

UNIVERSIDAD DEL VALLE DE GUATEMALA

Ingeniería de Software 2, sección 30



Excelencia que trasciende

DEL VALLE
GRUPO EDUCATIVO

NOVENO 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 noveno sprint del proyecto de la web app tipo CRUD para la ecommerce de venta de bikinis, el equipo centró sus esfuerzos en reducir la deuda técnica y mejorar la seguridad de la aplicación. Se realizó una refactorización significativa en áreas críticas, como la autenticación y la gestión de inventarios, para optimizar el rendimiento y la mantenibilidad del código. Además, se implementaron mejoras en la integración del frontend con el backend, asegurando que los datos de productos, clientes y órdenes se gestionen de manera eficiente y segura. También se avanzó en la visualización de reportes financieros diarios, agregando el cálculo de ingresos por día y la gestión de nuevos clientes, con una interfaz más amigable y consistente. Estos cambios preparan la base para sprints futuros, donde se seguirán optimizando las funciones clave y asegurando la estabilidad del sistema.

Introducción

El noveno sprint tiene como objetivo principal abordar la deuda técnica acumulada y mejorar la seguridad general del sistema. Hasta este punto, la aplicación ha implementado con éxito las funcionalidades de CRUD, pero ahora se busca optimizar el código y fortalecer la infraestructura. Se están haciendo mejoras en la gestión de inventarios, autenticación de usuarios y permisos, así como en la generación de reportes diarios para la administración de la tienda. Este sprint también introduce nuevas funciones para el cálculo de ingresos por día y la integración completa de clientes y productos en la visualización de datos. Además, la refactorización del código garantiza que las futuras expansiones sean más fáciles de implementar, reduciendo riesgos de errores y mejorando la eficiencia del sistema en general.

Product backlog

I D	Descripción	Im p	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 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	Crear la base de datos preliminar completa	Alta	3	Diagrama ER de la base de datos	Definir tablas y relaciones según el DER

14	Creación de productService	Alta	3	Mostrar productService.js implementando las operaciones CRUD.	Incluir conexión a la base de datos y métodos CRUD.
15	Creación de productRoute	Alta	3	Mostrar productRoute.js con rutas de API para productos.	Definir endpoints para operaciones CRUD y middleware.
16	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.
22	Documentación de colores	Media	3	Documentar métodos y rutas de colores.	Detallar uso, parámetros, cuerpos de solicitud y respuestas.
23	Creación de collectionService	Alta	3	Presentar collectionService.js con lógica de negocio para colecciones.	Métodos CRUD para colecciones.
24	Creación de collectionRoute	Alta	3	Mostrar collectionRoute.js con rutas API para colecciones.	Rutas para interacción con información de

					colecciones.
25	Documentación de colecciones	Media	3	Documentar métodos y rutas de colecciones.	Instrucciones detalladas de uso y ejemplos de respuestas.
26	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 productColor	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
32	Creación de todos los archivos front end público	Alta	3	Crear los archivos de las secciones necesarias	Crear carpeta para guardar css
33	Configuración de rutas	Alta	3	Configurar las rutas a todas las páginas creadas	
34	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 hipervinculos

3 5	Creación Componentes pagina Product	Me dia	2	Hacer los componentes necesario para la página de product	
3 6	Implementación componentes pagina Product	Me dia	1	Implementar los componentes creados en la pagina produc	
3 7	Edición pagina product con css	baj a	1	Editar la página Product con css	
3 8	Creación Componentes página About us	Me dia	2	Hacer los componentes necesario para la página About us	
3 9	Implementación componentes página About Us	Me dia	1	Implementar los componentes creados en la página About Us	
4 0	Edición página About Us con css	baj a	1	Editar la página About us con css	
4 1	Creación template básico Main Page (Empleados) para empleados	Alt a	1	Crear el archivo Main Page para los empleados	
4 2	Creación Componentes Main Page (Empleados)	Me dia	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 o 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.

7 6	Desarrollar sistema de notificaciones	Media	4	Sistema de notificaciones en la app.	Notificar a los usuarios sobre cambios importantes, como bajos niveles de inventario.
7 7	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.
7 8	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 CRUD de productos	Alta	6	API endpoints para productos funcionando.	Crear, leer, actualizar, eliminar 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.

9 1	Optimización inicial de imágenes para la web	Media	3	Imágenes optimizadas en la página de inicio.	Usar herramientas como ImageOptim o servicios en línea.
9 2	Implementación de sistema de logging	Media	3	Logs de aplicación y errores configurados.	Usar Winston o Morgan para logging en el backend.
9 3	Preparación de entorno de pruebas para frontend	Media	3	Configuración de Jest para pruebas de React.	Mostrar pruebas básicas de componentes React.
9 4	Preparación y ejecución de pruebas de carga	Baja	4	Pruebas de carga ejecutadas en la API.	Usar Artillery o JMeter para simular tráfico.
9 5	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.

					te 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 funcionalidades 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.
111	Revisión final de accesibilidad web	Media	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	Media	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	Demostrar funcionalidad responsiva completa del navbar.	Modificaciones para mejorar la compatibilidad con diversos

					tamaños de pantalla.
1 2 4	Arreglitos x	Media	2	Mostrar pequeños ajustes realizados en el proyecto.	Correcciones menores en el código para optimizar funcionalidades.
1 2 5	Linea bonita	Alta	3	Mostrar mejoras visuales en elementos de línea.	Estilización y mejoras en los elementos de línea para una interfaz más atractiva.
1 2 6	Cambio al contenido.jsx se agregó un diccionario para mostrar inventario	Alta	2	Demostrar la funcionalidad de lista de inventario con diccionario.	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	Mostrar los estilos aplicados a la lista de inventario.	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.

