

UNIVERSIDAD DEL VALLE DE GUATEMALA

Ingeniería de Software 1, sección 30



Excelencia que trasciende

DEL VALLE
GRUPO EDUCATIVO

PRIMER SPRINT

Sofia Garcia - 22210

Julio Garcia Salas - 22076

Joaquin Campos - 22155

Juan Fernando Menéndez - 17444

Juan Pablo Solis - 22102

GUATEMALA, marzo de 2024

Índice

Índice.....	2
Resumen.....	2
Introducción.....	2
Planificación del Sprint.....	2
Product Backlog.....	3
Pila del sprint.....	11
Resultados del sprint.....	13
Métricas del sprint.....	14
Discusión.....	14
Código desarrollado.....	15
Documentación.....	15

Resumen

Este documento detalla el desarrollo y resultados del primer sprint del proyecto de Ingeniería de Software 1. El objetivo principal fue establecer una base sólida para el proyecto, enfocándose en la configuración del entorno de desarrollo, la selección e instalación de tecnologías clave como frameworks de backend y frontend, configuración de Docker y gestor de base de datos, y el desarrollo de una API inicial y una página de inicio estática. A través de una meticulosa planificación y ejecución de tareas, el equipo logró completar todas las actividades programadas, sentando así las bases para los sprints futuros. Este sprint no solo sirvió para avanzar técnicamente en el proyecto sino también para evaluar la capacidad de trabajo en equipo, la gestión del tiempo, y la distribución de tareas, ofreciendo valiosas lecciones para los próximos sprints.

Introducción

El presente documento es una compilación detallada del trabajo del Sprint 1. Durante este sprint, el equipo se enfocó en configurar y preparar el entorno de desarrollo necesario para el proyecto, además de iniciar el desarrollo con la creación de componentes fundamentales tanto en el backend como en el frontend. El proceso se llevó a cabo siguiendo metodologías ágiles, específicamente Scrum, lo que permitió una gestión eficiente del proyecto y una adaptación flexible a los requisitos y desafíos que surgieron. Este documento cubre desde la planificación inicial del sprint, pasando por la definición del product backlog, la ejecución y resultados de las tareas del sprint, hasta la evaluación de las métricas del sprint, ofreciendo una visión completa de los logros y aprendizajes del equipo durante este periodo inicial del proyecto.

Planificación del Sprint

Sprint Planning			
Importancia baja		Importancia media	Importancia alta
Sprint 1	Documentar la API inicial Inicio: 1 de abril Fin: 2 de abril	Crear repositorio Git y establecer flujo de trabajo Inicio: 21 de marzo Fin: 21 de marzo	Configurar entorno de desarrollo local Inicio: 18 de marzo Fin: 18 de marzo
	Crear estructura de proyecto y organizar archivos Inicio: 20 de marzo Fin: 20 de marzo	Implementar página de inicio estática Inicio: 25 de marzo Fin: 25 de marzo	Instalar framework backend (Vite y Express JS) Inicio: 18 de marzo Fin: 19 de marzo
		Configurar conexión a la base de datos en un contenedor Docker Inicio: 27 de marzo Fin: 27 de marzo	Instalar framework frontend (React) Inicio: 18 de marzo Fin: 19 de marzo
			Configurar gestor de base de datos (PostgreSQL) Inicio: 25 de marzo Fin: 25 de marzo
			Configurar Docker para el entorno de desarrollo backend Inicio: 25 de marzo Fin: 26 de marzo
			Crear endpoints básicos de API para productos Inicio: 25 de marzo Fin: 29 de marzo

User Story Mapping:

https://miro.com/welcomeonboard/dEMweWZKNU1qS00xSFAzN0JiWGg3V1hNUkVTYnA2SnA1dE5GTmE2Y3E4SHZodEE5T1RWa1czU0RrTHJsN3IMS3wzNDU4NzY0NTc5NzMwMDMwOTM1fDI=?share_link_id=423128447307

