

**UNIVERSIDAD DEL VALLE DE GUATEMALA**

Ingeniería de Software 2, sección 30



*Excelencia que trasciende*

**DEL VALLE**  
GRUPO EDUCATIVO

## **QUINTO SPRINT**

Sofia Garcia - 22210

Julio Garcia Salas - 22076

Joaquin Campos - 22155

Juan Fernando Menéndez - 17444

Juan Pablo Solis - 22102

**GUATEMALA, Julio de 2024**

## Índice

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

## Resumen

En este quinto sprint, se completó el desarrollo del API, abarcando desde la creación de servicios y rutas hasta su integración final en el proyecto. Adicionalmente, se realizaron avances significativos en el frontend, principalmente en la implementación de mockups y la estructuración de componentes React.

## Introducción

El quinto sprint del proyecto de Ingeniería de Software 2, se enfocó en finalizar el API y avanzar en el frontend del sistema. Durante este periodo, el equipo se concentró en la creación y documentación de servicios y rutas del API, así como en la construcción y mejora de mockups de las interfaces de usuario. Estos esfuerzos están alineados con nuestro objetivo de crear una plataforma ecommerce robusta y funcional, optimizada para el mercado guatemalteco.

## 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 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 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	Mostrar cómo 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

35	Creación Componentes pagina Product	Media	2	Hacer los componentes necesario para la página de product	
36	Implementación componentes pagina Product	Media	1	Implementar los componentes creados en la pagina produc	
37	Edición pagina product con css	baja	1	Editar la página Product con css	
38	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 productCollectionService	Alta	3	Implementar CRUD en productCollectionService.js	Incluir conexión a base de datos y métodos CRUD.
45	Creación de productCollectionRoute	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	Alta	4	Sistema de navegación	Crear un menú



	navegación intuitivo			implementado.	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 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.
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	Media	3	Configuración de Jest para pruebas de React.	Mostrar pruebas básicas de

	para frontend				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 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	Media	3	Auditoría de accesibilidad	Asegurar

	accesibilidad web			completada y mejoras aplicadas.	cumplimiento con WCAG y mejorar la accesibilidad general.
11 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.
11 3	Monitoreo post-lanzamiento y recopilación de feedback	Alta	On goi ng	Informe de feedback inicial recopilado.	Establecer herramientas de monitoreo en tiempo real y canales de feedback.
11 4	Ajustes post-lanzamiento basados en feedback	Media	On goi ng	Ajustes implementados según feedback.	Priorizar y aplicar mejoras y correcciones rápidas post-lanzamiento.
11 5	Desarrollo de documentación para usuarios finales	Baja	4	Documentación de usuario disponible.	Crear guías de usuario, FAQs y documentación de ayuda.
11 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.
11 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.
11 8	Arreglo del navbar	Alta	2	Demostrar navbar completamente responsivo	Ajustes para mejorar la responsividad y el alineamiento del

