Universidad del Valle de Guatemala

Facultad de Ingeniería

Departamento de Ciencia de la Computación

Ingeniería de Software I

DESIGN BETTER

PLATAFORMA DE DISEÑO Y PERSONALIZACIÓN DE MODA

Pablo José Méndez Alvarado – 23975

Luis Fernando Palacios López – 239333

Roberto Samuel Nájera Marroquín – 23781

André Emilio Pivaral López – 23574

Catedrático: Erick Francisco Marroquín Rodríguez

Sección: 20

Nueva Guatemala de la Asunción, 7 de abril de 2024

Índice

Índice 1
Product Backlog 2
Épica 1: Gestión de Usuarios 2
Épica 2: Personalización de Prendas 2
Épica 3: Visualización Realista (3D) 2
Épica 4: Seguimiento de Pedidos 3
Épica 5: Comunicación y Colaboración 4
Épica 6: Administración y Gestión de Plataforma 4
Épica 7: Infraestructura Técnica 4
Épica 8: Gestión de Pagos 5
Épica 9: Despliegue y Publicación del Sistema 5
Épica 10: Mantenimiento y Escalabilidad 5
User Story MappingSprint Backlog 6
Pila del Sprint 6
Calendario 13
Elección de Contenedor 14
Justificación Detallada: 14
Evidencia del uso de las técnicas de scrum. 15
Resultados del Sprint 17
Software Funcionando 17
Tareas Concluidas 17
Tareas planificadas, pero no concluidas 18
Repositorios con el código desarrollado: 19
Métricas del Sprint 19
Gráfico de Sprint Burndown de puntos de historia 19
Gráfico de Sprint Burndown de Recuento de Incidencias 19
Reporte de Velocidad 20
Éxito del Sprint 21
Evidencia de Pruebas con Usuarios finales 22
Reflexión del Desempeño del Equipo en el Sprint I 23
Aspectos Clave Para Mejorar para el Próximo Sprint: 23

Product Backlog

Épica 1: Gestión de Usuarios
Prioridad: Alta

Historia de usuario: Como usuario (diseñador o cliente), quiero poder registrarme e iniciar sesión para acceder a funcionalidades específicas de la plataforma.

Tareas:

- Crear formulario de registro de usuarios.
- Implementar autenticación vía correo electrónico (con confirmación).
- Integrar inicio de sesión con redes sociales (Google, Facebook).
- Crear sistema para recuperación de contraseña.
- Desarrollar interfaces diferenciadas para cliente, diseñador y administrador.

Épica 2: Personalización de Prendas
Prioridad: Alta

Historia de usuario: Como cliente, quiero personalizar prendas según mis preferencias y medidas corporales para recibir productos únicos.

Tareas:

- Crear base de datos de plantillas y moldes estándar personalizables.
- Implementar interfaz interactiva para elección de tipos de prendas, telas, colores y estilos.
- Desarrollar sistema para ingresar medidas personalizadas y almacenarlas.
- Crear el layout básico de la vista de personalización (frontend).
- Implementar selección interactiva de tipos de prendas (menú desplegable, botones).
- Desarrollar selección de colores y materiales usando paletas visuales dinámicas.
- Desarrollar sistema de recomendación inteligente (IA) de estilos según medidas.
- Implementar persistencia temporal de selecciones del usuario usando almacenamiento local (localStorage).
- Conectar frontend con el backend mediante llamadas REST API para almacenar personalizaciones permanentemente.

Épica 3: Visualización Realista (3D)
Prioridad: Alta

Historia de usuario: Como usuario, quiero previsualizar digitalmente mis prendas personalizadas antes de su confección.

Tareas:

- Investigar e integrar biblioteca para visualizaciones en 3D (por ejemplo, Three.js, Babylon.js).

- Investigar documentación técnica y tutoriales de biblioteca elegida (Three.js o Babylon.js).
- Configurar entorno inicial básico para renderizado 3D en frontend.
- Crear modelo básico estático para prueba inicial de visualización.
- Implementar carga dinámica de modelos 3D desde backend.
- Programar interacción en tiempo real con controles básicos (zoom, rotación, desplazamiento).
- Desarrollar integración con personalizaciones elegidas por usuario (color, material, tamaño).
- Crear función para actualizar visualización inmediatamente al cambiar opciones del usuario.
- Optimizar rendimiento de carga de modelos (carga progresiva, caching).
- Realizar pruebas de rendimiento y usabilidad de la visualización 3D.
- Desarrollar función para visualización dinámica en tiempo real (actualización inmediata tras cambios).
- Implementar creación de avatares personalizados según medidas del cliente.

Épica 4: Seguimiento de Pedidos
Prioridad: Media-alta

Historia de usuario: Como cliente, deseo conocer en tiempo real el estado de mi pedido desde su creación hasta la entrega.

Tareas:

- Crear un sistema visual de tracking del estado de pedidos.
- Desarrollar notificaciones automáticas sobre cambios de estado.
- Implementar historial detallado de modificaciones en cada pedido.
- Definir estados específicos del pedido (e.g., Recibido, En diseño, En producción, Enviado, Entregado).
- Diseñar visualmente el componente gráfico para tracking en el frontend (barra de progreso visual).
- Implementar la lógica en backend para almacenar y modificar estados del pedido.
- Crear API REST para consultar y actualizar estado de pedidos desde frontend.
- Desarrollar frontend para reflejar en tiempo real cambios en los estados del pedido.

- Configurar sistema automático de notificaciones (email y/o push) al usuario al cambiar estados.
- Desarrollar módulo de historial visual para mostrar la trazabilidad completa del pedido.

Épica 5: Comunicación y Colaboración
Prioridad: Alta

Historia de usuario: Como usuario, quiero una comunicación efectiva y directa entre cliente y diseñador.