Product Backlog

ID	Descripción	Imp	Est	Demo	Notas
1	Configurar entorno de desarrollo local	Alta	3	Mostrar IDE configurado con todas las herramientas necesarias.	Incluir Git, editor de código, y entornos de servidor local.
2	Instalar framework backend (Vite y Express JS)	Alta	2	Crear un proyecto básico y ejecutarlo localmente.	Seleccionar Vite y Express JS por su estructura MVC y facilidad de uso.
3	Instalar framework frontend (React)	Alta	2	Configurar React y mostrar una página de prueba.	React es elegido por su simplicidad y documentación amigable.
4	Configurar gestor de base de datos (PostgreSQL)	Alta	3	Conexión a base de datos desde el backend.	Demostrar conexión exitosa y creación de tablas iniciales.
5	Crear repositorio Git y establecer	Media	2	Repositorio creado y primer commit.	Mostrar el repositorio en

	flujo de trabajo				GitHub.
6	Implementar página de inicio estática	Media	5	Página de inicio accesible localmente.	Incluir elementos SEO básicos como meta tags (para la página web).
7	Crear estructura de proyecto y organizar archivos	Alta	2	Estructura de carpetas mostrada en la demo.	Seguir las mejores prácticas de estructuración de proyectos.
8	Configurar Docker para el entorno de desarrollo backend	Alta	3	Mostrar el entorno de desarrollo Docker funcionando.	Incluir Dockerfile y docker-compose.yml para el backend y la base de datos.
9	Configurar conexión a la base de datos en un contenedor Docker	Alta	2	Mostrar una conexión exitosa a la base de datos en Docker.	Utilizar docker-compose para levantar el servicio de base de datos.
10	Crear endpoints básicos de API para productos	Alta	5	Demostrar CRUD de productos vía API.	Incluir endpoints para listar, crear, actualizar y eliminar productos.
11	Documentar la API inicial	Media	2	Presentar documentación básica de la API.	Considerar herramientas compatibles con Docker para la documentación.
12	Investigación de palabras clave SEO	Alta	4	Lista de palabras clave objetivo.	Usar herramientas como SEMrush o Google Keyword Planner.
13	Optimización de meta tags para SEO	Alta	4	Página de inicio con meta tags optimizados.	Incluir tags como title, description y keywords.
14	Configuración de Google Analytics	Alta	2	Integración de Google Analytics verificada.	Mostrar datos de tráfico en el dashboard de Analytics.
15	Configuración de	Alta	2	Sitio verificado en Google	Configurar

	Google Search Console			Search Console.	sitemap y verificar indexación.
16	Diseño de la arquitectura backend	Alta	4	Diagrama de arquitectura del backend.	Definir estructura de carpetas y módulos principales.
17	Configuración inicial de Vite para backend	Alta	2	Proyecto Vite corriendo con Express JS.	Mostrar servidor Express funcionando con Vite.
18	Creación de modelos de datos con Sequelize	Alta	4	Modelos para Producto, Categoría y Orden implementados.	Usar Sequelize como ORM para PostgreSQL.
19	Implementación de routing en React	Alta	3	Navegación entre páginas de inicio y gestión de inventarios.	Usar React Router para gestionar rutas.
20	Implementación de la UI en React	Alta	8	Componentes React para gestión de inventarios.	Convertir los mockups de Figma a componentes React.
21	Creación de la base de datos en PostgreSQL	Alta	2	Base de datos creada y configurada en PostgreSQL.	Ejecutar scripts SQL para crear la base de datos y tablas.
22	Diseñar sistema de navegación intuitivo	Alta	4	Sistema de navegación implementado.	Crear un menú lateral o superior para acceder fácilmente a todas las secciones de la app.
23	Implementar diseño responsivo	Alta	6	Demostración de la app en diferentes dispositivos.	Asegurar que la web app sea completamente usable en móviles, tablets y desktops.
24	Crear componentes UI reutilizables	Alta	8	Biblioteca de componentes UI.	Desarrollar botones, formularios, tarjetas, y otros elementos de UI en React para reutilización.

