

UNIVERSIDAD DEL VALLE DE GUATEMALA

Ingeniería de Software 2, sección 30



Excelencia que trasciende

DEL VALLE
GRUPO EDUCATIVO

DÉCIMO SPRINT

Sofia Garcia - 22210

Julio Garcia Salas - 22076

Joaquin Campos - 22155

Juan Fernando Menéndez - 17444

Juan Pablo Solis - 22102

GUATEMALA, Septiembre de 2024

Índice

Índice.....	2
Resumen.....	2
Introducción.....	2
Product backlog.....	2
Pila del sprint.....	18
Resultados del sprint.....	21
Métricas del sprint.....	22
Velocidad del Sprint.....	22
Discusión.....	23
Código desarrollado.....	23
Presupuesto.....	24
Documentación.....	25

Resumen

En el décimo sprint del proyecto de la web app tipo CRUD para la ecommerce de venta de bikinis, el foco principal fue la refactorización y mejora del código para asegurar la calidad y sostenibilidad del sistema a largo plazo. Se realizaron optimizaciones en la presentación de los formularios, mejorando la modularidad y escalabilidad del código, y se ajustaron los componentes de autenticación para garantizar un manejo adecuado de permisos y persistencia de sesiones.

El equipo trabajó en resolver problemas relacionados con la mantenibilidad de los estilos CSS globales, implementando principios de DRY para centralizar reglas repetitivas y asegurar la consistencia visual en toda la aplicación. Además, se hicieron modificaciones clave en el manejo del formulario de inventario, facilitando la edición de productos y mejorando la gestión de estado con el uso de `useState` y `useEffect`.

También se llevaron a cabo ajustes en la integración entre el frontend y el backend para mejorar la visualización y gestión de datos en los reportes de órdenes y transacciones diarias. Esto permitió una interacción más sólida y redujo los errores de datos, mejorando así la experiencia de usuario y la fiabilidad del sistema.

El resultado de este sprint no solo cumplió con los objetivos de refactorización y optimización, sino que también disminuyó la deuda técnica del proyecto y facilitó la escalabilidad de futuras implementaciones, alineándose con los objetivos generales de la empresa para mejorar la eficiencia operativa y la experiencia del usuario.

Introducción

El décimo sprint del proyecto de la web app tipo CRUD para la ecommerce de venta de bikinis se enfocó en consolidar la calidad y la eficiencia del sistema mediante una serie de refactorizaciones y optimizaciones. Hasta este punto, se habían implementado las funcionalidades básicas y avanzadas de la aplicación, incluidas la gestión de productos, categorías y autenticación. No obstante, era necesario abordar la deuda técnica y asegurar que el código fuera sostenible, escalable y de fácil mantenimiento.

Este sprint fue diseñado para mejorar la arquitectura del código, garantizando que los formularios, tanto en el frontend como en la interacción con el backend, operaran de manera fluida y sin errores. También se trabajó en la consistencia visual de la aplicación mediante la refactorización de los estilos CSS, aplicando principios de reutilización de código para una mejor experiencia de usuario. Finalmente, el equipo se centró en robustecer la gestión de la autenticación y los permisos de los usuarios, asegurando la persistencia de las sesiones y reduciendo riesgos de seguridad en el manejo de roles y accesos.

Los esfuerzos realizados en este sprint no solo impactaron en la optimización del sistema, sino que también sentaron una base sólida para futuras expansiones y mejoras en la funcionalidad de la aplicación. Esto aseguró que la web app continuara evolucionando de manera eficiente y alineada con los objetivos de la empresa.

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 flujo de trabajo	Media	2	Repositorio creado y primer commit.	Mostrar el repositorio en 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	Alta	2	Estructura de carpetas mostrada en la demo.	Seguir las mejores prácticas de

	organizar archivos				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

1 2	Investigación de palabras clave SEO	Alta	4	Lista de palabras clave objetivo.	documentación. Usar herramientas como SEMrush o Google Keyword Planner.
1 3	Crear la base de datos preliminar completa	Alta	3	Diagrama ER de la base de datos	Definir tablas y relaciones según el DER
1 4	Creación de productService	Alta	3	Mostrar productService.js implementando las operaciones CRUD.	Incluir conexión a la base de datos y métodos CRUD.
1 5	Creación de productRoute	Alta	3	Mostrar productRoute.js con rutas de API para productos.	Definir endpoints para operaciones CRUD y middleware.
1 6	Documentación de producto	Media	3	Documentar métodos y rutas de productos.	Detallar uso, parámetros, cuerpos de

					solicitud y respuestas.
17	Creación de categoryService	Alta	3	Mostrar categoryService.js con funciones CRUD para categorías.	Métodos para recuperar todas las categorías y gestionar una sola.
18	Creación de categoryRoute	Alta	3	Mostrar categoryRoute.js con rutas de API para categorías.	Incluir rutas CRUD para categorías.
19	Documentación de categorías	Media	3	Documentar métodos y rutas de categorías.	Explicar uso, parámetros y respuestas.
20	Creación de colorService	Alta	3	Demostrar colorService.js manejando colores.	Funciones CRUD para colores.
21	Creación de colorRoute	Alta	3	Exhibir colorRoute.js con rutas para gestionar colores.	Rutas de API para operaciones de color.

2 2	Documentación de colores	Media	3	Documentar métodos y rutas de colores.	Detallar uso, parámetros, cuerpos de solicitud y respuestas.
2 3	Creación de collectionService	Alta	3	Presentar collectionService.js con lógica de negocio para colecciones.	Métodos CRUD para colecciones.
2 4	Creación de collectionRoute	Alta	3	Mostrar collectionRoute.js con rutas API para colecciones.	Rutas para interacción con información de colecciones.
2 5	Documentación de colecciones	Media	3	Documentar métodos y rutas de colecciones.	Instrucciones detalladas de uso y ejemplos de respuestas.
2 6	Creación de productColorService	Alta	3	Mostrar cómo productColorService.js relaciona productos con colores.	Asignación y eliminación de colores a productos.

27	Creación de rutas en productColorRoute	Alta	3	Enseñar productColorRoute.js gestionando asociaciones de colores de productos.	Rutas para crear y eliminar relaciones de productos y colores.
28	Documentación de productoColor	Media	3	Documentar métodos y rutas de asociación de colores de productos.	Proporcionar ejemplos de solicitudes y respuestas JSON.
29	Creación de clientService	Alta	3	Mostar como clientService.js funciona en postman	Crud básico para cliente
30	Creación de clientRoute	Alta	3	Mostrar que si está funcionando en postman	Rutas para el crud de cliente
31	Documentación de cliente	Media	3	Documentar métodos y rutas	Proporcionar ejemplos de solicitudes y respuestas JSON

3 2	Creación de todos los archivos front end público	Alta	3	Crear los archivos de las secciones necesarias	Crear carpeta para guardar css
3 3	Configuración de rutas	Alta	3	Configurar las rutas a todas las páginas creadas	
3 4	Conexión de rutas a la navbar	Baja	1	Hacer que desde la navbar se puedan acceder a las demás páginas	Asegurarse que los botones no se muestren como hipervínculos
3 5	Creación Componentes pagina Product	Media	2	Hacer los componentes necesario para la página de product	
3 6	Implementación componentes pagina Product	Media	1	Implementar los componentes creados en la pagina produc	
3 7	Edición pagina product con css	baja	1	Editar la página Product con css	
3 8	Creación Componentes página About us	Media	2	Hacer los componentes necesario para la página About us	

39	Implementación componentes página About Us	Media	1	Implementar los componentes creados en la página About Us	
40	Edición página About Us con css	baja	1	Editar la página About us con css	
41	Creación template básico Main Page (Empleados) para empleados	Alta	1	Crear el archivo Main Page para los empleados	
42	Creación Componentes Main Page (Empleados)	Media	2	Crear los componentes necesario para la Main page	
43	Implementación de componentes a la pagina	Alta	2	Implementar los componentes creados	
44	Creación de productCollection Service	Alta	3	Implementar CRUD en productCollectionService.js	Incluir conexión a base de datos y métodos CRUD.
45	Creación de productCollection Route	Alta	3	Implementar rutas en productCollectionRoute.js	Definir endpoints para operaciones CRUD.