Tareas:

- Desarrollar un módulo interno para mensajería y envío de archivos adjuntos.
- Implementar chatbot inteligente para responder preguntas frecuentes y derivar a un asesor cuando sea necesario.
- Investigar soluciones tecnológicas existentes para chatbots (por ejemplo, DialogFlow).
- Crear cuenta y configurar entorno básico del chatbot en la plataforma seleccionada.
- Programar flujo básico de conversación con respuestas predefinidas (FAQ inicial).
- Implementar lógica del chatbot para derivar consultas complejas a un asesor humano.
- Integrar chatbot en el frontend del sitio (widget flotante).
- Crear módulo básico de mensajería interna entre usuarios (cliente-diseñador).
- Desarrollar backend para almacenar mensajes y archivos adjuntos en base de datos.
- Crear interfaz gráfica en frontend para intercambio de mensajes en tiempo real (tipo chat).
- Programar notificaciones instantáneas al recibir nuevos mensajes.
- Integrar sección FAQ para consultas rápidas.

Épica 6: Administración y Gestión de Plataforma
Prioridad: Media

Historia de usuario: Como administrador, deseo gestionar usuarios, contenido y realizar análisis de datos sobre el uso de la plataforma.

Tareas:

- Desarrollar panel de administración con gestión de usuarios.
- Implementar herramienta para moderación de diseños publicados.
- Crear módulo de análisis estadístico de uso (dashboard de métricas).

Épica 7: Infraestructura Técnica
Prioridad: Alta

Historia de usuario: Como desarrollador, quiero un entorno de desarrollo replicable usando contenedores para facilitar el despliegue y mantenimiento del sistema.

Tareas:

- Instalar y configurar Docker o Podman.
- Crear Dockerfile para el servicio web y base de datos.
- Preparar archivo docker-compose para orquestación de servicios.
- Configurar variables de entorno necesarias para ambos servicios.
- Probar construcción local de imágenes Docker (docker build).
- Escribir documentación técnica (README.md) con pasos de ejecución de contenedores.

Épica 8: Gestión de Pagos
Prioridad: Media

Historia de usuario: Como cliente, quiero pagar de forma segura y rápida por los pedidos realizados.

Tareas:

- Investigar plataformas de pago compatibles (PayPal, Stripe, MercadoPago).
- Implementar módulo de pagos integrando la plataforma elegida.
- Configurar entorno backend para comunicación segura con la API de pagos.
- Desarrollar módulo backend para crear transacciones y manejar callbacks/respuestas de pagos.
- Implementar validación de datos y cifrado para garantizar seguridad en transacciones.
- Asegurar cumplimiento legal (facturación SAT Guatemala).
- Documentar claramente los procesos técnicos para futuras auditorías.

Épica 9: Despliegue y Publicación del Sistema
Prioridad: Media-alta

Historia de usuario: Como usuario final, deseo acceder fácilmente a la plataforma desde cualquier dispositivo.

Tareas:

- Seleccionar proveedor de hosting en la nube.
- Desplegar aplicación usando servicios en la nube (AWS, Heroku, DigitalOcean).
- Realizar pruebas en entorno de producción.

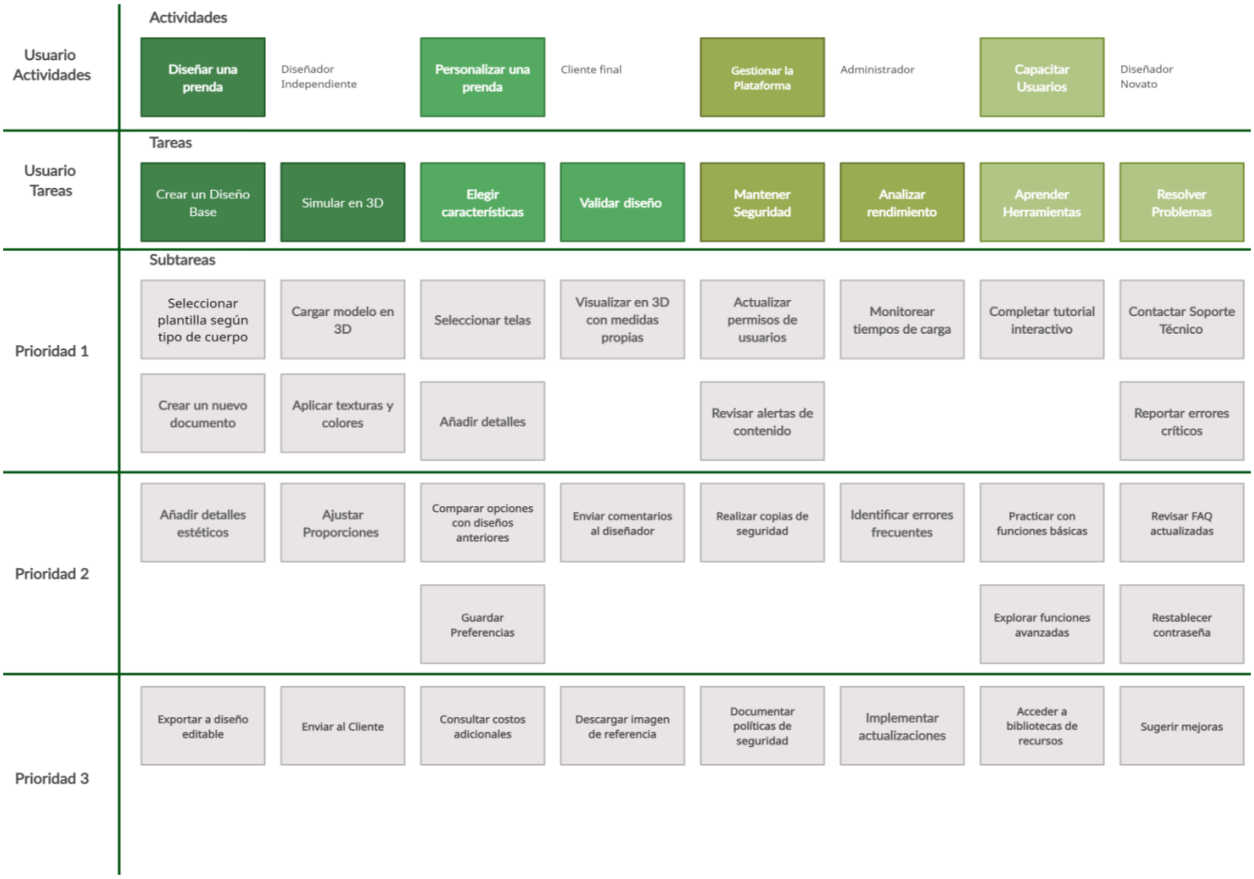
Épica 10: Mantenimiento y Escalabilidad
Prioridad: Baja-media

Historia de usuario: Como administrador del sistema, quiero garantizar que el sistema sea mantenible y escalable a largo plazo.

