



Nombre del Proyecto: **Backstore**

Entrega Fase 2 Avance del proyecto

Fecha Entrega: **17 de Octubre del 2024**

Docente: **Willy Bascuñán**

Integrantes:

**Juan Pablo Fuentes Vivanco**

**Alonso González Gómez**

**Paulo Sandoval Retamal**

**Tabla de contenido**

[1. Resumen avance Proyecto APT 3](#_heading=h.gjdgxs)

[1.1. Resumen de avance proyecto APT 3](#_heading=h.30j0zll)

[1.2. Objetivos 3](#_heading=h.1fob9te)

[1.3. Evidencias de avance 4](#_heading=h.3znysh7)

[2. Monitoreo del Plan de Trabajo 12](#_heading=h.2et92p0)

[3. Ajustes a partir del monitoreo 13](#_heading=h.3j8qtaxb9laq)

[3.1 Factores que han facilitado y/o dificultado el desarrollo de mi plan de trabajo 13](#_heading=h.3dy6vkm)

[3.2 Actividades ajustadas o eliminadas 15](#_heading=h.1t3h5sf)

[3.3 Actividades que no has iniciado o están retrasadas: 15](#_heading=h.4d34og8)

# Resumen avance Proyecto APT

## Resumen de avance proyecto APT

El proyecto Backstore ha avanzado significativamente desde su inicio, logrando implementar las principales funcionalidades de gestión de pedidos y asignación de tareas a los pickers para las tiendas de retail de Corona. Hasta la fecha, se ha completado el desarrollo del backend utilizando Node.js con NestJS y una base de datos MongoDB, y se ha avanzado con la creación del frontend utilizando React.js y Material UI. Se han implementado funcionalidades claves, como el sistema de autenticación mediante JWT para garantizar la seguridad, y el menú dinámico que se adapta al rol del usuario (Administrador Global, Administrador de Tienda, Picker). También se logró compilar la aplicación android para el usuario Picker, de esta manera, el usuario es capaz de poder realizar la recolección de productos dentro del a tienda.

Además, se han realizado pruebas unitarias para validar la correcta operación de los módulos individuales, pruebas de integración para verificar la comunicación entre los componentes, y pruebas funcionales para asegurar el cumplimiento de los requerimientos del sistema. También se han llevado a cabo reuniones de Scrum de forma regular para ajustar y coordinar el desarrollo del proyecto, facilitando el cumplimiento de los objetivos propuestos.

## Objetivos

Se realizaron algunos ajustes en los objetivos para realizar un enfoque más en la eficiencia de la comunicación entre los usuarios del sistema, lo cual se detectó como un punto crítico durante la fase de análisis de requerimientos. Es por esto que el proyecto se ha enfocado en poder mostrar un MVP al usuario, avanzando con módulos de gestión como gestión de usuarios y sucursales.

Para lograr esto se priorizaron otras historias de usuario y de esta forma se llegó al producto inicial. Se dejaron para el final los dos módulos; Gestión de órdenes de transporte y Gestión de entregas.

## Evidencias de avance

* ***Evidencias mínimas requeridas****:*

**Épicas e Historias de Usuario:**

Se adjunta documentación en la carpeta de anexos.

**Matriz módulo o funciones del producto versus Historias de usuario:**

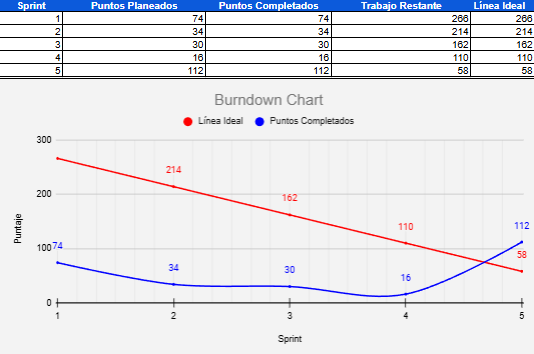
Se adjunta documentación en la carpeta de anexos.