46	Documentación de productColecciones	Alta	3	Documentar productCollectionService y Route	Detallar uso, parámetros, cuerpos de solicitud y respuestas.
47	Creación de InventoryService	Alta	3	Implementar CRUD en InventoryService.js	Incluir conexión a base de datos y métodos CRUD.
48	Creación de inventoryRoute	Alta	3	Implementar rutas en inventoryRoute.js	Definir endpoints para operaciones CRUD.
49	Documentación del inventario	Alta	3	Documentar InventoryService y Route	Detallar uso, parámetros, cuerpos de solicitud y respuestas.
50	Creación de promotionService	Alta	3	Implementar CRUD en promotionService.js	Incluir conexión a base de datos y métodos CRUD.

51	Creación de promotionRoute	Alta	3	Implementar rutas en promotionRoute.js	Definir endpoints para operaciones CRUD.
52	Documentación de promotion	Alta	3	Documentar promotionService y Route	Detallar uso, parámetros, cuerpos de solicitud y respuestas.
53	Creación de orderService	Alta	3	Implementación de CRUD en orderService.js	Incluir conexión a base de datos y métodos CRUD.
54	Creación de orderRoute	Alta	3	Implementación de rutas en orderRoute.js	Definir endpoints para operaciones CRUD.
55	Documentación de la API de órdenes	Media	2	Documentar orderService y orderRoute	Detallar uso, parámetros, cuerpos de solicitud y respuestas.

56	Creación de orderDetailService	Alta	3	Implementación de CRUD en orderDetailService.js	Incluir conexión a base de datos y métodos CRUD.
57	Creación de orderDetailRoute	Alta	3	Implementación de rutas en orderDetailRoute.js	Definir endpoints para operaciones CRUD.
58	Documentación de la API detalles de órdenes	Media	2	Documentar orderDetailService y orderDetailRoute	Detallar uso, parámetros, cuerpos de solicitud y respuestas.
59	Creación de productPromotionService	Alta	3	Implementación de CRUD en productPromotionService.js	Incluir conexión a base de datos y métodos CRUD.
60	Creación de productPromotionRoute	Alta	3	Implementación de rutas en productPromotionRoute.js	Definir endpoints para operaciones CRUD.

61	Documentación de la API de promociones de productos	Media	2	Documentar productPromotionService y productPromotionRoute	Detallar uso, parámetros, cuerpos de solicitud y respuestas.
62	Configuración de Google Search Console	Alta	2	Sitio verificado en Google Search Console.	Configurar sitemap y verificar indexación.
63	Diseño de la arquitectura backend	Alta	4	Diagrama de arquitectura del backend.	Definir estructura de carpetas y módulos principales.
64	Configuración inicial de Vite para backend	Alta	2	Proyecto Vite corriendo con Express JS.	Mostrar servidor Express funcionando con Vite.
65	Creación de modelos de datos con Sequelize	Alta	4	Modelos para Producto, Categoría y Orden implementados.	Usar Sequelize como ORM para PostgreSQL.

66	Implementación de routing en React	Alta	3	Navegación entre páginas de inicio y gestión de inventarios.	Usar React Router para gestionar rutas.
67	Implementación de la UI en React	Alta	8	Componentes React para gestión de inventarios.	Convertir los mockups de Figma a componentes React.
68	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.
69	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.
70	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.
71	Crear componentes UI reutilizables	Alta	8	Biblioteca de componentes UI.	Desarrollar botones, formularios, tarjetas, y otros elementos de UI en React para reutilización.
72	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.
73	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.

74	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.
75	Integrar sistema de búsqueda avanzada	Alta	6	Funcionalidad de búsqueda avanzada.	Permitir a los usuarios buscar productos por diferentes criterios.
76	Desarrollar sistema de notificaciones	Media	4	Sistema de notificaciones en la app.	Notificar a los usuarios sobre cambios importantes, como bajos niveles de inventario.
77	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.
78	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.
79	Personalizar temas y estilos de la app	Media	5	Temas personalizados aplicados.	Permitir a los usuarios elegir entre temas claros y oscuros para mejorar la experiencia visual.
80	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.
81	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.
82	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.
83	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.
84	Implementación de endpoints para	Alta	6	API endpoints para productos funcionando.	Crear, leer, actualizar, eliminar productos

	CRUD de productos				a través de la API.
85	Pruebas unitarias para modelos de datos	Alta	4	Pruebas unitarias para modelos pasando.	Usar Jest o Mocha para pruebas de modelos Sequelize.
86	Pruebas unitarias para endpoints de productos	Alta	4	Pruebas unitarias para API de productos pasando.	Asegurar que los endpoints retornan los datos esperados.
87	Configuración de seguridad básica en Express	Alta	3	Middleware de seguridad implementado.	Usar helmet y CORS en el servidor Express.
88	Implementación de autenticación JWT	Alta	4	Sistema de autenticación con JWT funcionando.	Permitir login y proteger rutas de la API con JWT.
89	Documentación de la API con Swagger	Media	3	Documentación de la API disponible.	Usar Swagger para documentar los

					endpoints de la API.
90	Configuración inicial de SSL con Let's Encrypt	Media	3	HTTPS configurado para el sitio web.	Demostrar el sitio funcionando sobre HTTPS.
91	Optimización inicial de imágenes para la web	Media	3	Imágenes optimizadas en la página de inicio.	Usar herramientas como ImageOptim o servicios en línea.
92	Implementación de sistema de logging	Media	3	Logs de aplicación y errores configurados.	Usar Winston o Morgan para logging en el backend.
93	Preparación de entorno de pruebas para frontend	Media	3	Configuración de Jest para pruebas de React.	Mostrar pruebas básicas de componentes React.
94	Preparación y ejecución de pruebas de carga	Baja	4	Pruebas de carga ejecutadas en la API.	Usar Artillery o JMeter para simular tráfico.

95	Revisión de accesibilidad web	Baja	3	Auditoría de accesibilidad completada.	Usar herramientas como Lighthouse para evaluar accesibilidad.
96	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.
97	Creación de contenido SEO para la página de inicio	Alta	4	Contenido optimizado para SEO implementado.	Escribir y optimizar el contenido de la página de inicio.
98	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.
99	Integración de pasarela de pago	Alta	8	Demostrar transacciones de prueba.	Integrar Stripe o PayPal para

					manejar pagos.
100	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.
101	Implementación de filtros de búsqueda para productos	Alta	5	Filtros de búsqueda funcionando.	Permitir a los usuarios filtrar productos por categoría, precio, etc.
102	Optimización de la base de datos	Alta	4	Mejoras en el rendimiento de consultas demostradas.	Indexar columnas clave y revisar consultas para optimización.
103	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.
104	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.
105	Pruebas de seguridad de la aplicación	Alta	6	Reporte de pruebas de seguridad.	Realizar pruebas de penetración y corregir vulnerabilidades encontradas.
106	Configuración de backups automáticos para la base de datos	Alta	3	Sistema de backups configurado y probado.	Asegurar que los backups de la base de datos se realicen diariamente.
107	Preparativos finales para el lanzamiento	Alta	5	Checklist de lanzamiento completado.	Revisar SEO, seguridad, rendimiento y funcionalidad

					ades clave antes del lanzamiento.
108	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.
109	Implementación de funcionalidades de reportes y análisis	Alta	6	Sistema de reportes implementado.	Desarrollar dashboards para seguimiento de ventas, inventario y análisis de usuarios.
110	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.