Tareas:

- Establecer procesos de backup automático de datos.
- Documentar claramente el código y procedimientos técnicos.
- Implementar monitoreo automático de rendimiento y alertas de caídas.

User Story Mapping



Sprint Backlog

Pila del Sprint

A continuación, se listan las tareas bajo cada historia de usuario. Se indica: nombre, descripción, horas estimadas, puntos de historia, responsable y fecha probable de finalización.

2.1. Historia de Usuario

Estimación total: 8 puntos (32h)

[SCRUMSITO-12] Formulario de registro de usuarios

Descripción: Crear y maquetar el formulario de registro, validación en frontend y conexión básica al backend.

Horas estimadas: 8

Puntos asignados (sub-tarea): 2

Responsable: Integrante 1 (Desarrollador principal)

Fecha fin (probable): 07/04/2025

[SCRUMSITO-13] Autenticación vía correo con confirmación

Descripción: Implementar envío de correo con token de confirmación y activar cuenta tras el enlace.

Horas estimadas: 6

Puntos asignados (sub-tarea): 2

Responsable: Integrante 1

Fecha fin (probable): 08/04/2025

[SCRUMSITO-14] Autenticación vía redes sociales

Descripción: Permitir registro/inicio de sesión con proveedores externos (Google, Facebook, etc.).

Horas estimadas: 6

Puntos asignados (sub-tarea): 2

Responsable: Integrante 1

Fecha fin (probable): 09/04/2025

[SCRUMSITO-15] Recuperación de contraseña

Descripción: Proporcionar un flujo de “olvido de contraseña”, envío de link y cambio seguro.

Horas estimadas: 4

Puntos asignados (sub-tarea): 1

Responsable: Integrante 1

Fecha fin (probable): 10/04/2025

[SCRUMSITO-16] Interfaces diferenciadas para cliente y diseñador

Descripción: Ajustar la interfaz según rol de usuario (cliente o diseñador), mostrando secciones adecuadas.

Horas estimadas: 8

Puntos asignados (sub-tarea): 1 (complemento funcional de la historia 45)

Responsable: Integrante 4 (apoya front-end siguiendo lineamientos de Integrante 1)

Fecha fin (probable): 10/04/2025

2.2. Historia de Usuario [SCRUMSITO-50]

Estimación total: 5 puntos (20h)

[SCRUMSITO-33] Panel de administración con gestión de usuarios

Descripción: Implementar vista y lógica para que el administrador pueda crear, editar, eliminar y listar usuarios.

Horas estimadas: 8

Puntos asignados (sub-tarea): 2

Responsable: Integrante 1

Fecha fin (probable): 11/04/2025

[SCRUMSITO-34] Herramienta para moderación de diseños publicados

Descripción: Permitir al administrador ver, aprobar o rechazar diseños subidos por diseñadores.

Horas estimadas: 8

Puntos asignados (sub-tarea): 2

Responsable: Integrante 1

Fecha fin (probable): 12/04/2025

[SCRUMSITO-16] Interfaz diferenciada para administrador

Descripción: Ajustar la interfaz de administrador, agregando menús y secciones que no vean otros roles.

Horas estimadas: 4

Puntos asignados (sub-tarea): 1

Responsable: Integrante 4

Fecha fin (probable): 12/04/2025

2.3. Historia de Usuario [SCRUMSITO-35]

Estimación total: 3 puntos (12h)

[SCRUMSITO-35] Módulos de análisis estadístico de uso

Descripción: Desarrollar módulo para recolectar, procesar y mostrar estadísticas de uso (inicios de sesión, frecuencia, etc.).

Horas estimadas: 12

Puntos asignados (sub-tarea): 3 (es la historia completa)

Responsable: Integrante 4

Fecha fin (probable): 14/04/2025

2.4. Tareas de Soporte / Infraestructura

Estas tareas no se formulan como historias de usuario para el cliente, pero se requieren dentro del sprint para lograr los objetivos del proyecto.

[SCRUMSITO-5] Elección Contenedor

Descripción: Investigar y definir la tecnología de contenedores (Docker, Podman, etc.) a utilizar.

Horas estimadas: 4

Responsable: Integrante 3 (investigación)

Fecha fin (probable): 05/04/2025

[SCRUMSITO-6] Herramienta Contenerización

Descripción: Configurar Dockerfile / Compose u otra herramienta según la elección previa.

Horas estimadas: 6

Responsable: Integrante 1 (implementación técnica)

Fecha fin (probable): 07/04/2025

[SCRUMSITO-7] Creación de Archivos Configuración

Descripción: Generar archivos YAML/JSON/ENV para la correcta configuración de la aplicación y contenedores.

Horas estimadas: 4

Responsable: Integrante 1

Fecha fin (probable): 07/04/2025

[SCRUMSITO-8] README con los comandos necesarios

Descripción: Redactar README para desarrolladores con las instrucciones de instalación y ejecución en contenedores.

Horas estimadas: 3

Responsable: Integrante 3

Fecha fin (probable): 08/04/2025

[SCRUMSITO-20] Pruebas de contenerización

Descripción: Levantar ambiente de pruebas en contenedor y validar que la aplicación corra correctamente.

Horas estimadas: 4

Responsable: Integrante 3

Fecha fin (probable): 09/04/2025

2.5. Tareas de Gestión y Documentación

[SCRUMSITO-3] Lista de Historias de Usuario

Descripción: Mantener actualizada la lista de historias en Jira, verificar que cumplan con formato INVEST.