**Product Backlog**

Las filas marcadas en color, corresponden a las HU realizadas por el equipo, además de contener los **Sprint Backlog.**

| **Sprint** | **Historia de Usuario** | **Prioridad** | **Estimación** | **Tareas** |
| --- | --- | --- | --- | --- |
| 1 | Revisión del Prototipo y Ajustes | Alta | 5 | Revisar los prototipos de las vistas principales de la app web y móvil |
| 1 | Revisión del Prototipo y Ajustes | Alta | 5 | Hacer ajustes en los wireframes y mockups según el feedback recibido |
| 1 | Preparación del Entorno de Desarrollo | Alta | 8 | Configurar el repositorio de código en Git |
| 1 | Preparación del Entorno de Desarrollo | Alta | 8 | Configurar el pipeline de CI/CD en Azure DevOps |
| 1 | Preparación del Entorno de Desarrollo | Alta | 8 | Asegurar la integración con JIRA para el seguimiento de tareas |
| 1 | Diseño Lógico y Diagramas del Sistema (4+1) | Alta | 10 | Crear diagramas de casos de uso para cada módulo |
| 1 | Diseño Lógico y Diagramas del Sistema (4+1) | Alta | 10 | Diseñar los diagramas de clases y objetos |
| 1 | Diseño Lógico y Diagramas del Sistema (4+1) | Alta | 10 | Crear diagramas de despliegue y componentes |
| 1 | Diseño Lógico y Diagramas del Sistema (4+1) | Alta | 10 | Validar el diseño con el equipo de desarrollo y stakeholders |
| 2 | Diseño de la API del Picking | Alta | 5 | Diseñar los endpoints necesarios para el picking |
| 2 | Diseño de la API del Picking | Alta | 5 | Definir los esquemas de datos y validaciones |
| 2 | Implementación del Backend para Picking | Alta | 8 | Implementar los endpoints en Node.js con NestJS |
| 2 | Implementación del Backend para Picking | Alta | 8 | Implementar la lógica para asignación y gestión de órdenes de picking |
| 2 | Desarrollo del Frontend (React) para Picking | Alta | 8 | Crear la interfaz para listar las órdenes de picking |
| 3 | Desarrollo del Frontend (React) para Picking | Alta | 8 | Implementar la funcionalidad de marcar productos como recogidos |
| 3 | Desarrollo del Frontend (React) para Picking | Alta | 8 | Validar que el estado de las órdenes se actualice correctamente |
| 3 | Implementación del Backend para Gestión de Usuarios | Alta | 7 | Implementar endpoints para gestionar usuarios crear, editar, eliminar |
| 3 | Implementación del Backend para Gestión de Usuarios | Alta | 7 | Implementar la lógica de roles y permisos en el backend |
| 4 | Desarrollo del Frontend para Gestión de Usuarios | Alta | 8 | Crear la vista de gestión de usuarios en el frontend |
| 4 | Desarrollo del Frontend para Gestión de Usuarios | Alta | 8 | Implementar la funcionalidad de asignar roles a los usuarios desde la interfaz |
| 5 | Desarrollo del Frontend para Gestión de Usuarios | Alta | 8 | Validar que los cambios se reflejen en tiempo real |
| 5 | Desarrollo del Frontend para Gestión de Sucursales | Alta | 8 | Crear la vista de gestión de sucursales en el frontend |
| 5 | Desarrollo del Frontend para Gestión de Sucursales | Alta | 8 | Implementar la funcionalidad de gestionar sucursales en BackEnd |
| 5 | Desarrollo del Frontend para Gestión de Sucursales | Alta | 8 | Validar que los cambios se reflejen en tiempo real |
| 5 | Desarrollo del Frontend (Flutter) para Picking (Móvil) | Alta | 8 | Implementar la funcionalidad para marcar productos como recogidos |
| 5 | Desarrollo del Frontend (Flutter) para Picking (Móvil) | Alta | 8 | Validar la sincronización en tiempo real con el backend |
| 5 | Integración del Módulo de Picking (Móvil) con el Backend | Alta | 7 | Implementar la conexión entre el módulo de picking móvil y el backend |
| 5 | Integración del Módulo de Picking (Móvil) con el Backend | Alta | 7 | Validar la correcta actualización de los datos en la base de datos MongoDB |