					navbar.
11 9	Primera versión arreglo statbat	Media	1	Mostrar el statbar alineado y responsivo.	Cambios iniciales para mejorar la visualización y funcionalidad del statbar.
12 0	Borrar cosas innecesarias	Alta	3	Confirmación de eliminación de archivos innecesarios.	Limpieza de código para optimizar el proyecto.
12 1	Statbar cambios iniciales	Media	2	Mostrar versión inicial del statbar con mejoras aplicadas.	Cambios para mejorar la estructura y estilo del statbar.
12 2	Statbar responsive y alineado	baja	2	Demostrar la responsividad y alineación del statbar.	Ajustes para asegurar la correcta visualización en diferentes dispositivos.
12 3	Navbar full responsive	Media	1	Demostrar funcionalidad responsiva completa del navbar.	Modificaciones para mejorar la compatibilidad con diversos tamaños de pantalla.
12 4	Arreglitos x	Media	2	Mostrar pequeños ajustes r	Correcciones menores en el código para optimizar funcionalidades.
12 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.
12 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.
12 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.
12 8	Estilos de la lista de inventario	Alta	3	Mostrar los estilos aplicados	Diseño y aplicación de estilos para mejorar la interfaz de la lista de inventario.
12 9	Se agrega el salesbar	Alta	3	Demostrar la funcionalidad y estilos del salesbar.	Incorporación de un salesbar para mejorar la interfaz de ventas.
13 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.
13 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.
13 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.
13 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.
13 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.
13 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.
13 6	Crear shippingService	Alta	3	Mostrar shippingService.js implementando las operaciones CRUD	Incluir conexión a la base de datos y métodos CRUD
13 7	Crear shippingRoute	Alta	3	Mostrar shippingRoute.js con rutas de API para envíos	Definir endpoints para operaciones CRUD y middleware
13 8	Crear paymentService	Alta	3	Mostrar paymentService.js implementando las operaciones CRUD	Incluir conexión a la base de datos y métodos CRUD
13 9	Crear paymentRoute	Alta	3	Mostrar paymentRoute.js con rutas de API para pagos	Definir endpoints para operaciones CRUD y middleware
14 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
14 1	Iniciar dashboardPage	Alta	2	Mostrar progreso de la página del dashboard	Incluir elementos iniciales y estructura básica
14 2	Agregar checkbox para el chequeo de pedidos en salesPage	Alta	2	Mostrar checkbox funcionando	Confirmar que los pedidos pueden ser seleccionados correctamente
14 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
14	Convertir	Alta	2	Que se vea responsive	Que se vea

4	responsive la página web				responsive
14 5	Hacer la home page de la página de admin	Alta	2	Hacer la home page	Que se vea la home page

## Pila del sprint

1. ID: 136
  - Descripción: Crear shippingService
  - Puntos de Historia: 3
  - Horas: 5
  - Realizado por: Julio García Salas
  - Fecha: 22-24 de Julio de 2024
2. ID: 137
  - Descripción: Crear shippingRoute
  - Puntos de Historia: 3
  - Horas: 5
  - Realizado por: Julio García Salas
  - Fecha: 24-25 de Julio de 2024
3. ID: 138
  - Descripción: Crear paymentService
  - Puntos de Historia: 3
  - Horas: 5
  - Realizado por: Julio García Salas
  - Fecha: 25-26 de Julio de 2024
4. ID: 139
  - Descripción: Crear paymentRoute
  - Puntos de Historia: 3
  - Horas: 5
  - Realizado por: Julio García Salas
  - Fecha: 26-27 de Julio de 2024
5. ID: 140
  - Descripción: Agregar shippingService y paymentService al index.js
  - Puntos de Historia: 2
  - Horas: 3
  - Realizado por: Julio García Salas
  - Fecha: 27-28 de Julio de 2024
6. ID: 141
  - Descripción: Iniciar dashboardPage
  - Puntos de Historia: 2
  - Horas: 4
  - Realizado por: Julio García Salas
  - Fecha: 28-29 de Julio de 2024
7. ID: 145
  - Descripción: Convertir responsive la página de admin

- Puntos de Historia: 2
- Horas: 2
- Realizado por: Sofía García
- Fecha: 28-29 de Julio de 2024

8. ID: 143

- Descripción: Añadir nombres a las columnas de la tabla de ventas
- Puntos de Historia: 2
- Horas: 2
- Realizado por: Sofía García
- Fecha: 28-29 de Julio de 2024

9. ID: 144

- Descripción: Actualización links de Navbar
- Puntos de Historia: 2
- Horas: 1
- Realizado por: Juan Pablo Solis
- Fecha: 28-29 de Julio de 2024

10. ID: 145

- Descripción: Añadir Botón para scrollear hacia arriba
- Puntos de Historia: 1
- Horas: 1
- Realizado por: Juan Pablo Solis
- Fecha: 25-26 de Julio de 2024