Horas estimadas: 2

Responsable: Integrante 2 (Scrum Master)

Fecha fin (probable): 04/04/2025

[SCRUMSITO-4] Calendario

Descripción: Configurar en Jira / Calendar las fechas de las reuniones diarias, revisiones, retrospectivas y demos.

Horas estimadas: 2

Responsable: Integrante 2

Fecha fin (probable): 04/04/2025

[SCRUMSITO-9] Resultados Sprint

Descripción: Capturar conclusiones del sprint (al final), redactar logros y elementos pendientes.

Horas estimadas: 2

Responsable: Integrante 2

Fecha fin (probable): 14/04/2025

[SCRUMSITO-10] Métricas Sprint

Descripción: Recopilar métricas de velocidad, puntos completados y analizar el rendimiento del equipo.

Horas estimadas: 2

Responsable: Integrante 2

Fecha fin (probable): 14/04/2025

[SCRUMSITO-11] Documentación

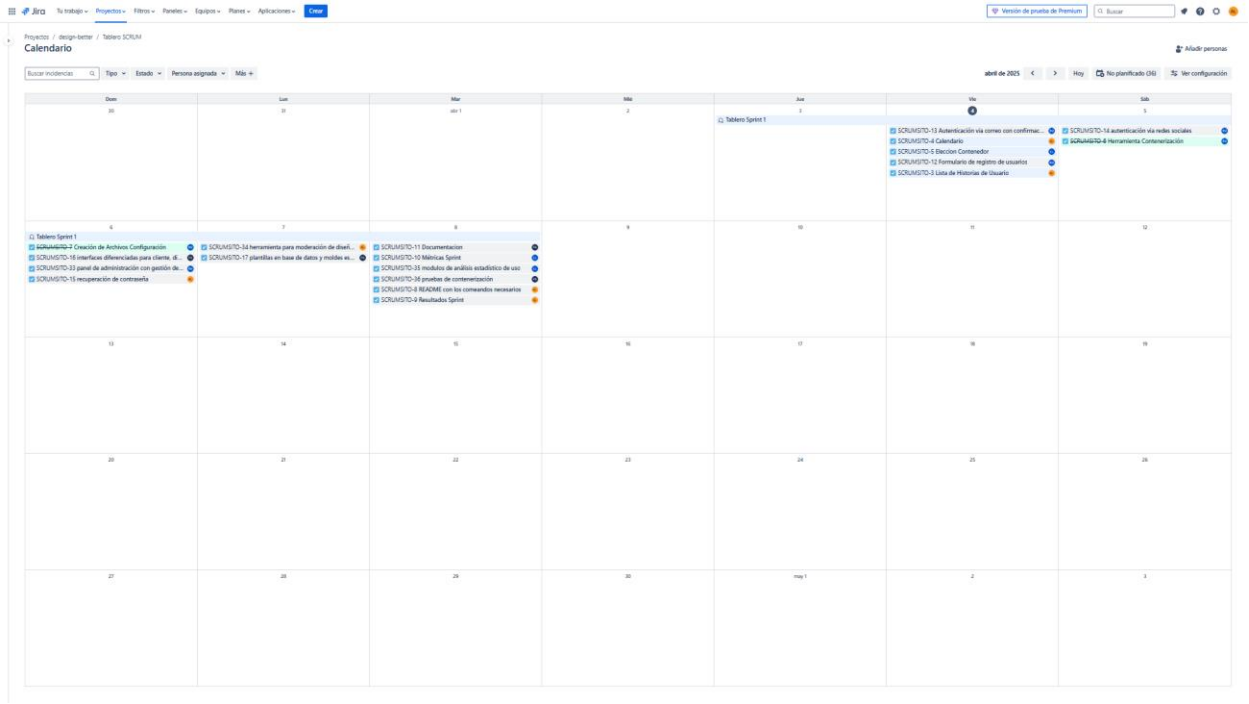
Descripción: Generar documentación básica del proyecto (arquitectura, manual de usuario/desarrollador).

Horas estimadas: 5

Responsable: Integrante 3 (con apoyo de Integrante 4)