| 5 | Pruebas Unitarias del Módulo de Picking (Móvil) | Media | 5 | Implementar pruebas unitarias para el módulo de picking en Flutter |
| 5 | Pruebas Unitarias del Módulo de Picking (Móvil) | Media | 5 | Ejecutar pruebas de integración con el backend |
| 5 | Pruebas de Integración Completa (Web y Móvil) | Alta | 8 | Realizar pruebas de integración entre los módulos web y móvil |
| 5 | Pruebas de Integración Completa (Web y Móvil) | Alta | 8 | Validar la integración entre el backend y los sistemas externos inventario y pedidos |
| 5 | Pruebas de Integración Completa (Web y Móvil) | Alta | 8 | Corregir cualquier error de integración que surja |
| 5 | Pruebas de Carga y Rendimiento | Alta | 8 | Ejecutar pruebas de carga simulando múltiples usuarios en la app web y móvil |
| 5 | Pruebas de Carga y Rendimiento | Alta | 8 | Identificar problemas de rendimiento y optimizar el código |
| 6 | Despliegue en Entorno de Preproducción | Alta | 6 | Preparar el entorno de preproducción en Azure |
| 6 | Despliegue en Entorno de Preproducción | Alta | 6 | Desplegar la aplicación web y móvil en el entorno de preproducción |
| 6 | Pruebas Finales en Preproducción | Alta | 5 | Ejecutar pruebas funcionales y de rendimiento en preproducción |
| 6 | Pruebas Finales en Preproducción | Alta | 5 | Corregir cualquier error crítico detectado en preproducción |
| 6 | Despliegue en Producción | Alta | 6 | Desplegar la aplicación web y móvil en producción |
| 6 | Despliegue en Producción | Alta | 6 | Validar que el despliegue en producción se haya realizado correctamente |
| 6 | Diseño de la API de Entregas | Alta | 5 | Definir los endpoints necesarios para gestionar las entregas |
| 6 | Diseño de la API de Entregas | Alta | 5 | Crear el esquema de datos de las entregas en MongoDB |
| 7 | Implementación del Backend para Entregas | Alta | 8 | Implementar los endpoints de gestión de entregas |
| 7 | Implementación del Backend para Entregas | Alta | 8 | Crear la lógica para el estado de las órdenes de entrega |
| 7 | Desarrollo del Frontend (React) para Entregas | Alta | 8 | Desarrollar la vista para gestionar las órdenes de entrega |
| 7 | Desarrollo del Frontend (React) para Entregas | Alta | 8 | Implementar la funcionalidad para cambiar el estado de las entregas |
| 7 | Pruebas Unitarias y de Integración del Módulo Entregas | Media | 5 | Implementar pruebas unitarias para los endpoints de entregas |
| 7 | Pruebas Unitarias y de Integración del Módulo Entregas | Media | 5 | Realizar pruebas de integración con el frontend |
| 8 | Diseño de la API de Órdenes de Transporte | Alta | 5 | Diseñar los endpoints para gestionar el transporte desde la tienda |
| 8 | Diseño de la API de Órdenes de Transporte | Alta | 5 | Definir el esquema de datos en MongoDB |
| 8 | Implementación del Backend para Órdenes de Transporte | Alta | 8 | Crear los endpoints necesarios para las órdenes de transporte |
| 8 | Implementación del Backend para Órdenes de Transporte | Alta | 8 | Implementar la lógica para cambiar el estado de transporte |
| 9 | Desarrollo del Frontend (React) para Órdenes de Transporte | Alta | 8 | Desarrollar la vista para gestionar las órdenes de transporte |
| 9 | Desarrollo del Frontend (React) para Órdenes de Transporte | Alta | 8 | Implementar la funcionalidad para actualizar el estado del transporte |
| 9 | Pruebas Funcionales y de Integración del Módulo Transporte | Media | 5 | Implementar pruebas funcionales para el backend de transporte |
| 9 | Pruebas Funcionales y de Integración del Módulo Transporte | Media | 5 | Realizar pruebas de integración con el frontend |