130	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.
131	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.
132	Cambios para corregir ciertos errores en responsive	Baja	2	Mostrar correcciones realizadas para mejorar la responsividad.	Ajustes específicos para solucionar problemas de responsividad detectados.
133	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	Mostrar 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
139	Crear paymentRoute	Alta	3	Mostrar paymentRoute.js con rutas de API para pagos	Definir endpoints para operaciones CRUD y middleware
140	Agregar shippingService y paymentService al index.js	Alta	2	Mostrar index.js actualizado	Confirmar que los servicios de envío y pago están integrados correctamente
141	Iniciar dashboardPage	Alta	2	Mostrar progreso de la página del dashboard	Incluir elementos iniciales y estructura básica
142	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
1 6 9	Modificar el servicio de usuario	Alta	3	Mostrar userService.js funcionando correctamente.	Incluir las modificaciones necesarias en el servicio.
1 7 0	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 agregar tabla de usuarios	Alta	3	Modificar la base de datos para incluir una nueva tabla de usuarios.	Asegurarse de que la migración y los 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 a 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 deshabilitados 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.
193	Hook de colecciones	Media	3	Mostrar el hook de colecciones funcionando.	Verificar que las colecciones se carguen correctamente y estén disponibles en el formulario.
194	Hook de categorías	Media	3	Demostrar el hook de categorías.	Asegurarse de que las categorías

					se pueden seleccionar correctamente en el formulario de productos.
195	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.
196	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

					nte múltiples imágenes por producto.
1 9 9	LoginPage modificado	Alta	3	Mostrar la página de login modificada.	Verificar que las modificacio nes funcionen y conecten con el backend de autenticaci ón.
2 0 0	Hook de login hecho	Alta	3	Demostrar el hook de login funcionando correctamente.	Asegurars e de que el hook maneje el proceso de autenticaci ón sin errores.
2 0 1	SettingsPage modificado y conectado a la base de datos	Alta	3	Mostrar los cambios en la SettingsPage conectada a la base de datos.	Verificar que los cambios se guardan correctame nte en la base de datos.
2 0 2	AuthContext modificado	Alta	3	Demostrar los cambios en el AuthContext.	Asegurars e de que el contexto de autenticaci ón esté actualizad o y funcionand o 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

					manejar bien la información del usuario autenticado.
208	Navbar actualizado	Media	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 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.
2 1 2	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.
2 1 3	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.
2 1 4	Cambio a userprofile css	Media	2	Mostrar los cambios aplicados en el CSS de UserProfile	Asegurarse de que los estilos del perfil sean consistentes y modernos.
2 1 5	Cambio a inventoryform	Alta	3	Verificar la edición del formulario de inventario	Revisar que los valores y el estilo del formulario funcionen correctamente.

216	Cambio a producto	Alta	3	Verificar la edición y actualización de productos	Asegurarse de que la funcionalidad de editar productos sea correcta.
217	CSS guapo	Media	2	Verificar los ajustes estéticos generales	Asegurarse de que los cambios mejoren la apariencia visual.
218	Mejora en settings content para que se pueda cerrar sesión	Alta	3	Presentar la funcionalidad de cierre de sesión	Asegurarse de que el usuario pueda cerrar sesión correctamente.
219	Cambios al inventoryform	Media	2	Presentar los ajustes en el formulario de inventario	Revisar que los ajustes adicionales al formulario funcionen como esperado.
220	Cambios al login css	Baja	1	Mostrar los cambios aplicados en el CSS de LoginPage	Asegurarse de que los estilos del login sean consistentes y

					funcionales.
2 2 1	Se cambia el inventoryform	Alta	3	Mostrar las mejoras en el formulario de inventario	Asegurarse de que las mejoras aplicadas funcionen correctamente.
2 2 2	Se agrega más CSS para el login y el authcontext	Alta	3	Verificar los cambios en el CSS y su aplicación en Login y AuthContext	Asegurarse de que los cambios en la autenticación estén correctos.
2 2 3	Authcontext para que funcione el f5	Alta	4	Mostrar que la funcionalidad de recargar la página mantiene al usuario logueado	Asegurarse de que la autenticación persista tras un refresh de página.
2 2 4	Para poder editar productos	Alta	4	Demostrar la funcionalidad completa de edición de productos	Asegurarse de que los productos se puedan editar y actualizar correctamente.
2 2 5	Se agrega dinero por día en la gráfica	Alta	4	Mostrar la visualización del dinero ganado por día en la gráfica	Asegurarse de que el gráfico refleje los ingresos

					correctamente.
2 2 6	Ahora aparece la cantidad de órdenes y se arregla el orden por día	Alta	4	Mostrar la cantidad de órdenes y su orden por día	Asegurarse de que el gráfico de órdenes se actualice correctamente.
2 2 7	Arreglar el eje x de la gráfica por días	Alta	4	Presentar la corrección en la visualización del eje X de la gráfica	Asegurarse de que el eje X muestre las fechas correctamente.
2 2 8	Ahora hay nuevos clientes	Alta	4	Mostrar la funcionalidad de visualización de nuevos clientes	Asegurarse de que los nuevos clientes se muestren y no se dupliquen.
2 2 9	Cambio al contenido de inventoryform para poder editarlo	Alta	3	Presentar la funcionalidad completa del formulario de inventario editable	Revisar que los cambios en el formulario permitan una edición correcta.
2 3 0	Correcciones a report	Medio	2	Mostrar las correcciones aplicadas al componente de reporte	Verificar que los cambios en el reporte se reflejen

					correctamente.
2 3 1	Se conecta el reporte con la base de datos	Alta	4	Demostrar que el reporte ahora está vinculado a la base de datos	Asegurarse de que los datos del reporte se obtienen correctamente.
2 3 2	Tabla de órdenes completa	Alta	4	Mostrar la implementación de la tabla de órdenes	Asegurarse de que la tabla de órdenes funciona correctamente.
2 3 3	Se arregló lo del monto	Alta	4	Demostrar la corrección en el cálculo del monto de las órdenes	Asegurarse de que el monto total se calcula y muestra correctamente.
2 3 4	Cambio a orderslist	Media	3	Verificar los cambios en la lista de órdenes	Asegurarse de que los ajustes en la lista de órdenes se reflejen bien.
2 3 5	Hook para órdenes	Alta	3	Mostrar la implementación del hook para gestionar órdenes	Asegurarse de que el hook para órdenes funciona correctamente.