25	Desarrollar pantalla de gestión de inventarios	Alta	8	Pantalla de gestión de inventarios funcional.	Permitir a los usuarios ver, añadir, editar y eliminar inventarios fácilmente.
26	Implementar tablas dinámicas para listado de productos	Alta	6	Tablas dinámicas en la sección de productos.	Usar una librería como React Table para mostrar productos con opciones de filtrado y paginación.
27	Diseñar e implementar dashboards de análisis	Alta	8	Dashboards de análisis para seguimiento de inventario.	Mostrar estadísticas clave, gráficos de ventas, y tendencias de inventario.
28	Integrar sistema de búsqueda avanzada	Alta	6	Funcionalidad de búsqueda avanzada.	Permitir a los usuarios buscar productos por diferentes criterios.
29	Desarrollar sistema de notificaciones	Media	4	Sistema de notificaciones en la app.	Notificar a los usuarios sobre cambios importantes, como bajos niveles de inventario.
30	Crear pantallas de carga y feedback al usuario	Media	4	Implementación de feedback visual.	Mostrar indicadores de carga y mensajes de éxito/error tras acciones del usuario.
31	Implementar modales para creación/edición de productos	Alta	6	Modales para añadir y editar productos.	Usar ventanas modales para formularios de productos, mejorando la interacción sin recargar la página.
32	Personalizar temas	Media	5	Temas personalizados	Permitir a los

	y estilos de la app			aplicados.	usuarios elegir entre temas claros y oscuros para mejorar la experiencia visual.
33	Optimización de la interfaz para accesibilidad	Alta	6	Mejoras en accesibilidad verificadas.	Asegurar cumplimiento con estándares WCAG, incluyendo contraste de colores, navegación por teclado y etiquetas ARIA.
34	Pruebas de usabilidad con usuarios reales	Alta	8	Reporte de pruebas de usabilidad.	Recoger feedback de usuarios para iterar sobre el diseño y la funcionalidad de la UI.
35	Ajustes de UI basados en feedback de pruebas	Media	6	Implementación de ajustes en la UI.	Realizar mejoras en la interfaz basadas en los resultados de las pruebas de usabilidad.
36	Desarrollo de ayuda contextual y tooltips	Baja	4	Ayuda y tooltips implementados.	Proporcionar información útil y consejos sobre cómo usar diferentes partes de la app.
37	Implementación de endpoints para CRUD de productos	Alta	6	API endpoints para productos funcionando.	Crear, leer, actualizar, eliminar productos a través de la API.
38	Pruebas unitarias para modelos de datos	Alta	4	Pruebas unitarias para modelos pasando.	Usar Jest o Mocha para pruebas de modelos Sequelize.
39	Pruebas unitarias para endpoints de	Alta	4	Pruebas unitarias para API de productos	Asegurar que los endpoints retornan

	productos			pasando.	los datos esperados.
40	Configuración de seguridad básica en Express	Alta	3	Middleware de seguridad implementado.	Usar helmet y CORS en el servidor Express.
41	Implementación de autenticación JWT	Alta	4	Sistema de autenticación con JWT funcionando.	Permitir login y proteger rutas de la API con JWT.
42	Documentación de la API con Swagger	Media	3	Documentación de la API disponible.	Usar Swagger para documentar los endpoints de la API.
43	Configuración inicial de SSL con Let's Encrypt	Media	3	HTTPS configurado para el sitio web.	Demostrar el sitio funcionando sobre HTTPS.
44	Optimización inicial de imágenes para la web	Media	3	Imágenes optimizadas en la página de inicio.	Usar herramientas como ImageOptim o servicios en línea.
45	Implementación de sistema de logging	Media	3	Logs de aplicación y errores configurados.	Usar Winston o Morgan para logging en el backend.
46	Preparación de entorno de pruebas para frontend	Media	3	Configuración de Jest para pruebas de React.	Mostrar pruebas básicas de componentes React.
47	Preparación y ejecución de pruebas de carga	Baja	4	Pruebas de carga ejecutadas en la API.	Usar Artillery o JMeter para simular tráfico.
48	Revisión de accesibilidad web	Baja	3	Auditoría de accesibilidad completada.	Usar herramientas como Lighthouse para evaluar accesibilidad.
49	Configuración de CI/CD con GitHub Actions	Media	4	Pipeline de CI/CD funcionando para pruebas y despliegue.	Automatizar pruebas y despliegue con GitHub Actions.
50	Creación de contenido SEO	Alta	4	Contenido optimizado para SEO implementado.	Escribir y optimizar el