11. ID: 146

- Descripción: Añadir Animaciones a la pestaña de filtros de Collections
- Puntos de Historia: 2
- Horas: 3
- Realizado por: Juan Pablo Solis
- Fecha: 27-28 de Julio de 2024

12. ID: 147

- Descripción: Enrutamiento dinámico para pagina Product
- Puntos de Historia: 3
- Horas: 3
- Realizado por: Juan Pablo Solis
- Fecha: 28-29 de Julio de 2024

13. ID: 148

- Descripción: Enrutamiento dinámico product Cards
- Puntos de Historia: 2
- Horas: 2
- Realizado por: Juan Pablo Solis
- Fecha: 28-29 de Julio de 2024

14. ID: 149

- Descripción: Funcionalidades en botones salesbar
- Puntos de Historia: 3

- Horas: 3
- Realizado por: Jose Joaquin Campos
- Fecha: 27-28 de Julio de 2024

15. ID: 150

- Descripción: Creación Inventory Form
- Puntos de Historia: 2
- Horas: 2
- Realizado por: Jose Joaquin Campos
- Fecha: 28-29 de Julio de 2024

16. ID: 151

- Descripción: Revalidación si las funcionalidades están completas y eficientes en relación a la base de datos
- Puntos de Historia: 3
- Horas: 3
- Realizado por: Jose Joaquin Campos
- Fecha: 29 de Julio de 2024

17. ID: 152

- Descripción: Búsqueda de inventario
- Puntos de Historia: 2
- Horas: 2
- Realizado por: Jose Joaquin Campos
- Fecha: 28-29 de Julio de 2024

18. ID 153

- Descripción: Hacer home page
- Puntos de Historia: 2
- Horas: 2
- Realizado por: Sofía García
- Fecha: 28-29 de Julio de 2024

19. ID 154

- Descripción: Arreglar componentes para la página de pago
- Puntos de Historia: 2
- Horas: 1
- Realizado por: Juan Menéndez
- Fecha: 28-29 de Julio de 2024

20. ID 155

- Descripción: Arreglar error en el merge
- Puntos de Historia: 3
- Horas: 2
- Realizado por: Juan Menéndez
- Fecha: 28-29 de Julio de 2024

21. ID 156

- Descripción: Mejora del archivo css de la página de entrega
- Puntos de Historia: 2
- Horas: 2
- Realizado por: Juan Menéndez
- Fecha: 28-29 de Julio de 2024

22. ID 157

- Descripción: Enrutamiento de producto y pago (función temporal)

- Puntos de Historia: 2
- Horas: 1
- Realizado por: Juan Menéndez
- Fecha: 28-29 de Julio de 2024

### 23. ID 158

- Descripción: Traducción de footer
- Puntos de Historia: 1
- Horas: 1
- Realizado por: Juan Menéndez
- Fecha: 28-29 de Julio de 2024