Pila del sprint

- **ID 214**

Descripción: Cambio a `userprofile.css`

Puntos de Historia: 2

Horas: 2 horas

Realizado por: Julio García Salas

Fecha: 14 de Octubre de 2024

- **ID 215**

Descripción: Cambio a `inventoryform`

Puntos de Historia: 3

Horas: 3 horas

Realizado por: Julio García Salas

Fecha: 14 de Octubre de 2024

- **ID 216**

Descripción: Cambio a producto

Puntos de Historia: 3

Horas: 2 horas

Realizado por: Julio García Salas

Fecha: 14 de Octubre de 2024

- **ID 217**

Descripción: CSS guapo

Puntos de Historia: 2

Horas: 1 hora

Realizado por: Julio García Salas

Fecha: 14 de Octubre de 2024

- **ID 218**

Descripción: Se mejora el `settings` content para que se pueda cerrar sesión

Puntos de Historia: 3

Horas: 2 horas

Realizado por: Julio García Salas

Fecha: 14 de Octubre de 2024

- **ID 219**

Descripción: Cambios al [inventoryform](#)

Puntos de Historia: 2

Horas: 2 horas

Realizado por: Julio García Salas

Fecha: 14 de Octubre de 2024

- **ID 220**

Descripción: Cambios al login css

Puntos de Historia: 1

Horas: 1 hora

Realizado por: Julio García Salas

Fecha: 14 de Octubre de 2024

- **ID 221**

Descripción: Se cambia el [inventoryform](#)

Puntos de Historia: 3

Horas: 3 horas

Realizado por: Julio García Salas

Fecha: 14 de Octubre de 2024

- **ID 222**

Descripción: Se agrega más CSS para el login y el authcontext

Puntos de Historia: 3

Horas: 2 horas

Realizado por: Julio García Salas

Fecha: 14 de Octubre de 2024

- **ID 223**

Descripción: AuthContext para que funcione el F5

Puntos de Historia: 4

Horas: 2 horas

Realizado por: Julio García Salas

Fecha: 14 de Octubre de 2024

- **ID 224**

Descripción: Para poder editar productos

Puntos de Historia: 4

Horas: 4 horas

Realizado por: Julio García Salas
Fecha: 14 de Octubre de 2024

- **ID 225**

Descripción: Se agrega dinero por día en la gráfica
Puntos de Historia: 4
Horas: 3 horas
Realizado por: Sofía García
Fecha: 14 de Octubre de 2024

- **ID 226**

Descripción: Ahora aparece la cantidad de órdenes y se arregla el orden por día
Puntos de Historia: 4
Horas: 2 horas
Realizado por: Sofía García
Fecha: 14 de Octubre de 2024

- **ID 227**

Descripción: Arreglar el eje X de la gráfica por días
Puntos de Historia: 4
Horas: 1 hora
Realizado por: Sofía García
Fecha: 14 de Octubre de 2024

- **ID 228**

Descripción: Ahora hay nuevos clientes
Puntos de Historia: 4
Horas: 2 horas
Realizado por: Sofía García
Fecha: 14 de Octubre de 2024

- **ID 229**

Descripción: Cambio al contenido de `inventoryform` para poder editarlo
Puntos de Historia: 3
Horas: 2 horas
Realizado por: Sofía García
Fecha: 14 de Octubre de 2024

- **ID 230**

Descripción: Correcciones a **report**

Puntos de Historia: 2

Horas: 1 hora

Realizado por: Sofía García

Fecha: 14 de Octubre de 2024

- **ID 231**

Descripción: Se conecta el reporte con la base de datos

Puntos de Historia: 4

Horas: 3 horas

Realizado por: Sofía García

Fecha: 14 de Octubre de 2024

- **ID 232**

Descripción: Tabla de órdenes completa

Puntos de Historia: 4

Horas: 3 horas

Realizado por: Sofía García

Fecha: 14 de Octubre de 2024

- **ID 233**

Descripción: Se arregló lo del monto

Puntos de Historia: 4

Horas: 2 horas

Realizado por: Sofía García

Fecha: 14 de Octubre de 2024

- **ID 234**

Descripción: Cambio a **orderslist**

Puntos de Historia: 3

Horas: 2 horas

Realizado por: Sofía García

Fecha: 14 de Octubre de 2024

- **ID 235**

Descripción: Hook para órdenes

Puntos de Historia: 3

Horas: 2 horas

Realizado por: Sofía García
Fecha: 14 de Octubre de 2024

- **ID 236**

Descripción: Creación de hook para conectar a base de datos
Puntos de Historia: 3
Horas: 3 horas
Realizado por: Juan Pablo Solis
Fecha: 12 de Octubre de 2024

- **ID 237**

Descripción: Creación de hook para conectar a base de datos
Puntos de Historia: 3
Horas: 3 horas
Realizado por: Juan Pablo Solis
Fecha: 12 de Octubre de 2024

- **ID 238**

Descripción: Conexión página Collections a API
Puntos de Historia: 1
Horas: 1 horas
Realizado por: Juan Pablo Solis
Fecha: 12 de Octubre de 2024

- **ID 239**

Descripción: Conexión Product Page a API
Puntos de Historia: 3
Horas: 3 horas
Realizado por: Juan Pablo Solis
Fecha: 13 de Octubre de 2024

- **ID 240**

Descripción: Solucion discounted Price
Puntos de Historia: 2
Horas: 2 horas
Realizado por: Juan Pablo Solis
Fecha: 14 de Octubre de 2024