**Burndown Chart**

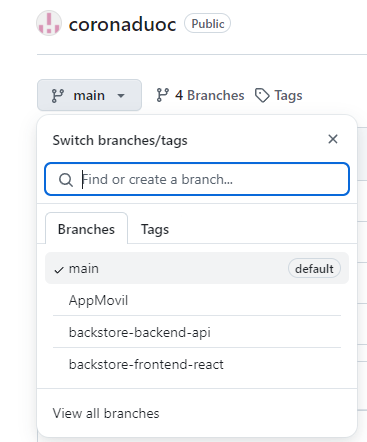
El gráfico BurndownChar representa los cinco primeros sprint realizados por el equipo. ****

**Avances en el desarrollo**: Entrega del código fuente web y/o aplicación móvil. Script base de datos. A continuación, el detalle y ubicación del código fuente de cada uno de los componentes:

Github: <https://github.com/alonsillo88/coronaduoc>

**Ramas:**

* + FrontEnd React (Web): **AppMovil**
  + FrontEnd Flutter (Mobile): **backstore-backend-react**
  + Backend NodeJS: **backstore-backend-api**



**Implementación de ambientes:**

Se adjunta evidencia por ambiente.

**Pruebas realizadas**

Se adjunta evidencia por ambiente.

**Capturas de pantalla y logs**

Se adjunta evidencia por ambiente.

**Comparación con la industria**

Gestión Ágil con Scrum: En la mayoría de los proyectos de desarrollo software de la industria, se utilizan metodologías ágiles, como Scrum, para gestionar de forma efectiva el trabajo en equipo y adaptarse rápidamente a los cambios. En el desarrollo de Backstore, la planificación mediante sprints, la constante revisión de objetivos, y el uso de herramientas como Jira para la gestión de tareas, aseguraron que el proyecto avanzara conforme a los requisitos. Estas prácticas reflejan el enfoque ágil predominante en la industria, donde la capacidad de adaptación y la entrega continua de valor son fundamentales.

Seguridad con Estándares de Autenticación: La implementación de JWT (JSON Web Tokens) para la autenticación de usuarios cumple con las mejores prácticas de seguridad de la industria. JWT se usa ampliamente en la autenticación de aplicaciones debido a su capacidad de garantizar una transferencia segura de datos entre el cliente y el servidor. La correcta implementación de este tipo de autenticación garantiza que solo los usuarios autorizados puedan acceder al sistema, protegiendo la información sensible de los usuarios y las operaciones internas. Esta práctica es un estándar adoptado por grandes organizaciones que buscan asegurar la integridad y privacidad de los datos.

Desarrollo Modular y Mantenible: El diseño de la arquitectura del sistema, utilizando NestJS para el backend y React.js con Material UI para el frontend, sigue principios de modularidad y reutilización de componentes, que son esenciales en el desarrollo moderno. La separación de responsabilidades y el diseño orientado a componentes permiten una fácil mantenibilidad y actualización del sistema, alineándose con la práctica estándar de la industria de desarrollar software que pueda evolucionar con los cambios de negocio y tecnologías emergentes.

Experiencia de Usuario (UX): La iteración del diseño de la interfaz con Material UI y la validación constante con los usuarios también cumplen con estándares de la industria, donde la experiencia del usuario (UX) es prioritaria para asegurar la adopción y satisfacción del cliente. El feedback continuo obtenido de usuarios simulados permitió mejorar la interfaz, haciendo que el sistema sea intuitivo y fácil de usar, algo que es crucial en plataformas de gestión de pedidos para retail.