1 1 1	Revisión final de accesibilidad web	Me dia	3	Auditoría de accesibilidad completada y mejoras aplicadas.	Asegurar cumplimiento con WCAG y mejorar la accesibilidad general.
1 1 2	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.
1 1 3	Monitoreo post-lanzamiento y recopilación de feedback	Alta	Ongoing	Informe de feedback inicial recopilado.	Establecer herramientas de monitoreo en tiempo real y canales de feedback.
1 1 4	Ajustes post-lanzamiento basados en feedback	Me dia	Ongoing	Ajustes implementados según feedback.	Priorizar y aplicar mejoras y correcciones rápidas post-lanzamiento.

1 1 5	Desarrollo de documentación para usuarios finales	Baja	4	Documentación de usuario disponible.	Crear guías de usuario, FAQs y documentación de ayuda.
1 1 6	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.
1 1 7	Arreglo de imports	Media	1	Mostrar los cambios en los archivos de importación	Corrección realizada en archivos de configuración para mejorar la gestión de dependencias.
1 1 8	Arreglo del navbar	Alta	2	Mostrar navbar completamente responsivo	Ajustes para mejorar la responsividad y el

					alineamiento del navbar.
1 1 9	Primera versión arreglo statbat	Media	1	Mostrar el statbar alineado y responsivo.	Cambios iniciales para mejorar la visualización y funcionalidad del statbar.
1 2 0	Borrar cosas innecesarias	Alta	3	Confirmación de eliminación de archivos innecesarios.	Limpieza de código para optimizar el proyecto.
1 2 1	Statbar cambios iniciales	Media	2	Mostrar versión inicial del statbar con mejoras aplicadas.	Cambios para mejorar la estructura y estilo del statbar.
1 2 2	Statbar responsive y alineado	baja	2	Demostrar la responsividad y alineación del statbar.	Ajustes para asegurar la correcta visualización en diferentes dispositivos.

1 2 3	Navbar full responsive	Media	1	<p>Mostrar funcionalidad responsive completa del navbar.</p>	<p>Modificaciones para mejorar la compatibilidad con diversos tamaños de pantalla.</p>
1 2 4	Arreglitos x	Media	2	<p>Mostrar pequeños ajustes realizados en el proyecto.</p>	<p>Correcciones menores en el código para optimizar funcionalidades.</p>
1 2 5	Linea bonita	Alta	3	<p>Mostrar mejoras visuales en elementos de línea.</p>	<p>Estilización y mejoras en los elementos de línea para una interfaz más atractiva.</p>

1 2 6	Cambio al contenido.jsx se agregó un diccionario para mostrar inventario	Alta	2	<p>Mostrar la funcionalidad de lista de inventario con diccionario.</p>	Implementación de un diccionario para mejorar la gestión y visualización de inventarios.
1 2 7	Componente para mostrar la lista de inventario	Alta	1	Mostrar componente de lista de inventario en acción.	Creación de un componente dedicado para gestionar y mostrar el inventario.
1 2 8	Estilos de la lista de inventario	Alta	3	<p>Mostrar los estilos aplicados a la lista de inventario.</p>	Diseño y aplicación de estilos para mejorar la interfaz de la lista de inventario.

1 2 9	Se agrega el salesbar	Alta	3	Demostrar la funcionalidad y estilos del salesbar.	Incorporación de un salesbar para mejorar la interfaz de ventas.
1 3 0	Se agregan los estilos y cambios a algunos archivos	Media	2	Mostrar los archivos afectados por los cambios de estilos.	Ajustes de estilos para mejorar la coherencia visual del proyecto.
1 3 1	Responsive la lista de productos en inventario	Baja	1	Demostrar la lista de productos responsiva.	Asegurar que la lista de productos se muestre correctamente en diferentes dispositivos.
1 3 2	Cambios para corregir ciertos errores en responsive	Baja	2	Mostrar correcciones realizadas para mejorar la responsividad.	Ajustes específicos para solucionar problemas de responsividad detectados.

1 3 3	Completamente responsive correcciones de border-bottom y cambio de color		3	Mostrar los ajustes responsivos y visuales realizados.	Cambios en estilos para mejorar la visualización y funcionalidad responsiva.
1 3 4	Se agrega el componente para mostrar la lista de ordenes		3	Demostrar funcionalidad del componente de lista de ordenes.	Creación de un componente para la gestión y visualización de ordenes.
1 3 5	Se agregan los estilos del componente de lista de ordenes		1	Mostrar estilos aplicados al componente de lista de ordenes.	Diseño y aplicación de estilos específicos para el componente de lista de ordenes.
1 3 6	Crear shippingService	Alta	3	Mostrar shippingService.js implementando las operaciones CRUD	Incluir conexión a la base de datos y métodos CRUD

1 3 7	Crear shippingRoute	Alta	3	Mostrar shippingRoute.js con rutas de API para envíos	Definir endpoints para operaciones CRUD y middleware
1 3 8	Crear paymentService	Alta	3	Mostrar paymentService.js implementando las operaciones CRUD	Incluir conexión a la base de datos y métodos CRUD
1 3 9	Crear paymentRoute	Alta	3	Mostrar paymentRoute.js con rutas de API para pagos	Definir endpoints para operaciones CRUD y middleware
1 4 0	Agregar shippingService y paymentService al index.js	Alta	2	Mostrar index.js actualizado	Confirmar que los servicios de envío y pago están integrados correctamente
1 4 1	Iniciar dashboardPage	Alta	2	Mostrar progreso de la página del dashboard	Incluir elementos iniciales y

					estructura básica
1 4 2	Agregar checkbox para el chequeo de pedidos en salesPage	Alta	2	Mostrar checkbox funcionando	Confirmar que los pedidos pueden ser seleccionados correctamente
1 4 3	Añadir nombres a las columnas de la tabla de ventas	Alta	2	Mostrar nombres de columnas en salesPage	Confirmar que todas las columnas tienen nombres claros y descriptivos
1 5 9	Crear test de servicio al cliente	Alta	3	Mostrar clientService.js funcionando correctamente	Incluir todas las operaciones CRUD necesarias
1 6 0	Crear test de ruta de clientes	Alta	3	Mostrar clientRoute.js funcionando correctamente	Incluir todas las operaciones CRUD necesarias
1 6 1	Crear test del servicio de promociones	Alta	3	Mostrar promotionService.js funcionando correctamente	Incluir todas las operaciones CRUD necesarias

1 6 2	Crear test de la ruta de promociones	Alta	3	Mostrar promotionRoute.js funcionando correctamente	Incluir todas las operaciones CRUD necesarias
1 6 3	Crear test del servicio de productos	Alta	3	Mostrar productService.js funcionando correctamente	Incluir todas las operaciones CRUD necesarias
1 6 4	Crear test de ruta de productos	Alta	3	Mostrar productRoute.js funcionando correctamente	Incluir todas las operaciones CRUD necesarias
1 6 5	Crear test del servicio de orden	Alta	3	Mostrar orderService.js funcionando correctamente	Incluir todas las operaciones CRUD necesarias
1 6 6	Crear test de la ruta de orden	Alta	3	Mostrar orderRoute.js funcionando correctamente	Incluir todas las operaciones CRUD necesarias
1 6 7	Crear test del servicio de detalles de órdenes	Alta	3	Mostrar orderDetailService.js funcionando correctamente	Incluir todas las operaciones CRUD necesarias
1 6 8	Crear test de la ruta de los detalles de orden	Alta	3	Mostrar orderDetailRoute.js funcionando correctamente	Incluir todas las operaciones CRUD necesarias