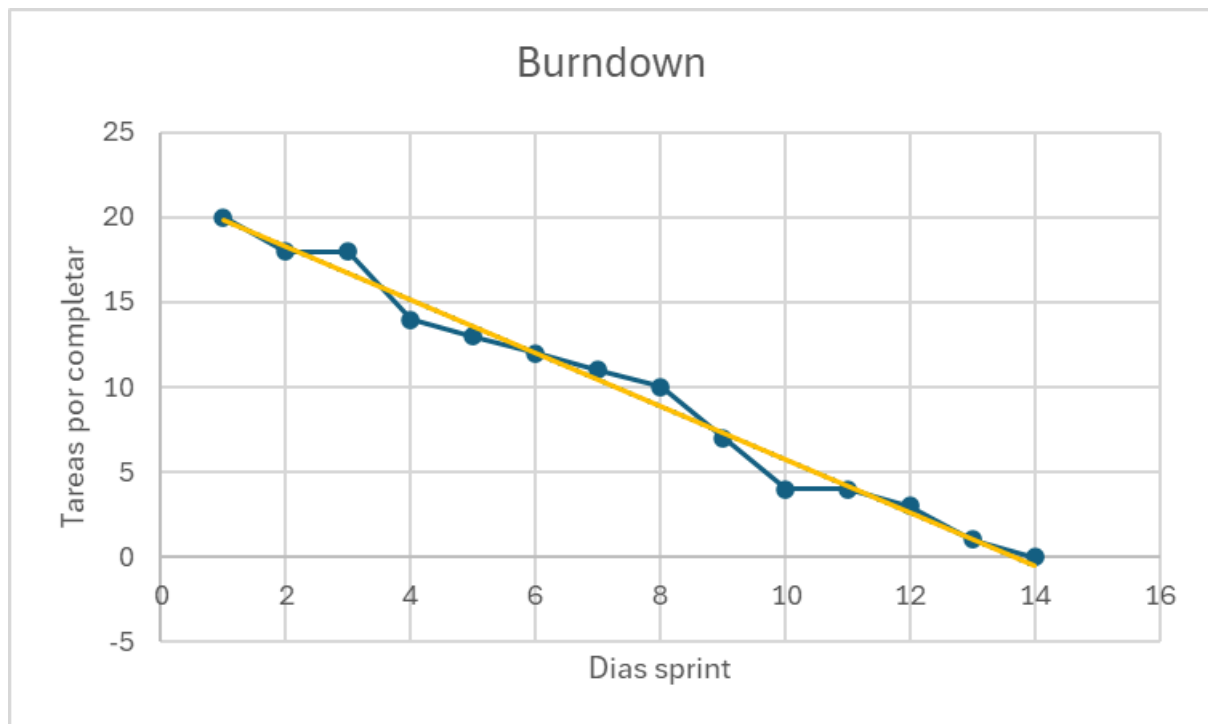
## Resultados del sprint

Tablero de jira:

<https://uvg-team-g8inmhj5.atlassian.net/jira/software/projects/SCRUM/boards/1?atlOrigin=eyJpIjoiNDIzMzA1NzA3ZTFkNDIkJzhlYzliMjkxYzk3ZTE3ODMiLCJwIjoiajJ9>

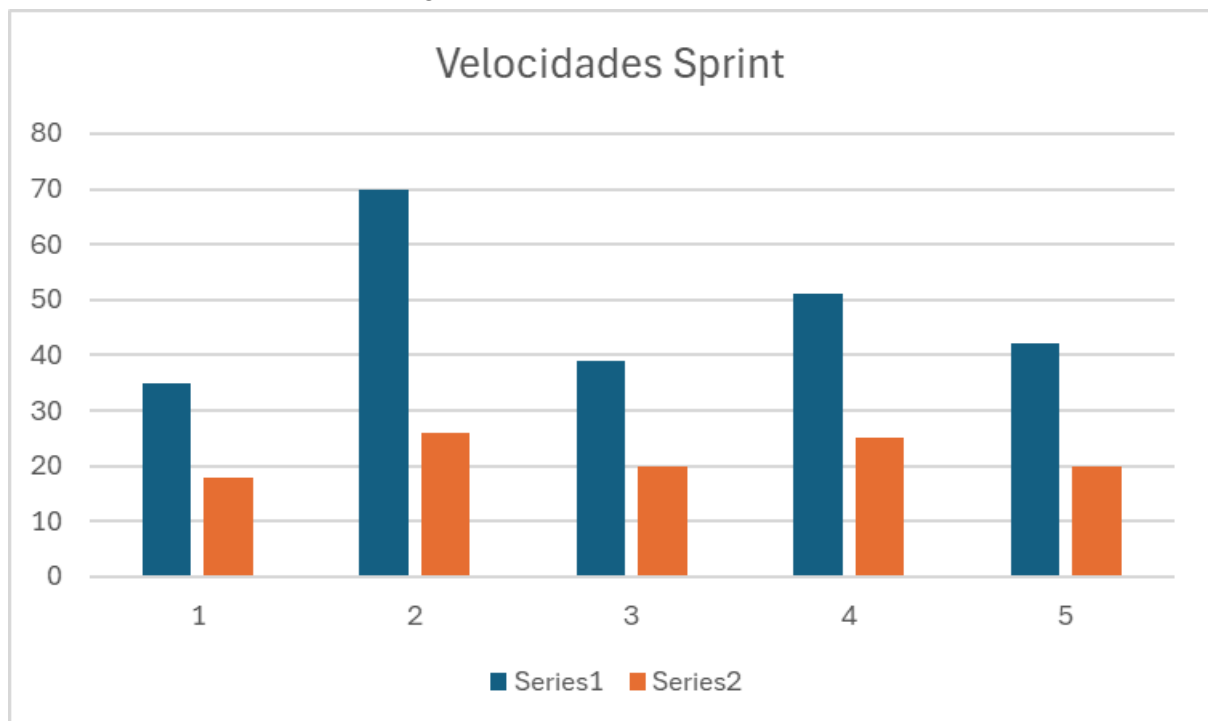
Se lograron hacer todas las tareas planificadas para este sprint,y en cambio con el último sprint ya se logró colocar un mejor margen entre tareas asignadas y tiempo.