Fecha fin (probable): 10/04/2025

Calendario



Elección de Contenedor

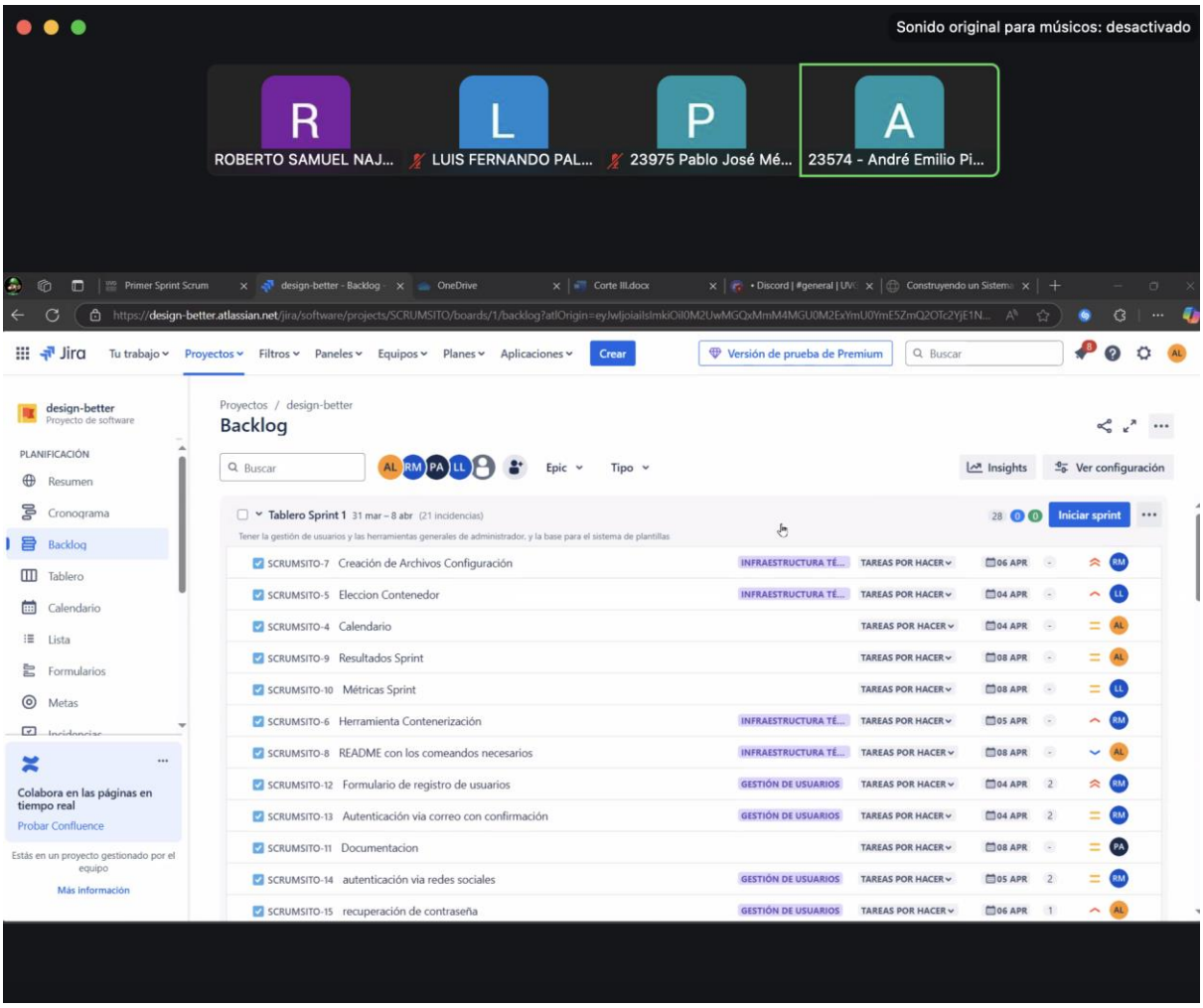
Justificación Detallada:

1. Compatibilidad con las Tecnologías Seleccionadas:
    a. Frontend (React.js + Next.js): Docker cuenta con imágenes oficiales de Node.js y soporte nativo para aplicaciones Next.js, facilitando la configuración de entornos de desarrollo y producción.
    b. Backend (Django + Python): Las imágenes oficiales de Python en Docker Hub están optimizadas para frameworks como Django, permitiendo una integración fluida con bases de datos y dependencias.
    c. Base de Datos (PostgreSQL): Docker ofrece imágenes oficiales de PostgreSQL, simplificando la configuración, escalabilidad y persistencia de datos.
    d. Herramientas Adicionales: Integración con servicios complementarios (Redis, Nginx, etc.) mediante Docker Compose.
2. Ecosistema y Comunidad:
    a. Docker es el estándar de la industria con una comunidad activa, documentación extensa y soluciones probadas para problemas comunes. Esto reduce riesgos y acelera la resolución de incidencias.
    b. Amplio soporte en plataformas en la nube (AWS, Azure, Google Cloud) y herramientas de CI/CD (GitHub Actions, GitLab CI), alineándose con los requisitos de escalabilidad y portabilidad.
3. Herramientas de Orquestación:
    a. Docker Compose: Ideal para entornos locales y despliegues sencillos, permite definir y gestionar múltiples servicios (frontend, backend, base de datos) en un solo archivo `docker-compose.yml`.
    b. Integración con Kubernetes: Si el proyecto escala, Docker facilita la migración a clusters Kubernetes gracias a su compatibilidad nativa, cumpliendo con el requisito de mantenibilidad.
4. Mantenibilidad y Reproducibilidad:
    a. Los contenedores Docker garantizan entornos consistentes entre desarrollo, testing y producción, eliminando el problema de *"funciona en mi máquina"*.
    b. Permite versionar imágenes y configuraciones, facilitando actualizaciones y rollbacks.
5. Ventajas frente a Alternativas (Podman):
    a. Madurez: Docker tiene una trayectoria más larga y adopción masiva en proyectos empresariales, lo que garantiza estabilidad.
    b. Herramientas Auxiliares: Docker Desktop ofrece una interfaz gráfica intuitiva para usuarios menos técnicos, mientras que Podman requiere mayor expertise en CLI.
    c. Integración con Ecosistemas: Docker se integra mejor con servicios de monitorización (Prometheus, Grafana) y seguridad (Trivy, Clair), críticos para cumplir requisitos no funcionales de confiabilidad y seguridad.
6. Alineación con los Requisitos del Proyecto:
    a. Rendimiento: La optimización de imágenes Docker (mediante multi-stage builds) reduce el tamaño final, acelerando despliegues.

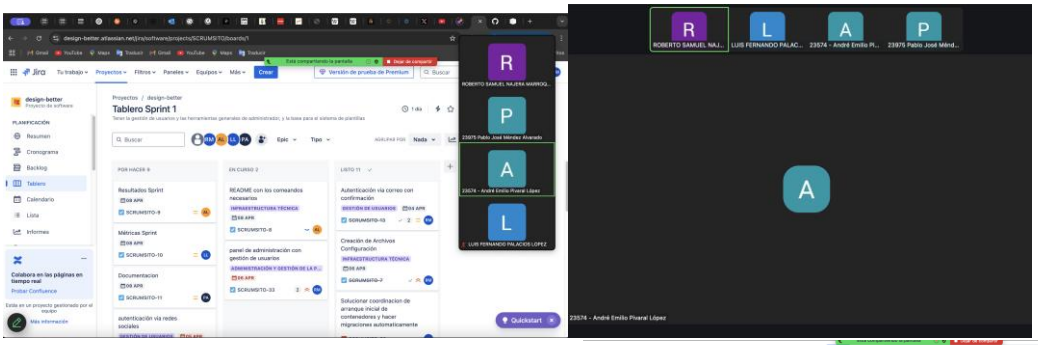
b. Cumplimiento Legal: Las imágenes oficiales de Docker incluyen licencias validadas, mitigando riesgos legales.
c. Portabilidad: Los contenedores se ejecutan en cualquier sistema operativo (Windows, macOS, Linux), cumpliendo el requisito de portabilidad.