169	Modificar el servicio de usuario	Alta	3	Mostrar userService.js funcionando correctamente.	Incluir las modificaciones necesarias en el servicio.
170	Crear middleware para usuarios	Alta	3	Implementar un middleware específico para usuarios.	Asegurarse de que userMiddleware.js funcione correctamente.
171	Agregar middleware de multer en productos	Alta	3	Incluir un middleware de multer para manejar imágenes en productos.	Verificar que la integración sea exitosa.
172	Actualizar producto para integrar imagen	Alta	3	Modificar la funcionalidad de productos para soportar la carga de imágenes.	Validar que los cambios estén implementados correctamente.
173	Crear rutas y servicios para usuario	Alta	3	Desarrollar las rutas y servicios necesarios para gestionar usuarios.	Verificar que todas las operaciones CRUD funcionen correctamente.
174	Cambio a la base de datos para	Alta	3	Modificar la base de datos para incluir una nueva tabla de usuarios.	Asegurarse de que la migración y los

	agregar tabla de usuarios				cambios sean exitosos.
175	Crear Docker Compose para deploy	Alta	3	Configurar un Docker Compose para desplegar la aplicación.	Verificar que la aplicación se despliegue correctamente.
176	Crear usuarios y ruta con su test	Alta	3	Desarrollar la ruta de usuarios y los tests correspondientes.	Validar que la ruta y los tests funcionen correctamente.
177	Modificar test de clientes y ruta para que funcione	Alta	3	Actualizar los tests y rutas de clientes para asegurar su funcionamiento.	Validar que los cambios estén correctos.
178	Mejorar test de productos	Alta	3	Optimizar los tests de productos.	Asegurarse de que cubran todas las operaciones CRUD necesarias.
179	Corregir index y ruta de usuarios	Alta	3	Corregir el archivo index.js y la ruta de usuarios.	Validar que los cambios funcionen correctamente.

180	Crear ruta de clientes	Alta	3	Desarrollar la ruta para gestionar clientes.	Asegurarse de que la ruta esté funcionando correctamente.
181	Mostrar perfil de usuario con detalles básicos (correo, usuario, rol)	Alta	Completado	Implementado en UserProfile.js	Muestra la información básica del usuario como correo, nombre de usuario y rol.
182	Imagen de perfil predeterminada o personalizada	Alta	Completado	Implementado en UserProfile.js	Si no se proporciona una imagen, se usa una predeterminada.
183	Edición de permisos solo para roles específicos (Administrador y Supervisora de Tienda)		Completado	Implementado en UserProfile.js	Los permisos solo pueden ser modificados por Administradores o Supervisores de Tienda
184	Deshabilitar edición de permisos para otros roles		Completado	Implementado en UserProfile.js	Los checkboxes de permisos están

					deshabilita dos para roles que no son Administrador o Supervisor .
185	Logout funcional		Completado	Implementado en UserProfile.js	El botón de "Cerrar Sesión" cierra la sesión del usuario.
186	Diseño responsive para el perfil de usuario		Completado	Implementado en UserProfile.css	El diseño se adapta a pantallas más pequeñas para una mejor experiencia de usuario.
187	Pruebas de cambios para poder crear un producto	Alta	3	Mostrar la funcionalidad de creación de productos.	Incluir validaciones y asegurarse de que se puede añadir un producto sin errores.
188	Se crea el usePromocion	Media	3	Mostrar cómo el hook maneja las promociones dentro del producto.	Verificar que el hook se aplique correctamente al producto en la vista

					y esté bien enlazado.
189	Se edita el inventoryForm para manejar promociones	Media	3	Mostrar cómo se integran las promociones en el inventario.	Asegurarse de que los campos de promociones aparezcan y se puedan modificar desde el formulario.
190	Se mejora cómo funciona el productPromotion	Media	3	Mostrar la funcionalidad de promociones mejorada.	Asegurarse de que las promociones se apliquen correctamente a los productos y se reflejen en la UI.
191	Hook de productos	Alta	3	Demostrar el nuevo hook de productos funcionando.	Asegurarse de que el hook carga y actualiza productos correctamente.
192	Hook de colores	Media	3	Mostrar el hook de colores.	Verificar que los colores carguen y se puedan seleccionar sin problemas.

1 9 3	Hook de colecciones	Media	3	Mostrar el hook de colecciones funcionando.	Verificar que las colecciones se carguen correctamente y estén disponibles en el formulario.
1 9 4	Hook de categorías	Media	3	Demostrar el hook de categorías.	Asegurarse de que las categorías se pueden seleccionar correctamente en el formulario de productos.
1 9 5	Se agrega colección y color	Media	3	Demostrar la funcionalidad de agregar colección y color a productos.	Verificar que las colecciones y colores se añaden correctamente al crear un producto.
1 9 6	Cambio al service de producto	Alta	3	Mostrar cómo el servicio de producto gestiona los cambios.	Asegurarse de que el servicio está actualizado para manejar la segunda imagen y otros cambios.

197	Cambio al index	Media	3	Mostrar los cambios en el index.js.	Verificar que los cambios están correctamente integrados en el flujo del proyecto.
198	Se modifica la ruta de productos y los servicios para segunda imagen	Alta	3	Demostrar cómo se ha modificado la ruta de productos para manejar la segunda imagen.	Asegurarse de que el sistema pueda manejar correctamente múltiples imágenes por producto.
199	LoginPage modificado	Alta	3	Mostrar la página de login modificada.	Verificar que las modificaciones funcionen y conecten con el backend de autenticación.
200	Hook de login hecho	Alta	3	Demostrar el hook de login funcionando correctamente.	Asegurarse de que el hook maneje el proceso de autenticación sin errores.
201	SettingsPage modificado y	Alta	3	Mostrar los cambios en la SettingsPage conectada a la base de datos.	Verificar que los cambios

	conectado a la base de datos				se guardan correctamente en la base de datos.
202	AuthContext modificado	Alta	3	Demostrar los cambios en el AuthContext.	Asegurarse de que el contexto de autenticación esté actualizado y funcionando sin errores.
203	Cambio en los servicios de usuario	Media	3	Mostrar los cambios en los servicios de usuario.	Verificar que el servicio maneje correctamente las solicitudes relacionadas con usuarios.
204	Cambio en la ruta de usuarios	Media	3	Mostrar los cambios en la ruta de usuarios.	Verificar que la ruta esté correctamente enlazada al servicio y maneje bien las operaciones CRUD.
205	Cambios en las dependencias	Baja	3	Mostrar los cambios en las dependencias.	Asegurarse de que todas las dependencias estén actualizadas y no

					rompan el sistema.
206	Login creado	Alta	3	Demostrar la funcionalidad del login.	Verificar que el login funcione correctamente con el backend y frontend conectados.
207	Hook de useUser creado	Alta	3	Demostrar el hook de useUser.	Asegurarse de que el hook maneje bien la información del usuario autenticado.
208	Navbar actualizado	Medio	3	Mostrar los cambios en la navbar.	Verificar que la navbar esté actualizada y funcione correctamente con los cambios de autenticación.
209	Cambio a la ruta privada	Alta	3	Mostrar los cambios en la ruta privada.	Asegurarse de que la ruta privada maneje correctamente las autenticaciones.

