well-organized task coordination, ensuring that deadlines and goals were consistently met in each sprint.

Additionally, the early incorporation of feedback and a focus on initial testing helped refine functionalities and optimize the user experience. This methodical and technical approach not only fulfilled expectations but also delivered a robust and efficient final product ready to meet user needs.

Space Sync positions itself as an innovative and comprehensive solution, ready for deployment and scalable to adapt to future requirements.