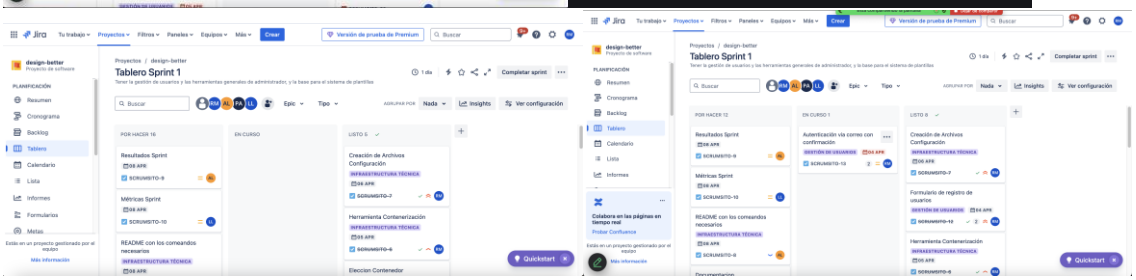
Evidencia del uso de las técnicas de scrum.

Reunión inicial de planificación del Sprint 1:



Reuniones diarias de scrum (viruales por zoom):





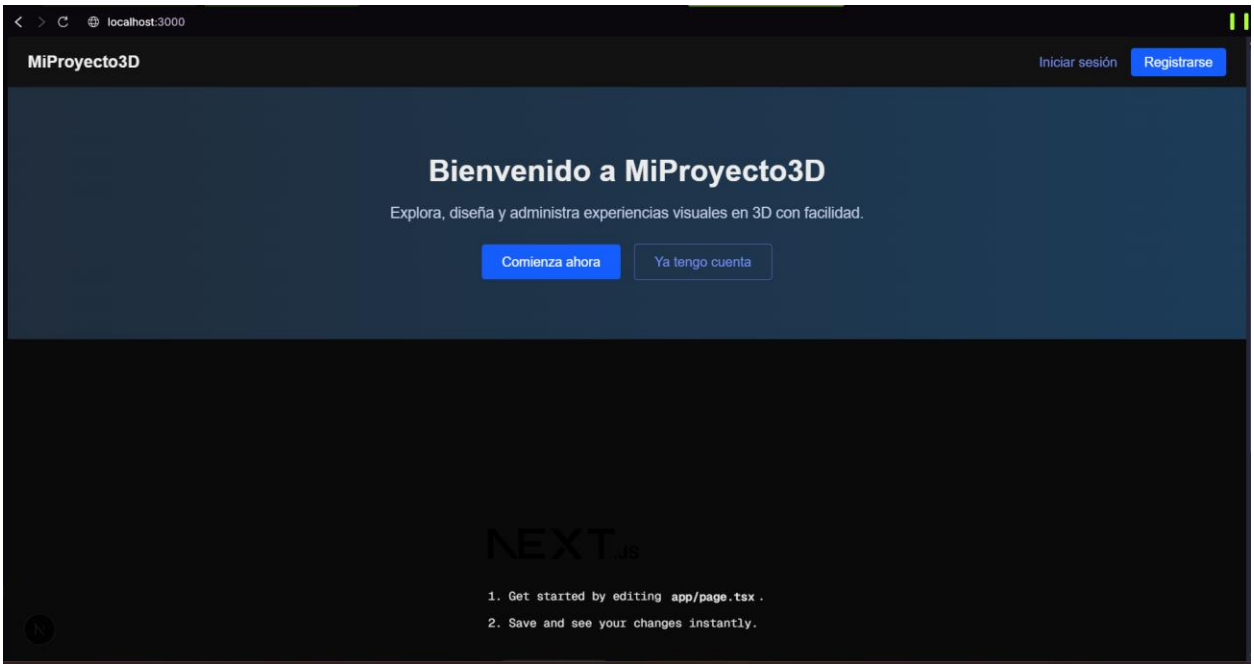
Reunión final de Sprint:

Enlace de Jira: https://design-better.atlassian.net/jira/software/projects/SCRUMSITO/boards/1?atlOrigin=eyJpIjoiNjY0OGUzYjk5NzU0NGQ5YmFhMzk5NTZiYjZhMDVkMTAiLCJwIjoiaiJ9

Carpeta de grabaciones: [Evidencias Scrum]

Resultados del Sprint

Software Funcionando



Tareas Concluidas

Tipo	Clave	Resumen	Persona asignada	Informador	Prio
	~~SCRUMSITO-5~~	Solucionar coordinacion de arranque inicial de contenedores y hacer migraciones automaticamente	LUIS FERNANDO...	ROBERTO SAMU...	Medium
	~~SCRUMSITO-3~~	pruebas de contenerización	PABLO JOSE ME...	ROBERTO SAMU...	Medium
	~~SCRUMSITO-1~~	interfaces diferenciadas para cliente, diseñador y administrador	PABLO JOSE ME...	ROBERTO SAMU...	Medium
	~~SCRUMSITO-1~~	recuperación de contraseña	ANDRE EMILIO P...	ROBERTO SAMU...	High
	~~SCRUMSITO-1~~	Autenticación via correo con confirmación	ROBERTO SAMU...	ROBERTO SAMU...	Medium
	~~SCRUMSITO-1~~	Formulario de registro de usuarios	ROBERTO SAMU...	ROBERTO SAMU...	Highest
	~~SCRUMSITO-1~~	Documentacion	PABLO JOSE ME...	ANDRE EMILIO P...	Medium
	~~SCRUMSITO-1~~	Métricas Sprint	LUIS FERNANDO...	ANDRE EMILIO P...	Medium
	~~SCRUMSITO-9~~	Resultados Sprint	ANDRE EMILIO P...	ANDRE EMILIO P...	Medium
	~~SCRUMSITO-8~~	README con los comeandos necesarios	ANDRE EMILIO P...	ANDRE EMILIO P...	Low
	~~SCRUMSITO-7~~	Creación de Archivos Configuración	ROBERTO SAMU...	ANDRE EMILIO P...	Highest
	~~SCRUMSITO-6~~	Herramienta Contenerización	ROBERTO SAMU...	ANDRE EMILIO P...	High
	~~SCRUMSITO-5~~	Eleccion Contenedor	LUIS FERNANDO...	ANDRE EMILIO P...	High
	~~SCRUMSITO-4~~	Calendario	ANDRE EMILIO P...	ANDRE EMILIO P...	Medium
	~~SCRUMSITO-3~~	Lista de Historias de Usuario	ANDRE EMILIO P...	ANDRE EMILIO P...	Medium