	para la página de inicio				contenido de la página de inicio.
51	Análisis y mejora de la velocidad de carga de la página	Alta	4	Mejoras en la velocidad de carga demostradas.	Implementar técnicas de optimización como lazy loading.
52	Integración de pasarela de pago	Alta	8	Demostrar transacciones de prueba.	Integrar Stripe o PayPal para manejar pagos.
53	Desarrollo de funcionalidad de carrito de compras	Alta	6	Carrito de compras funcional en la web app.	Permitir a los usuarios agregar y modificar productos en el carrito.
54	Implementación de filtros de búsqueda para productos	Alta	5	Filtros de búsqueda funcionando.	Permitir a los usuarios filtrar productos por categoría, precio, etc.
55	Optimización de la base de datos	Alta	4	Mejoras en el rendimiento de consultas demostradas.	Indexar columnas clave y revisar consultas para optimización.
56	Implementación de respuestas caché en la API	Media	3	Caché de respuestas implementado para endpoints clave.	Usar Redis o memoria caché similar para mejorar la velocidad de respuesta.
57	Desarrollo de la sección de administración de inventarios	Alta	8	Sección de administración completa y funcional.	Permitir a los administradores gestionar inventarios directamente desde la web app.
58	Pruebas de seguridad de la aplicación	Alta	6	Reporte de pruebas de seguridad.	Realizar pruebas de penetración y corregir vulnerabilidades encontradas.
59	Configuración de backups	Alta	3	Sistema de backups configurado y probado.	Asegurar que los backups de la base

	automáticos para la base de datos				de datos se realicen diariamente.
60	Preparativos finales para el lanzamiento	Alta	5	Checklist de lanzamiento completado.	Revisar SEO, seguridad, rendimiento y funcionalidades clave antes del lanzamiento.
61	Realización de pruebas A/B para la página de inicio	Media	4	Resultados de pruebas A/B implementados.	Usar herramientas como Google Optimize para realizar pruebas A/B.
62	Implementación de funcionalidades de reportes y análisis	Alta	6	Sistema de reportes implementado.	Desarrollar dashboards para seguimiento de ventas, inventario y análisis de usuarios.
63	Optimización final de rendimiento del frontend	Alta	5	Mejoras en el rendimiento del frontend demostradas.	Aplicar técnicas como code splitting, lazy loading y optimización de imágenes.
64	Revisión final de accesibilidad web	Media	3	Auditoría de accesibilidad completada y mejoras aplicadas.	Asegurar cumplimiento con WCAG y mejorar la accesibilidad general.
65	Preparación y ejecución de plan de lanzamiento	Alta	4	Plan de lanzamiento ejecutado.	Coordinar comunicados de prensa, publicaciones en redes sociales y campañas de email marketing.
66	Monitoreo post-lanzamiento y recopilación de feedback	Alta	On going	Informe de feedback inicial recopilado.	Establecer herramientas de monitoreo en tiempo real y canales de

					feedback.
67	Ajustes post-lanzamiento basados en feedback	Media	On going	Ajustes implementados según feedback.	Priorizar y aplicar mejoras y correcciones rápidas post-lanzamiento.
68	Desarrollo de documentación para usuarios finales	Baja	4	Documentación de usuario disponible.	Crear guías de usuario, FAQs y documentación de ayuda.
69	Capacitación del equipo de soporte	Media	4	Sesión de capacitación completada.	Preparar al equipo de soporte con conocimientos sobre la aplicación y solución de problemas comunes.

Pila del sprint

Historia de Usuario: Configuración del Entorno de Desarrollo

Puntos de Historia: 8 horas

Tareas

- Tarea ID 1: Configurar entorno de desarrollo local
 - Descripción: Instalar y configurar herramientas de desarrollo necesarias localmente.
 - Horas Estimadas: 4 horas
 - Responsable: Julio García Salas
 - Fecha de Terminación: 18 de marzo
- Tarea ID 2: Crear repositorio Git y establecer flujo de trabajo
 - Descripción: Inicializar repositorio Git y definir el flujo de trabajo del equipo.
 - Horas Estimadas: 4 horas
 - Responsable: Sofía García
 - Fecha de Terminación: 21 de marzo