## Métricas del sprint



## Velocidad del Sprint

Comparación de velocidades según sprints



- Se concluyó el sprint con un puntaje de 42 puntos completando las 20 tareas asignadas previamente para el sprint.

- En cuanto a velocidad media del Scrum total (47.4) podemos decir que la velocidad que ya nos estamos acercando al promedio de manera que no está de más darle valor al hecho que las correlación entre velocidades ya esta agarrando forma lineal de manera que estamos encontrando el punto de inflexión entre eficiencia y carga, cerca de encontrar lo ideal.

## Discusión

### Cumplimiento de Objetivos

- Positivo: El equipo completó las 20 tareas asignadas, lo cual indica que se cumplieron los objetivos del sprint en términos de volumen de trabajo.
- Para mejorar: La distribución de la carga de trabajo a lo largo del sprint fue un poco más considerable que en sprints pasados ya que normalmente se dejaba para de último, no obstante todavía hay áreas de mejora.

### Gestión del Tiempo y Distribución del Trabajo

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

### Calidad del Trabajo

- A considerar: Aunque el gráfico muestra que se completaron todas las tareas, la calidad de las tareas completadas también es importante, lo que siempre habrá área de mejora, no obstante estamos acercándonos a tener el proyecto completado ahora ya con el esqueleto casi completo es satisfactorio poder presentar cosas al PO.

### Comunicación y Colaboración en el Equipo

- Positivo: La finalización exitosa de las tareas sugiere una buena colaboración dentro del equipo, especialmente en la fase de alta presión hacia el final.
- Para mejorar: Sería útil analizar si la comunicación es efectiva dentro del equipo.

### Sostenibilidad del Esfuerzo

- Preocupación: La cantidad de tareas en relación a la cantidad de días involucrados se puede ver un poco reducida de manera que se intentará en el futuro agregar mejores y más tareas

### Aprendizajes y Mejoras para Sprints Futuros

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

## Código desarrollado

[https://github.com/JPS4321/React\\_Proyecto](https://github.com/JPS4321/React_Proyecto)

Lista de funcionalidades terminadas:

- Creación de títulos en la tabla de ordenes admin
- Modificación de tabla de ordenes admin para ser responsive
- Creación de home page de página admin
- Creación de services de shipping API
- Creación de services de payment API
- Enrutamiento de Shipping API
- Enrutamiento de Payment API
- Se añadió el index a API
- Creación de dashboard page
- Búsqueda de inventario
- Revalidación si las funcionalidades están completas y eficientes
- Creación Inventory Form
- Funcionalidades en botones salesbar
- Enrutamiento dinámico product Cards
- Enrutamiento dinámico para pagina Product
- Añadir Animaciones a la pestaña de filtros de Collections
- Añadir Botón para scrollear automáticamente hacia arriba
- Actualización links de Navbar

## Presupuesto

### Se calcula el presupuesto

#### Fase Actual: 4 Sprints completados

- Duración de un Sprint: Asumamos que cada sprint dura 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 cuarto sprint:

- $\$3,500 \times 4 = \$14,000$ .