Tareas planificadas, pero no concluidas

Repositorios con el código desarrollado:
https://github.com/Ultimate-Truth-Seeker/DesignBetterFrontend

https://github.com/Ultimate-Truth-Seeker/DesignBetterBackend

Métricas del Sprint

Gráfico de Sprint Burndown de puntos de historia

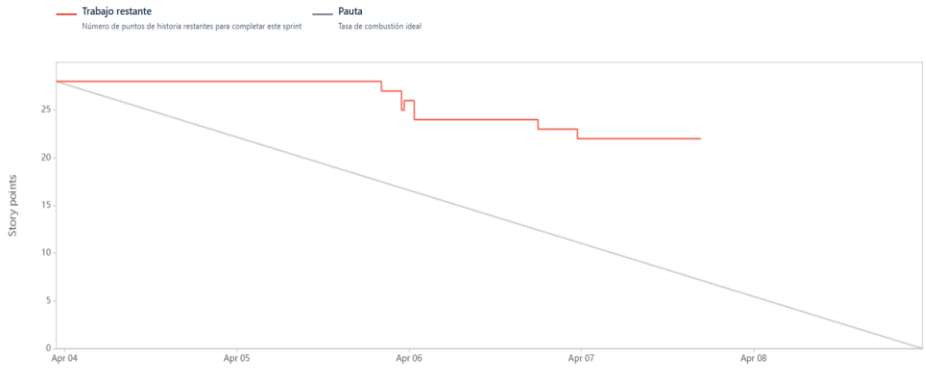
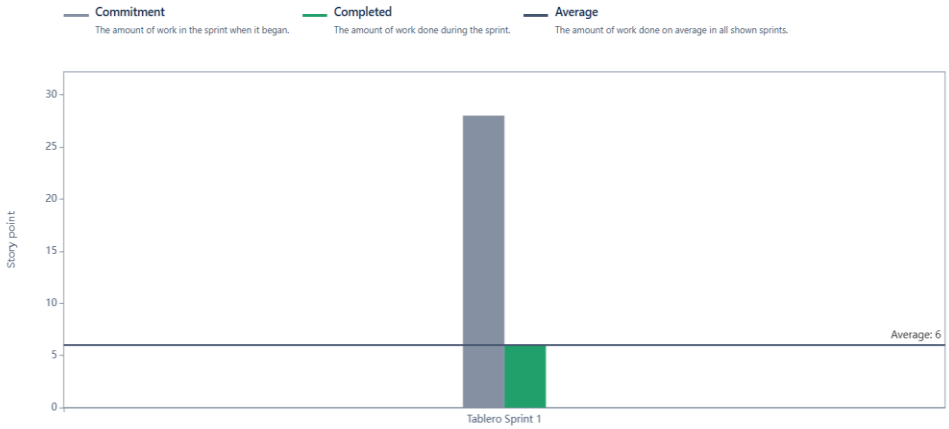


Gráfico de Sprint Burndown de Recuento de Incidencias

Reporte de Velocidad



Sprint	Commitment	Completed
Tablero Sprint 1	28	6

Éxito del Sprint

Para evaluar de manera objetiva el rendimiento del equipo durante el Sprint I, se utilizó un indicador de éxito basado en los puntos de historia comprometidos y completados.

Ecuación

Indicador de Éxito = (Puntos completados / Puntos comprometidos) × 100

= (19 / 28) × 100 ≈ 67.8%

- Puntos comprometidos al inicio del sprint: 28
- Puntos efectivamente completados: 19

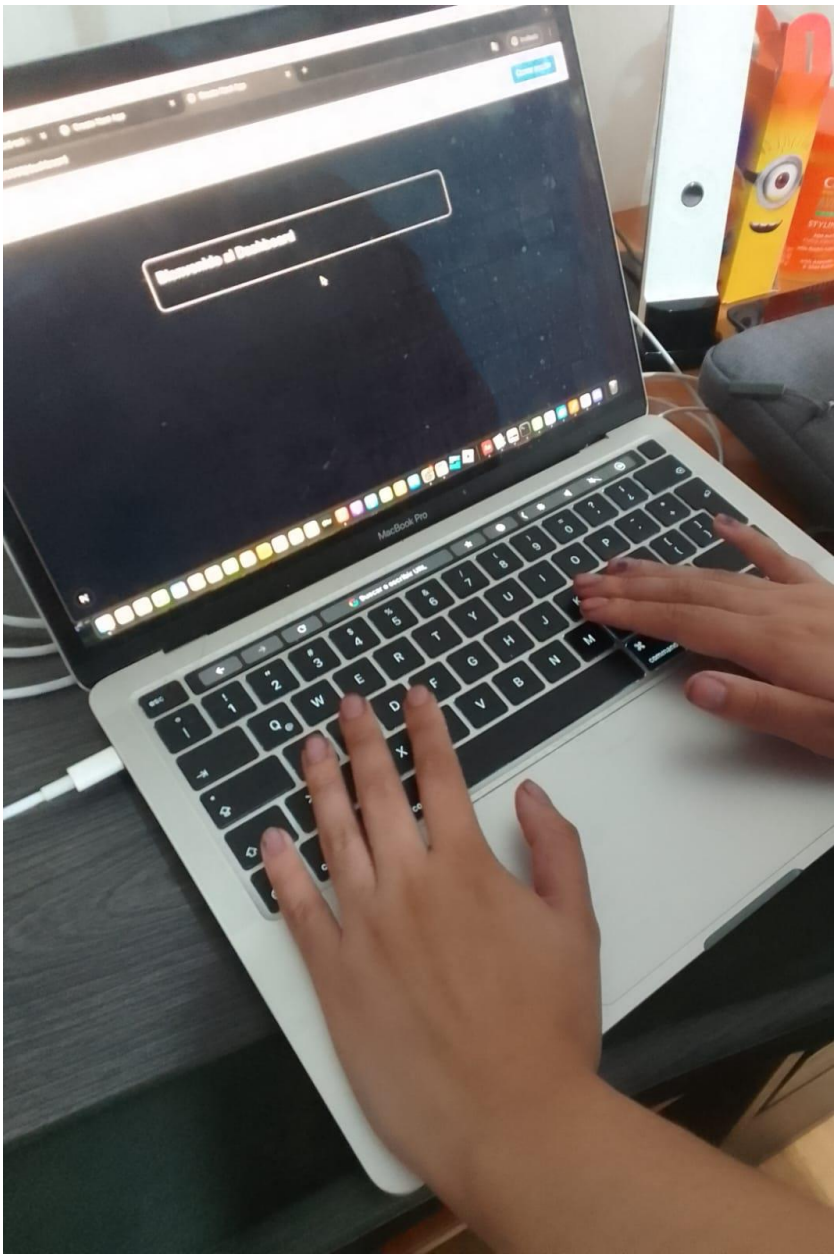
Basándose en este indicador de éxito, la calificación del Sprint sería de 7 de 10. El principal problema observado durante este primer Sprint fue la inexperiencia de todos los miembros del equipo para las herramientas utilizadas.