Historia de Usuario: Configuración de Frameworks y Docker

Puntos de Historia: 16 horas

Tareas

- Tarea ID 3: Instalar framework backend (Vite y Express JS)

- Descripción: Configurar el entorno backend utilizando Vite y Express JS.
- Horas Estimadas: 6 horas
- Responsable: Julio García Salas
- Fecha de Terminación: 19 de marzo
- Tarea ID 4: Instalar framework frontend (React)
 - Descripción: Configurar el entorno frontend utilizando React.
 - Horas Estimadas: 6 horas
 - Responsable: Joaquín Campos
 - Fecha de Terminación: 19 de marzo
- Tarea ID 5: Configurar Docker para el entorno de desarrollo
 - Descripción: Crear Dockerfile y docker-compose.yml para contenerizar la aplicación.
 - Horas Estimadas: 4 horas
 - Responsable: Julio García Salas
 - Fecha de Terminación: 26 de marzo

Historia de Usuario: Base de Datos y API Inicial

Puntos de Historia: 20 horas

Tareas

- Tarea ID 6: Configurar gestor de base de datos (PostgreSQL)
 - Descripción: Instalar y configurar PostgreSQL como sistema de gestión de base de datos.
 - Horas Estimadas: 4 horas
 - Responsable: Sofía García
 - Fecha de Terminación: 25 de marzo
- Tarea ID 7: Crear estructura de proyecto y organizar archivos
 - Descripción: Definir la estructura del proyecto y organizar los archivos de código fuente.
 - Horas Estimadas: 4 horas
 - Responsable: Juanfer Menendez
 - Fecha de Terminación: 20 de marzo
- Tarea ID 8: Crear endpoints básicos de API para productos
 - Descripción: Desarrollar endpoints CRUD para la gestión de productos.
 - Horas Estimadas: 12 horas
 - Responsable: Sofía García y Julio García Salas
 - Fecha de Terminación: 29 de marzo