**Historial de commits con capturas del uso de ramas y merges.**

Se adjunta anexo con la imagen solicitada.

**Diagramas:**

Se adjunta carpeta de diagramas solicitados.

**Video :**

Se adjuntan videos con demostración del front end web y la App mobile en la carpeta de fase 2 de este entregable.

# Monitoreo del Plan de Trabajo

**Se adjunta anexo Tabla de competencias.xlsx en la carpeta de entrega.**

# 3. Ajustes a partir del monitoreo

## Factores que han facilitado y/o dificultado el desarrollo de mi plan de trabajo

* Planificación de Sprints: La división del proyecto en sprints permitió que cada tarea estuviera claramente definida y alineada con un plazo específico. Al dividir el trabajo en ciclos más pequeños y manejables, se facilitó un enfoque claro sobre qué debía lograrse en cada fase del proyecto, contribuyendo a una sensación constante adaptabilidad al proyecto, priorizando cada vez más las tareas e historias de usuario más críticas para el proyecto..
* Uso de Herramientas de Gestión: La adopción de herramientas como Jira ayudó a planificar, priorizar y monitorear las tareas de manera eficiente. Jira nos proporcionó un marco visual que permitió a todos los integrantes del equipo mantenerse alineados sobre el estado de las tareas, ajustarse a los plazos y reorganizarse rápidamente si surgían cambios.
* Pair Programming: Implementar pair programming resultó ser un factor clave para abordar problemas técnicos complicados. Trabajar de a dos permitió no solo la resolución de problemas de una manera más rápida, sino que también sirvió para transmitir conocimientos dentro del equipo, elevando la capacidad técnica de cada miembro.
* Revisiones de Código: La práctica de revisar el código entre compañeros ayudó a identificar y resolver problemas antes de que se convirtieran en obstáculos más significativos. Estas revisiones mejoraron la calidad del código y aseguraron que las funcionalidades estuvieran alineadas con los objetivos del proyecto, además de contribuir a la creación de un producto más estable.

Factores que han dificultado el desarrollo del plan de trabajo:

* Integración del Backend con MongoDB: La integración entre NestJS y MongoDB presentó desafíos técnicos, principalmente problemas de conexión y optimización de consultas. La sincronización de los datos y la estructura de los documentos requirió ajustes repetitivos que ralentizaron el desarrollo en comparación con los plazos previstos.
* Problemas con la Autenticación JWT: Durante la implementación del sistema de autenticación con JWT, hubo dificultades en la configuración adecuada de la expiración de los tokens y la validación de roles de usuario. Estos problemas afectaron la gestión segura del acceso y la comunicación dentro de la aplicación, lo cual exigió un trabajo adicional para configurar y probar el proceso de autenticación.
* Cobertura de Pruebas Unitarias: Las pruebas unitarias se convirtieron en un reto, ya que la cobertura no era suficiente para algunos módulos críticos del sistema. Esto dificultó la validación de los componentes individuales y afectó el avance hacia las siguientes fases del desarrollo. La cobertura inadecuada generó preocupaciones respecto a la calidad y la estabilidad del software, lo cual requirió asignar más tiempo para mejorar y extender dichas pruebas.
* Retrasos en la Implementación de Módulos: Debido a estos desafíos técnicos, algunos módulos no pudieron completarse a tiempo o enfrentaron retrasos. Esto se reflejó principalmente en el desarrollo de las funcionalidades de gestión de órdenes de transporte y el módulo de entregas, que debieron ser reprogramados para futuros sprints.

## Actividades ajustadas o eliminadas

Desarrollo de módulo para gestión de órdenes de transporte.(ajustada)

Desarrollo de módulo para gestión de entregas (ajustada)

## Actividades que no has iniciado o están retrasadas:

Pruebas de estrés y carga