- **ID 241**

Descripción: Conexión JSON a shopping Cart

Puntos de Historia: 3

Horas: 2 horas

Realizado por: Juan Fernando Menendez

Fecha: 14 de Octubre de 2024

- **ID 242**

Descripción: Actualización JSON a shopping cart

Puntos de Historia: 1

Horas: 2 horas

Realizado por: Juan Fernando Menendez

Fecha: 14 de Octubre de 2024

- **ID 243**

Descripción: Solución de bug CSS con cart

Puntos de Historia: 2

Horas: 2 horas

Realizado por: Juan Fernando Menendez

Fecha: 14 de Octubre de 2024

- **ID 244**

Descripción: Lógica de reporte

Puntos de Historia: 2

Horas: 2 horas

Realizado por: Jose Joaquin Campos

Fecha: 14 de Octubre de 2024

- **ID 245**

Descripción: Acceso a administración

Puntos de Historia: 2

Horas: 2 horas

Realizado por: Jose Joaquin Campos

Fecha: 14 de Octubre de 2024

- **ID 246**

Descripción: creación y estilo de página a admin de usuarios

Puntos de Historia: 2

Horas: 2 horas

Realizado por: Jose Joaquin Campos

Fecha: 15 de Octubre de 2024

- **ID 247**

Descripción: componente contenido de admin de usuarios

Puntos de Historia: 2

Horas: 2 horas

Realizado por: Jose Joaquin Campos

Fecha: 15 de Octubre de 2024

Resultados del sprint

Proyectos / Mi proyecto de scrum

Tablero Sprint 9



EN CURSO

POR HACER

+ Crear incidencia

LISTO 20 ✓

Ajustes de UI basados en feedback de pruebas
SCRUM-64 ✓

Hook para ordenes
SCRUM-298 ✓

cambio orderlist
SCRUM-299 ✓

acceso admin de usuarios
SCRUM-302 ✓

Reportes
SCRUM-303 ✓

login
SCRUM-312 ✓

Conexion reporte bd
SCRUM-301 ✓

arreglos a producto
SCRUM-314 ✓

settings content para poder cerrar sesion
SCRUM-313 ✓

arreglos a inventory form
SCRUM-315 ✓

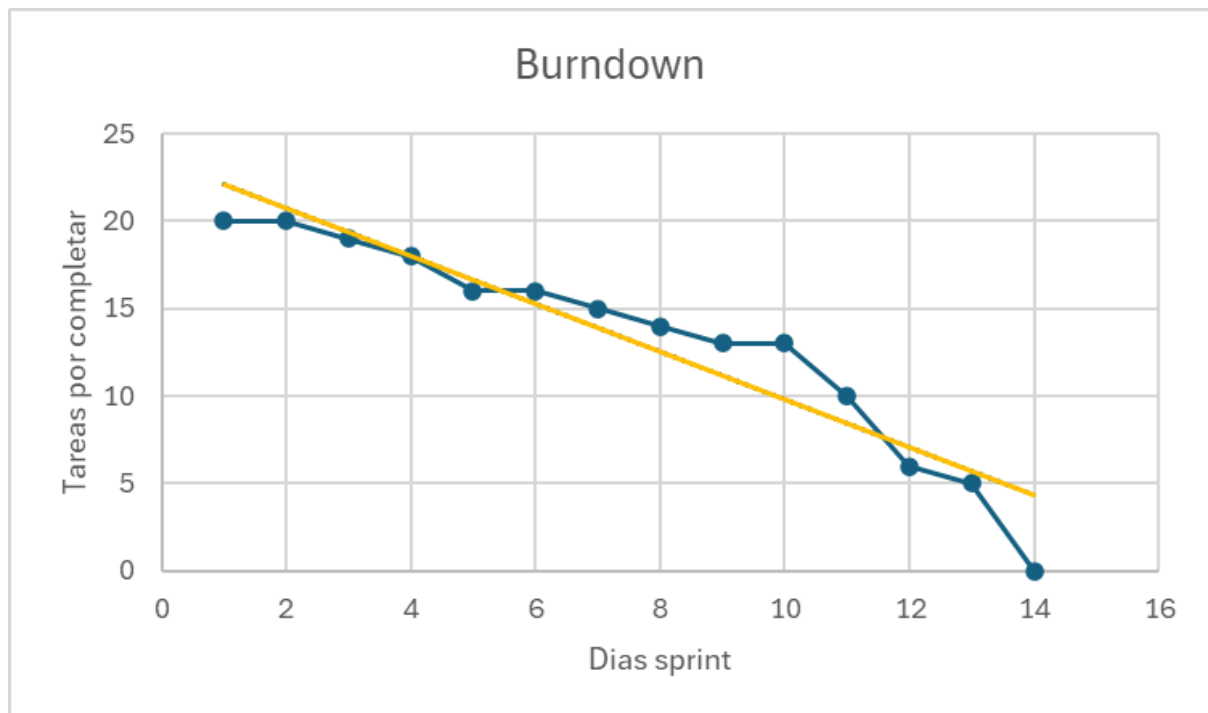
inventoryform

Tablero de jira:

<https://uvg-team-g8inmhj5.atlassian.net/jira/software/projects/SCRUM/boards/1?sprintStarted=true&atlOrigin=eyJpIjoiOTViM2VIZWU3ZWQwNDEzMdk2Y2QwNTFkYTQ4MzBiMWEiLCJwIjoiY9>

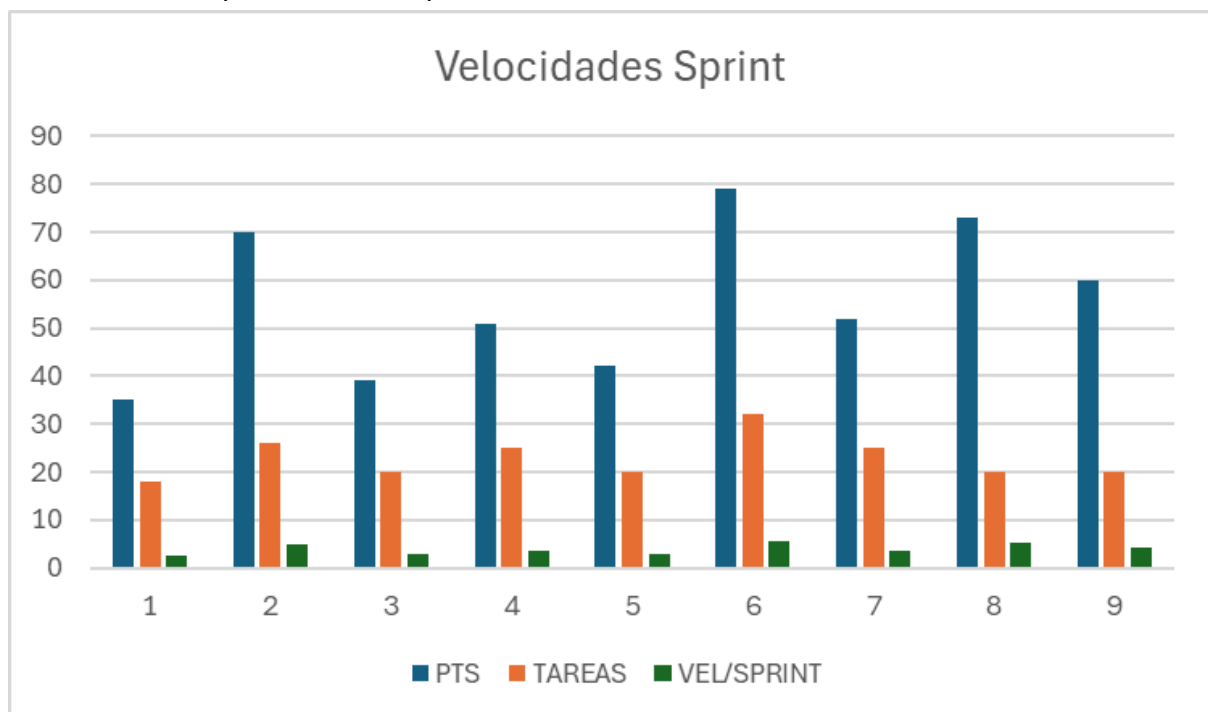
Se lograron hacer todas las tareas planificadas para este sprint, y en cambio con el último sprint ya se logró colocar un mejor margen entre tareas asignadas y tiempo.