Historia de Usuario: Página de Inicio Estática y Documentación

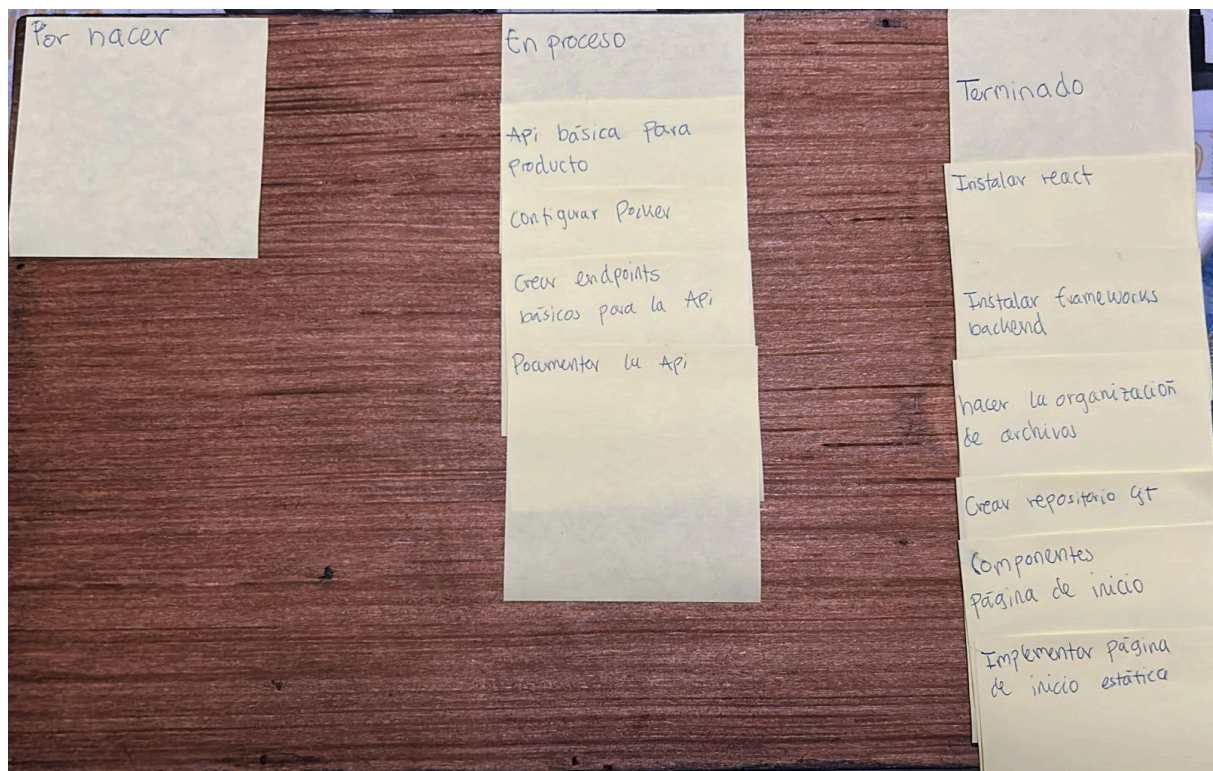
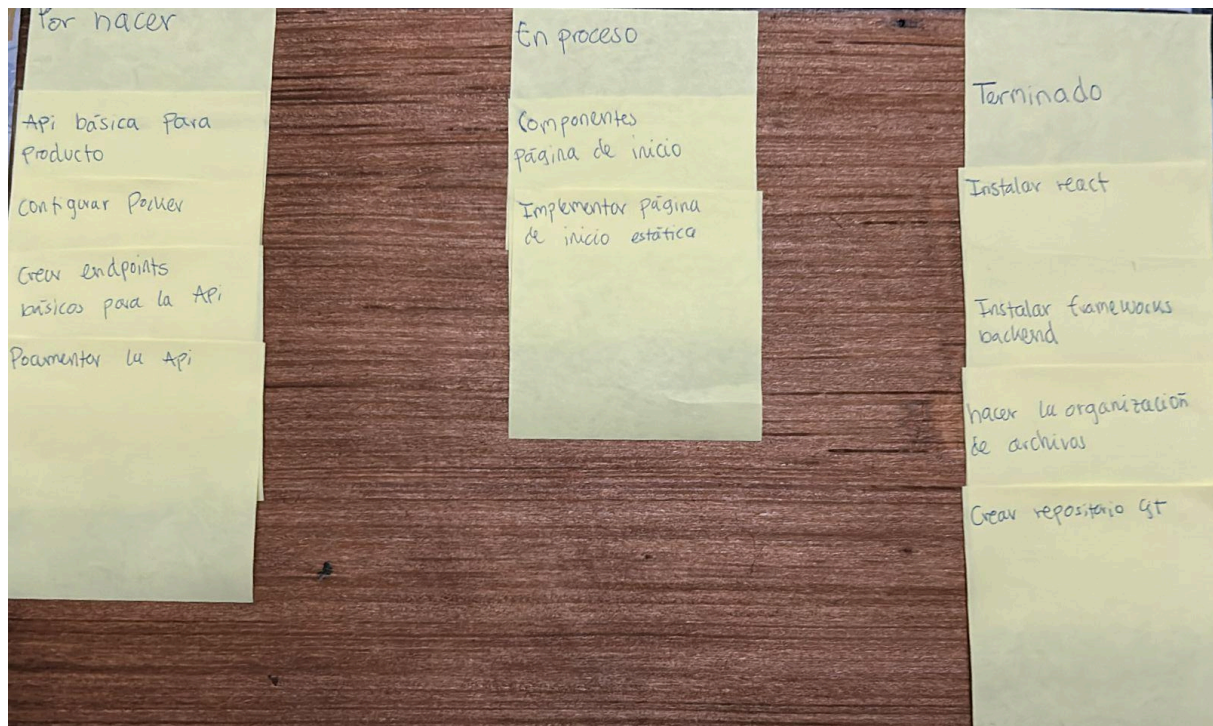
Puntos de Historia: 12 horas

Tareas

- Tarea ID 9: Implementar página de inicio estática

- Descripción: Crear una página de inicio básica para la web app.
 - Horas Estimadas: 8 horas
 - Responsable: Joaquín Campos, Juanfer Menendez y Juan Pablo Solis
 - Fecha de Terminación: 29 de marzo
- Tarea ID 10: Documentar la API inicial
 - Descripción: Escribir documentación básica para los endpoints de la API desarrollados.
 - Horas Estimadas: 4 horas
 - Responsable: Sofía García y Julio García Salas
 - Fecha de Terminación: 2 de abril

Resultados del sprint



Se lograron hacer todas las tareas planificadas para este sprint, sin embargo se cree que quedó un poco holgado el tiempo, por lo que para próximos sprints se subirá la carga de tareas para poder medir mejor la variable eficiencia vs tareas.