					ones y permisos.
210	AuthContext	Alta	3	Mostrar el funcionamiento actualizado del AuthContext.	Verificar que la autenticación y los permisos están siendo manejados adecuadamente.
211	Cambio a la app	Alta	3	Mostrar los cambios en la app.	Asegurarse de que los cambios aplicados a la app funcionen sin problemas.
212	Cambios a las dependencias	Baja	3	Mostrar los cambios a las dependencias	Verificar que las dependencias actualizadas no generen problemas en el entorno de desarrollo.
213	Prueba de fetching de datos	Alta	3	Mostrar las pruebas de fetching de datos entre backend y frontend.	Asegurarse de que los datos se carguen correctamente desde el backend al frontend.
214	Implementación de ajustes finales en la aplicación	Alta	5	Disponible en la versión de despliegue final. Últimos cambios antes del cierre de sprint	

2 1 5	Optimización de funcionalidades y corrección de bugs menores	Media	3	Verificada en el entorno de desarrollo. Se realizaron ajustes para mejorar el rendimiento	
2 1 6	Actualización del formulario de inventario para soporte de edición	Alta	5	Verificada en la sección de inventario. El formulario ahora permite editar entradas correctamente	
2 1 7	Revisión y mejoras en el servicio de productos para mayor eficiencia	Media	3	Funcional en el módulo de productos. Ajustes internos para optimizar consultas y procesamiento	
2 1 8	Modificaciones en el formulario de inventario para mejoras de usabilidad	Media	3	Integrada y funcional en el entorno de pruebas. Se mejoró la interfaz de usuario y la validación	
2 1 9	Ajustes adicionales en el servicio de productos para compatibilidad con nuevas funciones	Media	3	Presentada en la revisión del sprint. Actualización para alinear con los últimos requisitos del proyecto	
2 2 0	Optimización del hook de producto para manejo de tablas correlacionales	Alta	5	Demostrada en la gestión de inventarios. Se mejoró el rendimiento al actualizar tablas correlacionales	
2 2 1	Mejoras al formulario de inventario para habilitar la función de edición	Alta	5	Revisada en la vista de inventario. Integración fluida para una mejor experiencia de usuario	
2 2 2	Refinamiento del servicio de productos para	Alta	5	Pruebas exitosas en el módulo de productos. Se incorporó la capacidad	

	soportar la funcionalidad de edición			de edición con retroalimentación mejorada	
2 2 3	Eliminación de fusión de logs de acceso no necesarios para optimización del sistema	Me dia	3	Evaluado en los registros del sistema. Limpieza de código y mejora de la gestión de logs	

Pila del sprint

ID: 214

Descripción: Implementación de ajustes finales en la aplicación

Puntos de Historia: 5

Horas: 8 horas

Realizado por: Julio García Salas

Fecha: 15 de octubre de 2024

ID: 215

Descripción: Optimización de funcionalidades y corrección de bugs menores

Puntos de Historia: 3

Horas: 5 horas

Realizado por: Julio García Salas

Fecha: 17 de octubre de 2024

ID: 216

Descripción: Actualización del formulario de inventario para soporte de edición

Puntos de Historia: 5

Horas: 7 horas

Realizado por: Julio García Salas

Fecha: 19 de octubre de 2024

ID: 217

Descripción: Revisión y mejoras en el servicio de productos para mayor eficiencia

Puntos de Historia: 3

Horas: 6 horas

Realizado por: Julio García Salas

Fecha: 21 de octubre de 2024

ID: 218

Descripción: Modificaciones en el formulario de inventario para mejoras de usabilidad

Puntos de Historia: 3

Horas: 5 horas

Realizado por: Julio García Salas

Fecha: 23 de octubre de 2024

ID: 219

Descripción: Ajustes adicionales en el servicio de productos para compatibilidad con nuevas funciones

Puntos de Historia: 3

Horas: 6 horas

Realizado por: Julio García Salas

Fecha: 25 de octubre de 2024

ID: 220

Descripción: Optimización del hook de producto para manejo de tablas correlacionales

Puntos de Historia: 5

Horas: 8 horas

Realizado por: Julio García Salas

Fecha: 27 de octubre de 2024

ID: 221

Descripción: Mejoras al formulario de inventario para habilitar la función de edición

Puntos de Historia: 5

Horas: 7 horas

Realizado por: Julio García Salas

Fecha: 29 de octubre de 2024

ID: 222

Descripción: Refinamiento del servicio de productos para soportar la funcionalidad de edición

Puntos de Historia: 5

Horas: 8 horas

Realizado por: Julio García Salas

Fecha: 30 de octubre de 2024

ID: 223

Descripción: Eliminación de fusión de logs de acceso no necesarios para optimización del sistema

Puntos de Historia: 3

Horas: 5 horas

Realizado por: Julio García Salas

Fecha: 1 de noviembre de 2024

ID: 224

Descripción: Se logró hacer que el formulario apareciera completo

Puntos de Historia: 3

Horas: 4 horas

Realizado por: Sofia Garcia

Fecha: 15 de octubre de 2024

ID: 225

Descripción: Se crea una clase aparte para editar el formulario

Puntos de Historia: 3

Horas: 5 horas

Realizado por: Sofia Garcia

Fecha: 18 de octubre de 2024

ID: 226

Descripción: Se edita el productInventory para que funcionara el botón nuevamente

Puntos de Historia: 3

Horas: 4 horas

Realizado por: Sofia Garcia

Fecha: 21 de octubre de 2024

ID: 227

Descripción: Implementación de diseño responsive en el sistema

Puntos de Historia: 3

Horas: 6 horas

Realizado por: Sofia Garcia

Fecha: 24 de octubre de 2024

ID: 228

Descripción: Cambios a clases CSS para que tengan nombres específicos y no generales

Puntos de Historia: 3

Horas: 5 horas

Realizado por: Sofia Garcia
Fecha: 27 de octubre de 2024

ID: 229

Descripción: Cambios a clases CSS para que sean específicos y no globales

Puntos de Historia: 3

Horas: 5 horas

Realizado por: Sofia Garcia

Fecha: 30 de octubre de 2024

ID: 230

Descripción: Arreglo bug de Hook

Puntos de Historia: 1

Horas: 1 hora

Realizado por: Juan Fernando Menendez

Fecha: 25 de octubre de 2024

ID: 231

Descripción: Navbar Responsive

Puntos de Historia: 3

Horas: 4 hora

Realizado por: Juan Fernando Menendez

Fecha: 25 de octubre de 2024

ID: 232

Descripción: About Us Responsive

Puntos de Historia: 2

Horas: 3 hora

Realizado por: Juan Fernando Menendez

Fecha: 26 de octubre de 2024

ID: 233

Descripción: Contact Section Responsive

Puntos de Historia: 2

Horas: 3 hora

Realizado por: Juan Fernando Menendez

Fecha: 27 de octubre de 2024

ID: 234

Descripción: Footer Responsive

Puntos de Historia: 2

Horas: 3 hora

Realizado por: Juan Fernando Menendez

Fecha: 28 de octubre de 2024

ID: 235

Descripción: Imagen en cart Item

Puntos de Historia: 1

Horas: 1 hora

Realizado por: Juan Fernando Menendez

Fecha: 30 de octubre de 2024

ID: 236

Descripción: Creación Hook useOrder

Puntos de Historia: 1

Horas: 1 hora

Realizado por: Juan Pablo Solis

Fecha: 25 de octubre de 2024

ID: 237

Descripción: Modificación ContactForm, Shipping Form y payment screen para transferir valores de variables

Puntos de Historia: 3

Horas: 3 hora

Realizado por: Juan Pablo Solis

Fecha: 25 de octubre de 2024

ID: 238

Descripción: Implementación de funcionalidad para crear un cliente al presionar botón en payment screen