Métricas del sprint



Velocidad del Sprint

velocidades comparadas entre sprints:



Comparación de velocidades según sprint

- Se concluyó el sprint con un puntaje de 60 puntos completando las 20 tareas de las 20 asignadas previamente para el sprint.

- En cuanto a velocidad media del Sprint es de 4.28 podemos decir que la velocidad que ya nos estamos adelantando del promedio En relación a sprints pasados.

Discusión

Cumplimiento de Objetivos

- Positivo: El equipo completó las 20 tareas, lo cual indica que se cumplieron los objetivos del sprint en términos de volumen de trabajo.
- Para mejorar: al poder hacer las conexiones bien podemos tener la mejora de poder interactuar.

Gestión del Tiempo y Distribución del Trabajo

- Positivo: La capacidad del equipo para completar todas las tareas, incluso con un impulso final, demuestra compromiso en la mayoría de estudiantes
- 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 proyecto el problema es el compromiso entre todos los integrantes.

Calidad del Trabajo

- A considerar: Aunque el gráfico muestra que se completaron todas las tareas, la calidad de las tareas completadas también es importante, en base a la conexión entre la base de datos y la otra nos tiene preocupados

Comunicación y Colaboración en el Equipo

- Positivo: La finalización exitosa de las tareas sugiere la productividad del sprint.
- Para mejorar: Es importante que consideremos la comunicación entre los compañeros tocara imponer tareas a los que no aportan.

Sostenibilidad del Esfuerzo

- Preocupación: La cantidad de tareas distribuidas entre los integrantes del equipo se puede ver un poco diferenciada.

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:

- Arreglo de bugs en formulario de envío
- Simulación de opciones de pago
- Arreglos visuales en formulario de envío
- Crear campos obligatorios en formulario de envío
- Arreglar campos de dirección
- Arreglo de bug product cards clickable
- Funcionalidad descuentos en la product card
- Animacion hover en product cards

- Arreglar bugs en la navbar presente en todas las páginas
- Arreglo de bugs en página collections
- Creacion de login para admin
- Creacion de Context.jsx admin
- Creacion de Private root para admin
- Implementación de Login para admin
- Crear middleware para usuarios
- Agregar middleware de multer en productos
- Actualizar producto para integrar imagen
- Crear rutas y servicios para usuario
- Cambio a la base de datos para agregar tabla de usuarios
- Crear Docker Compose para deploy
- Crear usuarios y ruta con su test
- Modificar test de clientes y ruta para que funcione
- Mejorar test de productos
- Corregir index y ruta de usuarios
- Crear ruta de clientes
- Mostrar perfil de usuario con detalles básicos (correo, usuario, rol)
- Imagen de perfil predeterminada o personalizada
- Edición de permisos solo para roles específicos (Administrador y Supervisora de Tienda)
- Deshabilitar edición de permisos para otros roles
- Logout funcional
- Diseño responsive para el perfil de usuario

Presupuesto Actualizado

- Fase Actual: 9 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: $\$700 \times 5 = \$3,500$

Costo total hasta el noveno sprint

- Costo hasta el final del noveno sprint: $\$3,500 \times 9 = \$31,500$

Cálculo del Presupuesto para la Ejecución Total del Proyecto

- Total de sprints: 10 sprints
- Sprints completados: 9
- Sprints restantes: 1 sprint

Cálculo de sprints restantes

- Costo por sprint para 5 programadores: \$3,500
- Costo total por 1 sprint adicional: $\$3,500 \times 1 = \$3,500$

Presupuesto Total Estimado

- Costo hasta el final del noveno sprint: \$31,500
- Costo para el sprint adicional: \$3,500
- Presupuesto total estimado: $\$31,500 + \$3,500 = \$35,000$

Resumen

- Presupuesto hasta la fecha (9 sprints completados): \$31,500
- Presupuesto para la ejecución total del proyecto: \$35,000

Refactorización de código

El objetivo de la refactorización en este sprint fue mejorar la calidad del código y reducir la deuda técnica acumulada en las áreas más críticas de la aplicación. Esto incluyó la optimización de la gestión del estado en el frontend, mejoras en la interacción con la base de datos, y el refactor del código CSS para mejorar la mantenibilidad y la consistencia visual. También nos enfocamos en reducir los riesgos de seguridad relacionados con la autenticación y el manejo de permisos.

2. Partes del Código Refactorizadas en Este Sprint

Durante este sprint, se abordaron las siguientes áreas clave del código para refactorización:

1. Formulario de Inventario (InventoryForm):

- **Refactorización realizada:** Se optimizó el formulario de inventario para que permitiera la edición de productos. Se mejoró la gestión de estado con `useState` y `useEffect`, haciendo que el código sea más fácil de mantener y extensible. Se centralizó el manejo de imágenes y datos relacionados con los productos, lo que reduce duplicación de código y errores.
- **Impacto en la deuda técnica:** Anteriormente, la lógica de manejo de inventarios era repetitiva y difícil de escalar. La refactorización ha permitido un código más limpio, con una gestión clara de los productos, lo que reduce la probabilidad de errores en futuras expansiones o modificaciones del sistema.

2. Autenticación y Gestión de Permisos (AuthContext, UserProfile):

- **Refactorización realizada:** Se mejoró el manejo del contexto de autenticación para garantizar que los usuarios permanezcan autenticados incluso después de un refresh en la página. Además, se reorganizó el manejo de permisos por roles, optimizando la forma en que se gestionan los privilegios de los usuarios dentro de la aplicación.
- **Impacto en la deuda técnica:** Anteriormente, el código de autenticación carecía de persistencia adecuada tras recargas de la página, lo que ocasionaba frustración para los usuarios. Con esta refactorización, se ha reducido el número de vulnerabilidades relacionadas con la autenticación y la

fuga de sesiones. También, se ha reducido la complejidad técnica al centralizar la lógica de permisos.