Métricas del sprint

Gráfica Burndown



Velocidad del sprint

- Se concluyó el sprint con un puntaje de 35 puntos completando las 12 tareas asignadas previamente para el sprint, de manera que podemos concluir que las historias de usuarios tenían alta necesidad pero no un “puntaje” tan alto de manera que se podría considerar un promedio bajo por cantidad de integrantes, pero también se debe considerar que además de ser el primer sprint, podemos trabajar en el desempeño.

Discusión

Cumplimiento de Objetivos

- Positivo: El equipo completó las 12 tareas asignadas, lo cual indica que se cumplieron los objetivos del sprint en términos de volumen de trabajo.
- Para mejorar: La distribución de la carga de trabajo a lo largo del sprint fue desigual, con un esfuerzo considerable concentrado hacia el final. Esto indica una planificación o estimación inicial no óptima.

Gestión del Tiempo y Distribución del Trabajo

- Positivo: La capacidad del equipo para completar todas las tareas, incluso con un impulso final, demuestra flexibilidad y compromiso.
- Para mejorar: Una distribución más uniforme del trabajo podría ayudar a evitar el estrés y el riesgo asociados a la acumulación de tareas hacia el final del sprint.

Calidad del Trabajo

- A considerar: Aunque el gráfico muestra que se completaron todas las tareas, la calidad de las tareas completadas también es importante, lo que se debe mejorar para próximos sprints. Es importante complementar esta discusión con retroalimentación sobre la calidad y si se necesitaron compromisos para completar las tareas a tiempo.

Comunicación y Colaboración en el Equipo

- Positivo: La finalización exitosa de las tareas sugiere una buena colaboración dentro del equipo, especialmente en la fase de alta presión hacia el final.
- Para mejorar: Sería útil analizar si la comunicación efectiva dentro del equipo y con los stakeholders podría mejorarse para identificar cuellos de botella o redistribuir tareas antes en el sprint.

Sostenibilidad del Esfuerzo

- Preocupación: La concentración del trabajo hacia el final del sprint plantea preguntas sobre la sostenibilidad de este enfoque. Es crucial considerar el bienestar del equipo y si tal patrón de trabajo es viable a largo plazo.

Aprendizajes y Mejoras para Sprints Futuros

La reflexión sobre este sprint debe llevar a ajustes en la planificación y ejecución de sprints futuros. Considerar métodos para una mejor estimación de tareas, técnicas de priorización más efectivas, y estrategias para fomentar una distribución más equitativa del trabajo a lo largo del sprint.

Código desarrollado

Link al repositorio:

https://github.com/JPS4321/React_Proyecto

Lista de Funcionalidades Terminadas

- Barra de navegación
- Marquee (Texto infinito cíclico)
- Hero Section
- Carta de producto
- Catálogo de cartas de producto
- Pie de página
- Api básica para producto
- Base de datos para poder almacenar los datos del producto dummy

Documentación

Reflexión de desempeño

Desde el inicio del scrum se había definido como nos dividiríamos el trabajo de manera que el equipo rápidamente se estableció y empezó a trabajar, en este primer sprint no se tenía previsto avanzar demasiado, de manera que se cumplió rápidamente.

Lo que deberíamos mejorar como equipo es el tener presente que dependemos de los demás, en otras palabras podríamos trabajar en la organización del tiempo en el que se deben hacer las tareas, porque si las tenemos bien divididas pero no óptimamente para que todos trabajemos al mismo tiempo, de esta manera a veces hay compañeros que se quedan sin trabajo en el momento.

Link al documento

https://docs.google.com/document/d/17nQb813Lmqg8nApKeOzzzxP8n26meJMPnwVdvE_g_a4/edit?usp=sharing