Puntos de Historia: 3

Horas: 5 hora

Realizado por: Juan Pablo Solis

Fecha: 26 de octubre de 2024

ID: 239

Descripción: Implementación de funcionalidad para enviar datos a la tabla de ordenes

Puntos de Historia: 3

Horas: 4 hora

Realizado por: Juan Pablo Solis

Fecha: 27 de octubre de 2024

ID: 240

Descripción: Creación de componente Create en pagina admin

Puntos de Historia: 2

Horas: 2 hora

Realizado por: Jose Joaquin Campos

Fecha: 25 de octubre de 2024

ID: 241

Descripción: Creación de formulario para eliminar

Puntos de Historia: 2

Horas: 2 hora

Realizado por: Jose Joaquin Campos

Fecha: 25 de octubre de 2024

ID: 242

Descripción: Creación de formulario para eliminar

Puntos de Historia: 2

Horas: 2 hora

Realizado por: Jose Joaquin Campos

Fecha: 25 de octubre de 2024

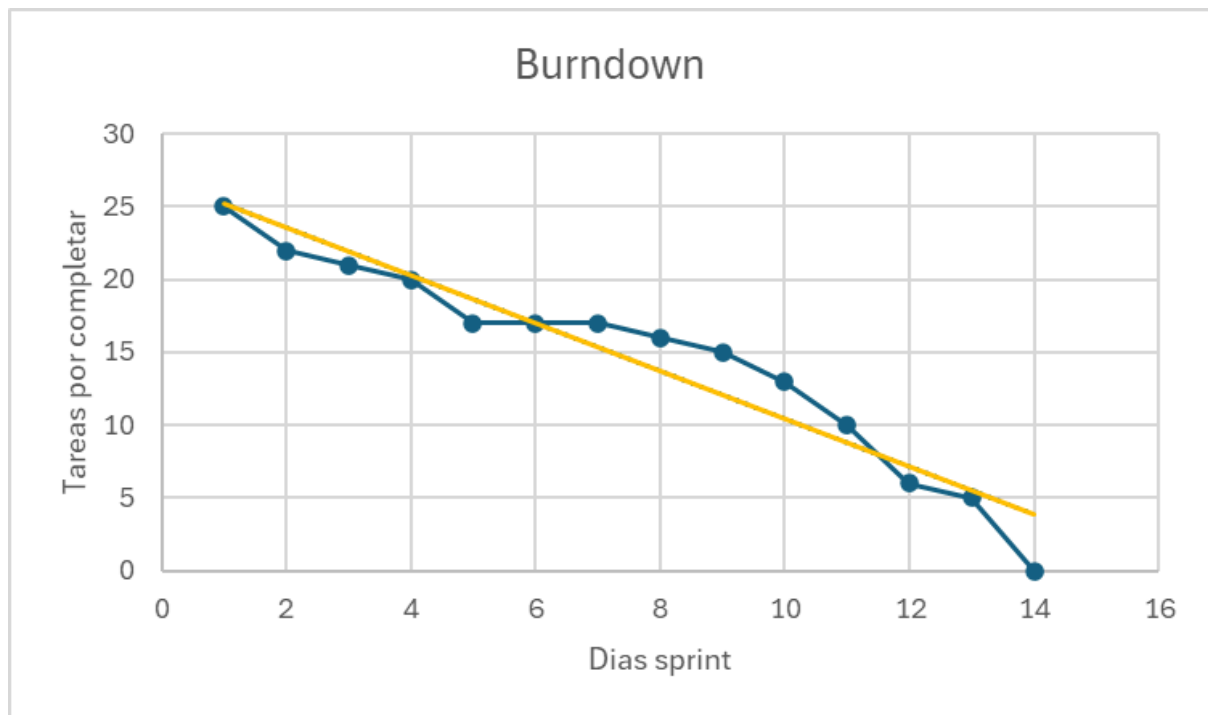
Resultados del sprint

Tablero de jira:

<https://uvg-team-g8inmhj5.atlassian.net/jira/software/projects/SCRUM/boards/1?sprintStartDate=true&atlassianOrigin=eyJpIjoiOTViM2ViZWU3ZWQwNDEzMDk2Y2QwNTFkYTQ4MzBiMWEiLCJwIjoiajI9>

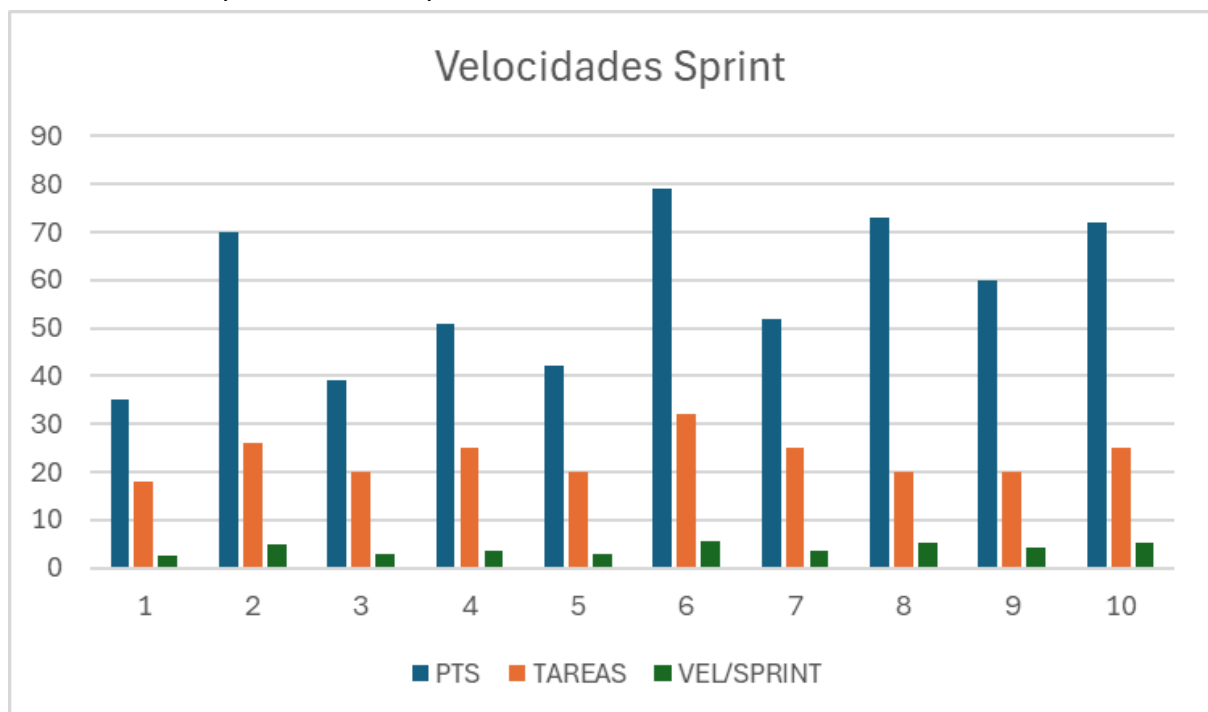
Se lograron hacer todas las tareas planificadas para este sprint, y en cambio con el último sprint ya se logró colocar un mejor margen entre tareas asignadas y tiempo.

Métricas del sprint



Velocidad del Sprint

velocidades comparadas entre sprints:



Comparación de velocidades según sprint

- Se concluyó el sprint con un puntaje de 72 puntos completando 25 tareas de las 34 asignadas previamente para el sprint.