3. Gráficas de Reportes:

- **Refactorización realizada:** Se implementaron refactorizaciones en las gráficas que muestran la cantidad de órdenes y el dinero generado por día. Se mejoró la lógica de manejo de datos en el frontend para asegurarse de que los datos financieros se calculan correctamente. También se realizó una mejor integración entre el frontend y el backend para reducir errores en la obtención de datos.
- **Impacto en la deuda técnica:** El manejo de datos en los gráficos estaba fragmentado y carecía de escalabilidad. Con la refactorización, se asegura que el gráfico ahora está bien estructurado, con una conexión más robusta a la base de datos, lo que previene inconsistencias en los datos y errores en futuras ampliaciones.

4. Estilos CSS Globales (UserProfile.css, Login.css):

- **Refactorización realizada:** Se optimizaron los estilos CSS, aplicando principios de DRY (Don't Repeat Yourself), lo que permitió centralizar las reglas de estilo repetitivas en varias partes del sistema. Además, se ajustaron los estilos para mejorar la experiencia de usuario y garantizar la consistencia visual en todos los dispositivos.
- **Impacto en la deuda técnica:** El código CSS previo era fragmentado, lo que dificultaba la mantenibilidad. Con esta refactorización, el código CSS es más modular y fácil de modificar o escalar. Esto disminuye la deuda técnica, ya que futuros cambios visuales serán más fáciles de implementar sin generar nuevas inconsistencias.

3. Contribución a la Reducción de la Deuda Técnica

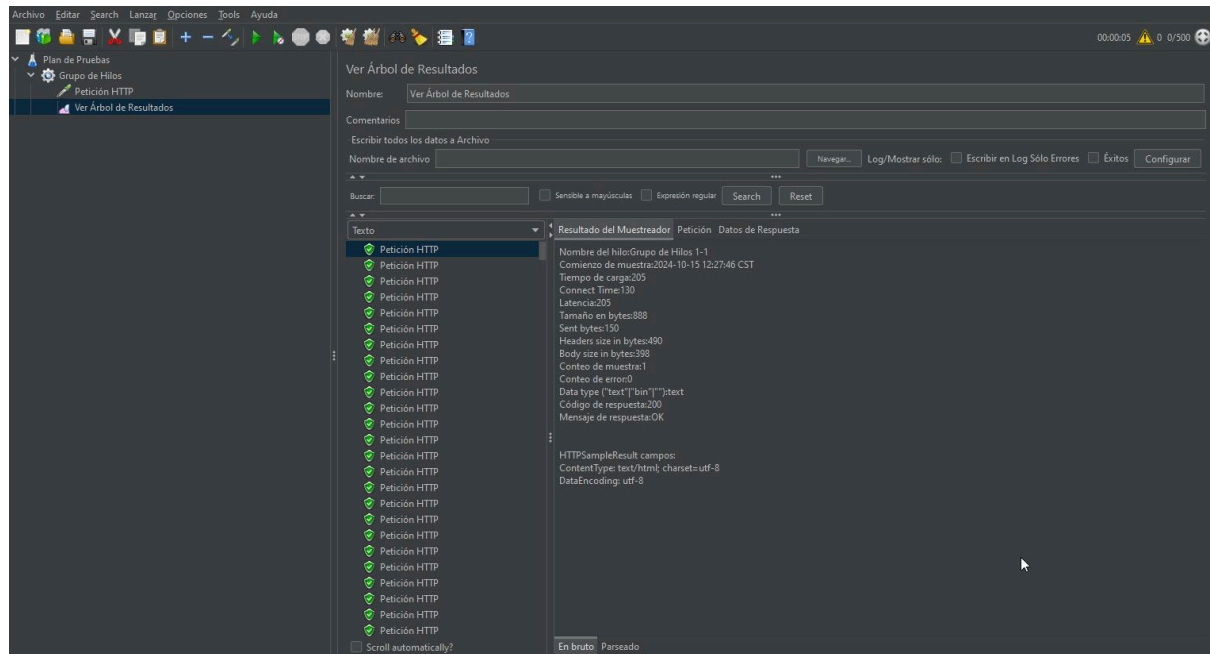
- **Mantenibilidad:** Al centralizar la lógica de autenticación, permisos, y el manejo de inventarios, se ha reducido la complejidad del código. Esto facilita futuras modificaciones y expansiones sin introducir errores.
- **Seguridad:** Mejoras en la autenticación y la gestión de permisos han reducido los riesgos de seguridad relacionados con accesos no autorizados. La lógica ahora es más robusta y menos propensa a errores.
- **Consistencia visual:** Refactorizar el CSS y aplicar patrones consistentes asegura que la experiencia de usuario sea fluida y profesional, lo que reduce la necesidad de futuras correcciones visuales a medida que se expanden las funcionalidades.
- **Escalabilidad:** Las mejoras en el manejo de datos (por ejemplo, en los gráficos) y la interacción entre el frontend y el backend hacen que la aplicación sea más escalable, ya que el código es más modular y reutilizable.

4. Próximos Pasos

En los próximos sprints, se continuará con la refactorización de áreas críticas que aún presentan deuda técnica. Esto incluye mejorar la gestión de errores en el frontend, optimizar

el rendimiento de las consultas a la base de datos, y fortalecer las medidas de seguridad en áreas clave, como el manejo de formularios y la validación de entradas de usuario.

Pruebas de volumen



Número Total de Muestras:

- **Valor:** 500 solicitudes
- Indica el número total de peticiones HTTP realizadas durante la prueba.

Tiempo Promedio de Respuesta (Media):

- **Valor:** 212 ms
- Representa el tiempo promedio que tarda el servidor en procesar cada solicitud.

Tiempo Mínimo de Respuesta:

- **Valor:** 185 ms
- Indica el tiempo más rápido de respuesta registrado en las peticiones.

Tiempo Máximo de Respuesta:

- **Valor:** 1090 ms
- Muestra el tiempo más largo que tomó una solicitud en completarse.

Desviación Estándar:

- **Valor:** 55.8 ms
- Mide la variación en los tiempos de respuesta. Un valor más bajo indica respuestas más consistentes, mientras que uno alto refleja una mayor dispersión en los tiempos de respuesta.