Iniciando en el ámbito de la contenerización, ya que nadie había usado antes Docker y adaptarse a esta herramienta sigue siendo desafiante incluso con el Sprint ya finalizado. Sumado a ello, de igual forma nadie tenía conocimiento acerca del uso de Jira, y aprender a usarlo resto días que se pudieron haber utilizado para las tareas que más tarde fueron asignadas.

Otro matiz que se podría considerar para un resultado no óptimo fue el hecho de asignar muchas tareas por cada punto de historia, por lo que en la Gráfica de Burndown de puntos de historia pareciese que no se alcanzó un buen porcentaje de tareas completas frente a las que se tenía esperado completar, sin embargo, esto se pudo evitar agregando menos tareas por punto de historia, lo cual habría mejorado no solo el gráfico sino la efectividad del Sprint.

Finalmente, un aspecto que afectó al rendimiento de este Sprint fue la subestimación de algunas tareas, ya que estas en muchos casos resultaron siendo más desafiantes y tardadas de lo que se tenía previsto, y esto afectó a que los tiempos planificados resultaran siendo imprecisos.

Evidencia de Pruebas con Usuarios finales



Usuario de prueba: diseñadora Gaby Nájera

Fecha: 8 de abril a las 20:30

Resumen de comentarios:

- Cuando se hizo el registro, preguntó si sí era seguro y si iba a recibir algo en su correo
- Preguntó si el enlace de activación funcionaba y le expliqué que de momento solo en la computadora de pruebas.
- Muy bien, bastante básico y hay que mejorar diseño gráfico.

Reflexión del Desempeño del Equipo en el Sprint I

El equipo demostró un desempeño colaborativo y comprometido, logrando completar una buena parte de los puntos de historia comprometidos. Tuvimos estos aspectos positivos. Cada integrante cumplió con sus responsabilidades asignadas, destacándose en la implementación de funcionalidades críticas como el formulario de registro, autenticación básica y configuración inicial de contenedores. Además, se mantuvo una comunicación constante para resolver incidencias, como la integración de Docker, lo que refleja adaptabilidad ante desafíos técnicos.

Aspectos Clave Para Mejorar para el Próximo Sprint:
Estimación Realista de Tareas:

La subestimación de tiempos en tareas técnicas afectó el avance. Se propone realizar sesiones de investigación técnica previa al sprint para definir requisitos y complejidades, mejorando la precisión en las estimaciones.

Distribución Equitativa de Cargas:

Algunos integrantes asumieron múltiples tareas críticas simultáneamente, generando cuellos de botella. Para evitarlo, se redistribuirán responsabilidades según especialidades y se asignarán tareas de soporte en pares para equilibrar la carga.

Capacitación en Herramientas:

La curva de aprendizaje en Docker y Jira consumió tiempo valioso. Se incorporarán micro-talleres prácticos al inicio del sprint para familiarizar al equipo con herramientas esenciales, acelerando la productividad.

Gestión de Historias de Usuario:

La acumulación excesiva de sub-tareas por historia dificultó el seguimiento del progreso. Se adoptará el principio INVEST (Independent, Negotiable, Valuable, Estimable, Small, Testable) para desglosar historias en tareas más manejables y priorizables.

Monitoreo Proactivo de Bloqueos:

Las reuniones diarias se enfocarán en identificar riesgos tempranos (ej. dependencias técnicas) y ajustar el plan en tiempo real, utilizando métricas como el Burndown Chart para visualizar desviaciones y actuar oportunamente.

En conclusión, el equipo ha sentado una base sólida de colaboración, pero optimizará su enfoque mediante una planificación más detallada, capacitación estratégica y una distribución balanceada de tareas. Esto permitirá elevar la eficiencia y cumplir metas con mayor precisión en los próximos sprints.

Informe general de gestión para el Sprint 1:

Fecha de Inicio: miércoles 2 de abril de 2025

Fecha de finalización: martes 8 de abril de 2025

Descripción de tareas propuestas y completadas: desglosadas en el sprint

Los formularios LOGT para el mismo se añadieron dentro de la carpeta específica dentro del repositorio de entrega:

https://github.com/Ultimate-Truth-Seeker/ProyectoIS

Los vínculos de los documentos de entregas anteriores se pueden consultar viendo el historial de commits para el README.md