- En cuanto a velocidad media del Sprint es de 5.14 podemos decir que la velocidad que se obtuvo en este sprint fue por debajo del promedio de lo que hemos logrado obtener de los sprints anteriores no obstante solo por la entrevista podemos rescatar bastantes puntos..

Discusión

Cumplimiento de Objetivos

- Positivo: El equipo completó las 25 tareas, lo cual indica que se cumplieron los objetivos del sprint en términos de volumen de trabajo.
- Para mejorar: Actualmente se podría decir que se está acumulando lo último del proyecto al poder conectar localmente con base de datos se ha hecho un gran avance no obstante se debe de adelantar bastante aun ya que en base a esto han surgido otros problemas

Gestión del Tiempo y Distribución del Trabajo

- Positivo: La capacidad del equipo para finalizar todas las tareas, incluso con un esfuerzo extra al final, refleja el compromiso de la mayoría de los estudiantes.
- Para mejorar: Una distribución de trabajo más equilibrada podría ayudar a evitar el estrés y los riesgos relacionados con la acumulación de tareas al cierre del proyecto. El desafío es mantener el compromiso entre todos los miembros.

Calidad del Trabajo

- A considerar: Aunque el gráfico indica que todas las tareas fueron completadas, es esencial tener en cuenta la calidad de las mismas, especialmente con la conexión entre la base de datos y otra parte que nos preocupa.

Comunicación y Colaboración en el Equipo

- Positivo: La finalización exitosa de las tareas refleja la productividad del sprint.
- Para mejorar: Es fundamental mejorar la comunicación entre los compañeros y asignar tareas a aquellos que no están contribuyendo suficientemente.

Sostenibilidad del Esfuerzo

- Preocupación: Existe cierta diferencia en la cantidad de tareas distribuidas entre los miembros del equipo.

Aprendizajes y Mejoras para Sprints Futuros

La reflexión sobre este sprint es que podríamos mejorar, la comunicación entre todos podría ayudar a encontrar tareas ya que el product backlog ya está casi delimitada.

Código desarrollado

https://github.com/JPS4321/React_Proyecto

Lista de funcionalidades terminadas:

- Implementación de ajustes finales en la aplicación
- Optimización de funcionalidades y corrección de bugs menores
- Actualización del formulario de inventario para soporte de edición

- Revisión y mejoras en el servicio de productos para mayor eficiencia
- Modificaciones en el formulario de inventario para mejoras de usabilidad
- Ajustes adicionales en el servicio de productos para compatibilidad con nuevas funciones
- Optimización del hook de producto para manejo de tablas correlacionales
- Mejoras al formulario de inventario para habilitar la función de edición
- Refinamiento del servicio de productos para soportar la funcionalidad de edición
- Eliminación de fusión de logs de acceso no necesarios para optimización del sistema
- Se logró hacer que el formulario apareciera completo
- Se crea una clase aparte para editar el formulario
- Se edita el productInventory para que funcionara el botón nuevamente
- Implementación de diseño responsive en el sistema
- Cambios a clases CSS para que tengan nombres específicos y no generales
- Cambios a clases CSS para que sean específicos y no globales
- Arreglo bug de Hook
- Navbar Responsive
- About Us Responsive
- Contact Section Responsive
- Footer Responsive
- Imagen en cart Item
- Creación Hook useOrder
- Modificación ContactForm, Shipping Form y payment screen para transferir valores de variables
- Implementación de funcionalidad para crear un cliente al presionar botón en payment screen
- Implementación de funcionalidad para enviar datos a la tabla de órdenes
- Creación de componente Create en página admin
- Creación de formulario para eliminar

Presupuesto Actualizado

- **Fase Actual:** 10 sprints completados
- **Duración de un Sprint:** 2 semanas (14 días)
- **Salario mensual:** \$1,400
- **Salario por sprint (2 semanas):** $\$1,400 / 2 = \700
- **Costo total por sprint para 5 programadores:** $\$700 \times 5 = \$3,500$

Cálculo del Costo Total hasta el Sprint 10:

- Costo total por sprint para 5 programadores: \$3,500
- Costo total por 10 sprints: $\$3,500 \times 10 = \$35,000$

Presupuesto Total Actualizado:

- Costo hasta el final del décimo sprint: \$35,000

Resumen:

- **Presupuesto hasta la fecha (10 sprints completados):** \$35,000

- **Presupuesto para la ejecución total del proyecto:** \$35,000

Refactorización de código

Durante este sprint, se llevaron a cabo refactorizaciones significativas para mejorar la calidad del código y reducir la deuda técnica acumulada en áreas críticas de la aplicación. Las mejoras se enfocaron en múltiples aspectos clave, incluyendo la optimización de la presentación y funcionalidad de componentes frontend y la organización de estilos CSS para mayor consistencia visual y mantenibilidad.

Uno de los puntos más destacados fue la refactorización del Formulario de Inventario (InventoryForm), donde se realizaron ajustes para asegurar que apareciera completo y fuera completamente funcional, permitiendo la edición de productos y garantizando una gestión de estado más clara y eficiente mediante el uso de `useState` y `useEffect`. También se creó una clase aparte para gestionar la edición, mejorando la modularidad del código.

En cuanto a los estilos, se realizaron cambios en las clases CSS para que tuvieran nombres específicos y no generales, mejorando la legibilidad y minimizando el riesgo de conflictos en el proyecto. Además, se trabajó en hacer el sistema más responsive, asegurando una experiencia de usuario óptima en diferentes dispositivos. Esta refactorización también incluyó ajustes al `ProductInventory` para reactivar la funcionalidad de ciertos botones y lograr un manejo más intuitivo de la interfaz.

Por último, se realizaron mejoras en la gestión de órdenes y envíos de datos a las tablas, garantizando que la información se procesara correctamente entre el frontend y el backend. Estos cambios permitieron una integración más robusta y la reducción de errores en el manejo de datos, lo que contribuye a un sistema más eficiente y escalable. En conjunto, estas refactorizaciones han reducido la deuda técnica y facilitado la expansión futura de la aplicación, incrementando la fiabilidad y mantenibilidad del proyecto.

Revisión técnica formal 2

Durante la revisión técnica formal 2, se realizó una evaluación exhaustiva del estado actual de la web app tipo CRUD para la ecommerce de venta de bikinis. Esta revisión tuvo como objetivo identificar problemas subyacentes en la aplicación y determinar los pasos a seguir para el último sprint del proyecto. El equipo técnico se enfocó en revisar la implementación de las recientes refactorizaciones y optimizaciones, así como en validar la eficiencia de las mejoras realizadas en la autenticación, el manejo de formularios, y la integración entre el frontend y el backend.

Resultados de la Revisión: Se detectaron un par de bugs que afectaban la experiencia de usuario y el rendimiento general de la aplicación. Uno de los problemas más relevantes fue un error intermitente en la carga de las imágenes de productos cuando se realizaba la actualización de los datos en el formulario de inventario. Este bug ocasionaba que, en

ciertas ocasiones, las imágenes no se mostraran correctamente después de ser actualizadas, lo que podía confundir a los usuarios al gestionar los productos.

Otro hallazgo importante fue una inconsistencia en el manejo de permisos de usuario que, bajo circunstancias específicas, permitía que ciertos roles accedieran a funciones no autorizadas. Aunque este problema no comprometía directamente la seguridad de la aplicación, era necesario resolverlo para mantener la integridad del sistema de permisos y garantizar que los usuarios solo pudieran realizar acciones asignadas a sus roles específicos.