#### Estimación para el quinto sprint

- Duración restante para el quinto sprint: 1 semana y 4 días (11 días).
- Costo por programador por día:  $\$1,400 / 30 \approx \$46.67$ .
- Costo por programador para 11 días:  $\$46.67 \times 11 \approx \$513.37$ .
- Costo total por 5 programadores para 11 días:  $\$513.37 \times 5 \approx \$2,566.85$ .

#### Costo total hasta el final del quinto sprint:

- $\$14,000 + \$2,566.85 \approx \$16,566.85$ .

### Se calcula el presupuesto para la ejecución total del proyecto

#### Asumir progreso futuro

- Total de sprints: 10 sprints.
- Sprints completados: 4.57 sprints (incluyendo 1 semana y 4 días del quinto sprint).



- Sprints restantes:  $10 - 4.57 = 5.43$  sprints.

#### Cálculo de sprints restantes

- Costo por sprint para 5 programadores: \$3,500.
- Costo total por 5.43 sprints adicionales:  $\$3,500 \times 5.43 = \$19,005$ .

#### Presupuesto Total Estimado

- Costo hasta el final del quinto sprint: \$16,566.85.
- Costo para sprints adicionales: \$19,005.
- Presupuesto total estimado:  $\$16,566.85 + \$19,005 \approx \$35,571.85$ .

#### Resumen

- Presupuesto hasta la fecha (4 sprints completados + 1 semana y 4 días del quinto sprint): \$16,566.85.
- Presupuesto para la ejecución total del proyecto: \$35,571.85.

#### Documentación

- Retrospectiva del sprint:

De lo que se puede rescatar del sprint es que hay bastante área de mejora, desde aumentar la cantidad de tareas, ya que actualmente la eficiencia y correlación de trabajo está bastante bien dispersa no obstante falta dedicación aun ya que prácticamente 4 personas hicieron 5 tareas, faltando un último 5 integrante.

#### Github de archivos:

<https://github.com/Hayser8/DivinoSeasIngSoft>

#### Anexo

LOGT Joaquin Campos:  
22155



Fecha	Objetivo	General	Inicio	Fin	Tiempo Interrupción	Delta tiempo	Fase	Comentarios
25/07/24	Funcionalidades en botones salesbar	Considerar el funcionamiento de los botones establecidos en salesbar	5:00PM	5:30 PM	0 min	30min	Sprint 5	Funcionalidades en botones salesbar. Considerar el funcionamiento de los botones establecidos

								s en salesbar.
26/07/24	Creación Inventory Form	Creación de componentes react inventoryForm	11:30 PM	12:30 AM	0	1:50 min	Sprint 5	Se creo el componente InventoryForm de la aplicación futura de los administradores
27/7/24	Revalidación si las funcionalidades	están completas y eficientes	11:30 AM	2 PM	30min	2hrs	Sprint 5	Revalidacion funcionalidades de que la base de datos tenga relación con todas las funciones estaremos dándole acceso al usuario.
28/07/24	Gráfico burndown	Evaluación de las tareas que teníamos planeadas	12:40 am	1:10am	0	40min	Sprint 5	Se creo la gráfica con la herramienta Excel
28/07/24	Métrica de velocidad	Puntos de historia por las 14 tareas	5:00pm	7:00pm	30	1:30 min	Sprint 5	Se sumaron los puntos de historia para obtener la métrica de velocidad
30/7	Búsqueda de inventario	Búsqueda de inventario para hacer funcional por texto	3PM	5PM	0	2hrs	Sprint 5	Se le agrego la función del texto para que este pueda buscarse a letra.