Porcentaje de Errores (% Error):

- **Valor:** 0%
- Todas las solicitudes fueron exitosas, lo que indica que no se detectaron errores durante la prueba.

Rendimiento (Throughput):

- **Valor:** 96.1 solicitudes/segundo
- Refleja cuántas solicitudes se manejan por segundo, lo que es una métrica importante para evaluar la capacidad del sistema bajo carga.

Datos Recibidos por Segundo (Kb/sec):

- **Valor:** 83.34 KB/seg
- Indica la cantidad de datos que el servidor envió por segundo durante la prueba.

Datos Enviados por Segundo (Sent KB/sec):

- **Valor:** 14.08 KB/seg
- Muestra la cantidad de datos que fueron enviados al servidor por segundo.

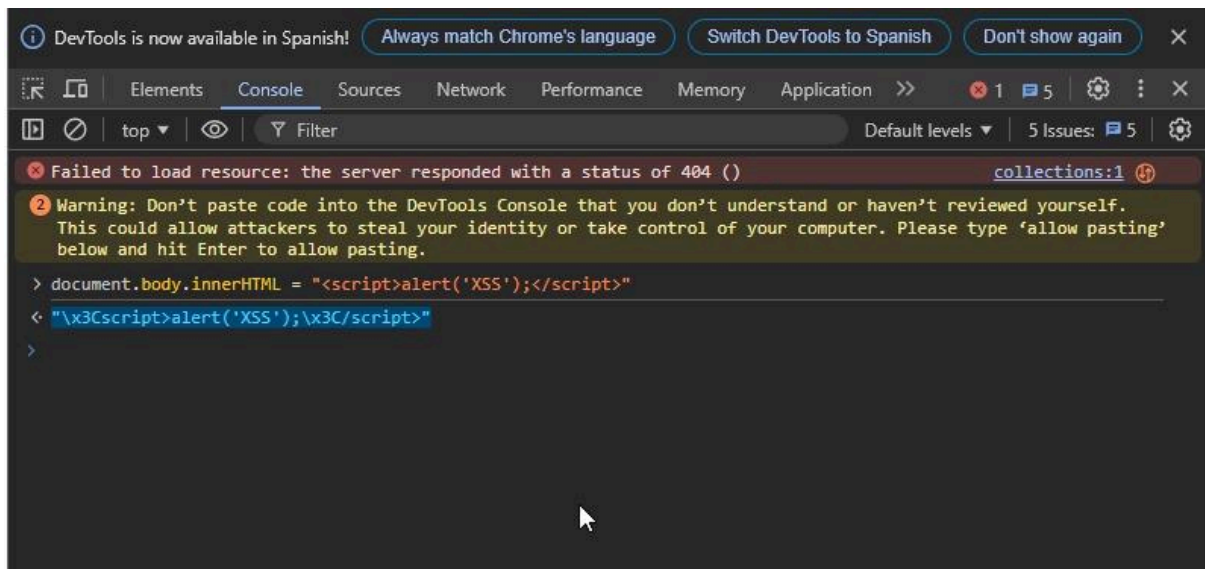
Tamaño Medio de la Respuesta (Media de Bytes):

- **Valor:** 888 bytes
- Representa el tamaño promedio de los datos recibidos por cada respuesta del servidor.

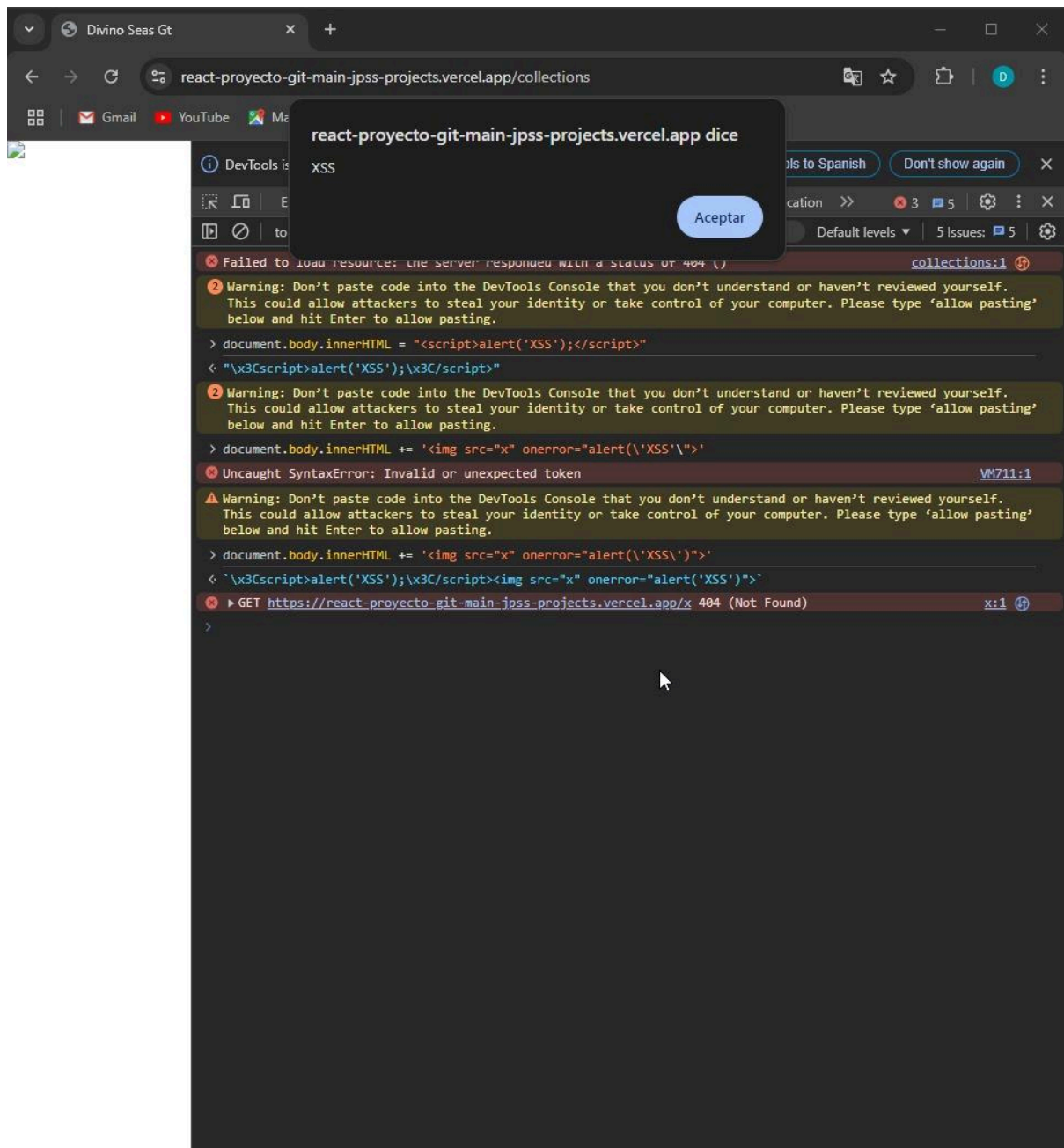
Conclusiones de las Métricas de Estrés:

- **Desempeño Consistente:** Con un tiempo de respuesta promedio de 212 ms y una desviación estándar relativamente baja (55.8 ms), las respuestas del servidor son bastante consistentes bajo una carga de 500 solicitudes.
- **Capacidad Adecuada:** El rendimiento de 96.1 solicitudes por segundo es bastante bueno, lo que sugiere que el servidor puede manejar la carga de manera eficiente sin degradación del rendimiento.
- **Sin Errores:** El hecho de que no se hayan producido errores (0% de errores) indica que el servidor manejó todas las solicitudes correctamente, lo que es un buen indicador de estabilidad.
- **Uso de Red:** Los datos enviados y recibidos por segundo (14.08 KB/seg enviados y 83.34 KB/seg recibidos) son indicadores útiles para medir el impacto en la red.

Pruebas de seguridad



Al inyectar un script, el sistema sanitiza correctamente los datos, escapando los caracteres especiales y evitando que el código malicioso se ejecute.



Sin embargo, al realizar la inyección a través de una imagen, el mensaje de alerta aparece, lo que indica que el frontend es vulnerable a este tipo de ataques de inyección.



Lo mismo sucede cuando se hace con el body.

Anexo

LOGT Joaquin Campos:
22155

Fecha	Objetivo	General	Inicio	Fin	Tiempo Interrupción	Delta tiempo	Fase	Comentarios
31/10/24	Reporte y logica de datos	Reporte y logica de datos	5:00PM	5:30 PM	0 min	30min	Sprint 6	Reporte y logica de datos
31/10/24	Acceso a administracion de usuario	Acceso a administracion de usuario	11:30 PM	12:30AM	0	1:50 min	Sprint 6	Acceso a administracion de usuario
2/10/24	Rutas	Rutas	11:30 AM	2 PM	30min	2hrs	Sprint 6	.N/A
4/10/24	Administracion de usuarios	Administracion de usuarios	12:40 am	1:10am	0	40min	Sprint 6	Administracion de usuarios
5/10/24	diseños administracion de usuarios	diseños administracion de usuarios	5:00pm	7:00pm	30	1:30 min	Sprint 6	diseños administracion de usuarios

5/10	Administración de usuarios final	Administración de usuarios final	3PM	5PM	0	2hrs	Sprint 6	Administración de usuarios final
15/10	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
10/10/24	Creación de hooks	Ahora se puede conectar la API	5:00PM	5:30 PM	0 min	30min	Sprint 9	
10/10/24	Conexión de collections a API	La página collection s recibe de API	12:30PM	1:30AM	0	1:00 min	Sprint 9	
12/10/24	Conexión de ProductPage a API	La página product page recibe de API	12:30 AM	1:30 PM	0min	1hrs	Sprint 9	

LOGT: Julio García Salas

Carnet: 22076

ID	Descripción	Fecha Inicio	Fecha Fin	Tiempo Interrupción	Delta Tiempo
214	Cambio a userprofile.css	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	2 horas
215	Cambio a inventoryform	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	3 horas
216	Cambio a producto	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	2 horas
217	CSS guapo	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	1 horas
218	Se mejora el settings content para que se pueda cerrar sesión	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	2 horas
219	Cambios al inventoryform	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	2 horas
220	Cambios al login css	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	1 horas
221	Se cambia el inventoryform	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	3 horas
222	Se agrega más CSS para el login y el authcontext	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	2 horas

223	AuthContext para que funcione el F5	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	2 horas
224	Para poder editar productos	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	4 horas

LOGT: Sofía García

Carnet:22210

ID	Descripción	Fecha Inicio	Fecha Fin	Tiempo Interrupción	Delta Tiempo
225	Se agrega dinero por día en la gráfica	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	3 horas
226	Ahora aparece la cantidad de órdenes y se arregla el orden por día	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	2 horas
227	Arreglar el eje X de la gráfica por días	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	1 horas
228	Ahora hay nuevos clientes	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	2 horas
229	Cambio al contenido de inventoryform para poder editarlo	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	2 horas

230	Correcciones a report	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	1 horas
231	Se conecta el reporte con la base de datos	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	3 horas
232	Tabla de órdenes completa	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	3 horas
233	Se arregló lo del monto	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	2 horas
234	Cambio a orderslist	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	2 horas
235	Hook para órdenes	14 de Octubre de 2024 08:00	14 de Octubre de 2024 10:00	5 minutos	2 horas

LOGT: Juan Menéndez

Carnet:17444

Fecha Inicio	Fecha Fin	Tiempo Interrupción	Delta Tiempo	Fase	Comentarios
10/10/2024 1:00 PM	10/10/2024 02:00 PM	0	1 hora	Sprint 9	Automatización del sistema de actualización del carrito de compras

10/10/2024 2:30 PM	10/10/2024 3:30 PM	0	1 hora	Sprint 9	Arreglo en el estilo CSS del carrito de compras
11/10/2024 1:00 PM	11/10/2024 6:00 PM	0	2 horas	Sprint 9	Creación de sistema de adición de productos mediantes archivo JSON.
11/10/2024 1:00 PM	11/10/2024 2:00 PM	0	1 horas	Sprint 9	Arreglos al sistema en cuanto a actualización y guardado de variantes en archivo JSON
15/09/2024 2:00 PM	15/09/2024 4:00 PM	0	2 horas	Sprint 9	Testeo de funcionalidad, y solución de errores provocados por deploy de aplicación y dockerización