Cambios Planeados para el Último Sprint: Durante el último sprint, el equipo se centrará en corregir los bugs identificados, incluyendo la mejora en la carga de imágenes para garantizar su correcta visualización tras la actualización de datos. Se implementarán pruebas adicionales de integración para asegurar que la carga de datos en el formulario de inventario funcione de manera robusta y eficiente. Además, se realizarán ajustes en el sistema de autenticación y gestión de permisos para reforzar la seguridad y prevenir accesos no autorizados.

Por último, se llevarán a cabo pruebas de estrés y optimización para confirmar que el sistema funcione adecuadamente bajo cargas elevadas y que los cambios realizados no introduzcan nuevos problemas de rendimiento. El objetivo es finalizar el proyecto con un sistema estable, optimizado y listo para futuras expansiones y mantenimientos, asegurando así el éxito a largo plazo de la ecommerce.

Anexo

LOGT Joaquin Campos:
22155

Fecha	Objetivo	General	Inicio	Fin	Tiempo Interrupción	Delta tiempo	Fase	Comentarios
20/9/24	Componente de Roles	Componente de roles	5:00PM	5:30 PM	0 min	30min	Sprint 11	Crear el componente de roles para poder llevar a cabo los diferentes roles de cada usuario

20/9/24	contenido para crear en page create	contenido para crear en page create	11:30 PM	12:30AM	0	1:50 min	Sprint 11	Se creo area de usuario para la webpage de admin
21/09/24	tarjetas para agregar y eliminar colecciones	tarjetas para agregar y eliminar colecciones	11:30 AM	2 PM	30min	2hrs	Sprint 6	Revalidacion funcionalidades de que la base de datos tenga relación con todas las funciones estaremos dándole acceso al usuario.
21/09/24	tarjetas para agregar y eliminar colores	tarjetas para agregar y eliminar colores	12:40 am	1:10am	0	40min	Sprint 6	Se creo la gráfica con la herramienta Excel
23/09/24	tarjetas para agregar y eliminar promociones	tarjetas para agregar y eliminar promociones	5:00pm	7:00pm	30	1:30 min	Sprint 6	Se sumaron los puntos de historia para obtener la métrica de velocidad
26/09	tarjetas para agregar y eliminar categorias	tarjetas para agregar y eliminar categorias	3PM	5PM	0	2hrs	Sprint 6	Se le agrego la función del texto para que este pueda seleccionarse en base a roles de administradores
27/09	arreglos conectando localmente con bd	arreglos conectando localmente con bd						

LOGT: Juan Pablo Solis
22102

Fecha	Objetivo	General	Inicio	Fin	Tiempo Interrupción	Delta tiempo	Fase	Comentarios
25/09/24	Mejora al sistema de filtrado	Ahora el sistema toma el precio con descuento.	5:00PM	5:30 PM	0 min	30min	Sprint 5	Ya filtra correctamente
26/09/24	Arreglo de bug en slider	El slider ahora filtra conforme se mueve	12:30PM	1:30AM	0	1:00 min	Sprint 5	Se modifiko el slider para que este filtrara de manera activa
27/09/24	Deshabilitacion de boton si no hay inventario	Ahora los botones de tallas en la pagina product no se pueden clicar si no hay inventario	12:30 AM	1:30 PM	0min	1hrs	Sprint 5	

LOGT: Julio García Salas

Carnet: 22076

Fecha Inicio	Fecha Fin	Tiempo Interrupción	Delta Tiempo	Fase	Comentarios
15/10/2024 09:00 AM	15/10/2024 12:00 PM	0	3 horas	Sprint 10	Últimos intentos de cambios
17/10/2024 01:00 PM	17/10/2024 04:00 PM	0	3 horas	Sprint 10	Más cambios

19/10/2024 10:00 AM	19/10/2024 01:00 PM	0	3 horas	Sprint 10	Se intenta cambiar el InventoryForm para que pueda funcionar el edit correctamente
21/10/2024 09:00 AM	21/10/2024 12:00 PM	0	3 horas	Sprint 10	Otra edición al servicio de productos
23/10/2024 01:00 PM	23/10/2024 04:00 PM	0	3 horas	Sprint 10	Cambios al InventoryForm
25/10/2024 09:00 AM	25/10/2024 12:00 PM	0	3 horas	Sprint 10	Más cambios al servicio de productos
27/10/2024 10:00 AM	27/10/2024 01:00 PM	0	3 horas	Sprint 10	Se mejora el hook de producto para llenar tablas correlacionales
29/10/2024 02:00 PM	29/10/2024 05:00 PM	0	3 horas	Sprint 10	Se mejora el formulario de inventario para que se pueda editar
30/10/2024 09:00 AM	30/10/2024 12:00 PM	0	3 horas	Sprint 10	Se mejora el servicio de productos para poder editarlos
01/11/2024 01:00 PM	01/11/2024 04:00 PM	0	3 horas	Sprint 10	Eliminación de fusión de logs de acceso no necesarios

Fecha Inicio	Fecha Fin	Tiempo Interrupción	Delta Tiempo	Fase	Comentarios
15/10/2024 09:00 AM	15/10/2024 01:00 PM	0	4 horas	Sprint 10	Se logró hacer que el formulario apareciera completo
18/10/2024 09:00 AM	18/10/2024 02:00 PM	0	5 horas	Sprint 10	Se crea una clase aparte para editar el formulario
21/10/2024 10:00 AM	21/10/2024 02:00 PM	0	4 horas	Sprint 10	Se edita el productInventory para que funcionara el botón nuevamente
24/10/2024 09:00 AM	24/10/2024 03:00 PM	0	6 horas	Sprint 10	Responsive design implementado en el sistema
27/10/2024 01:00 PM	27/10/2024 06:00 PM	0	5 horas	Sprint 10	Cambios a clases CSS para que tengan nombres específicos y no generales
30/10/2024 09:00 AM	30/10/2024 02:00 PM	0	5 horas	Sprint 10	Cambios a clases CSS para que sean específicos y no globales
15/10/2024 09:00 AM	15/10/2024 01:00 PM	0	4 horas	Sprint 10	Se logró hacer que el formulario apareciera completo
18/10/2024 09:00 AM	18/10/2024 02:00 PM	0	5 horas	Sprint 10	Se crea una clase aparte para editar el formulario

21/10/2024 10:00 AM	21/10/2024 02:00 PM	0	4 horas	Sprint 10	Se edita el productInventory para que funcionara el botón nuevamente
24/10/2024 09:00 AM	24/10/2024 03:00 PM	0	6 horas	Sprint 10	Responsive design implementado en el sistema
27/10/2024 01:00 PM	27/10/2024 06:00 PM	0	5 horas	Sprint 10	Cambios a clases CSS para que tengan nombres específicos y no generales
30/10/2024 09:00 AM	30/10/2024 02:00 PM	0	5 horas	Sprint 10	Cambios a clases CSS para que sean específicos y no globales

LOGT: Juan Menéndez
Carnet:17444

Fecha Inicio	Fecha Fin	Tiempo Interrupción	Delta Tiempo	Fase	Comentarios
28/10/2024 1:00 PM	23/09/2024 01:30 PM	0	1:30 horas	Sprint 10	Arreglos responsive de la página About us
28/10/2024 3:30 PM	13/09/2024 4:30 PM	0	1 hora	Sprint 10	Arreglos responsive Navbar
29/10/2024 4:00 PM	14/09/2024 6:00 PM	0	2 horas	Sprint 10	Arreglos responsive formulario de contacto

29/10/2024 6:00 PM	15/09/2024 8:00 PM	0	2 horas	Sprint 10	Arreglos responsive a footer
31/10/2024 4:00 PM	16/09/2024 6:00 PM	0	2 horas	Sprint 10	Actualización de visualización de producto de carrito incluyendo imagen de producto