

|  |
| --- |
| **1. Informe final Proyecto APT** |
| El objetivo de este informe es que describas los aspectos más relevantes de tu Proyecto APT. Es importante que fundamentes las decisiones que tuviste que tomar a lo largo del proceso.  A continuación, encontrarás distintos campos que deberás completar con la información solicitada, los que dan cuenta del resumen de tu proyecto APT y sus principales resultados. |

|  |  |
| --- | --- |
| Nombre del proyecto | *Backstore* |
| Área (s) de desempeño(s) | *Desarrollo de software, Diseño y gestión de requisitos. Gestión de bases de datos, Arquitectura.* |
| Competencias | *Ingeniería de software.*  *Arquitectura de software. Bases de datos. Desarrollo Móvil.*  *Desarrollo Web.* |
| Estudiantes | *Juan Pablo Fuentes Vivanco Alonso González Gómez Paulo Sandoval Retamal* |

Contenidos del informe final

[*1 Relevancia del proyecto APT 3*](#_heading=h.gjdgxs)

[*2 Objetivos 4*](#_heading=h.30j0zll)

[2.1 Objetivo general 4](#_heading=h.1fob9te)

[2.2 Objetivos específicos. 4](#_heading=h.3znysh7)

[*3 Metodología 6*](#_heading=h.2et92p0)

[*4 Desarrollo 8*](#_heading=h.tyjcwt)

[*5 Evidencias 11*](#_heading=h.3dy6vkm)

[5.1 Épicas e Historias de Usuario 11](#_heading=h.1t3h5sf)

[5.2 Matriz módulo o funciones del producto versus Historias de usuario 11](#_heading=h.4d34og8)

[5.3 Product Backlog 11](#_heading=h.2s8eyo1)

[5.4 Sprint Backlog 11](#_heading=h.17dp8vu)

[5.5 Burndown Chart 11](#_heading=h.3rdcrjn)

[5.6 Evidencias del desarrollo 12](#_heading=h.26in1rg)

[5.7 Implementación de ambientes 12](#_heading=h.lnxbz9)

[5.8 Comparación con la industria 12](#_heading=h.35nkun2)

[5.9 Diagramas y evidencias del control del proyecto 14](#_heading=h.1ksv4uv)

[5.10 Evidencia de funcionamiento (Manual) 14](#_heading=h.44sinio)

[5.11 Pruebas – QA del proyecto 14](#_heading=h.2jxsxqh)

[*6 Intereses y proyecciones profesionales 16*](#_heading=h.z337ya)

[*7 Proyecciones laborales a partir de Proyecto APT. 17*](#_heading=h.3j2qqm3)

[*8 Bibliografía 18*](#_heading=h.1y810tw)

[*9 Anexos 19*](#_heading=h.4i7ojhp)

# **Relevancia del proyecto APT**

El proyecto "Backstore" tiene como objetivo abordar la falta de eficiencia en la gestión de pedidos Ship From Store (SFS) y Click & Collect (C&C) para la empresa Corona. Esta problemática es particularmente relevante en el contexto actual, ya que los competidores del sector retail han adoptado metodologías avanzadas para optimizar sus procesos logísticos y mejorar la eficiencia operativa. La optimización de procesos mediante la tecnología es esencial para mantenerse competitivo, especialmente en el ámbito de la logística y el retail, áreas que están directamente relacionadas con la ingeniería en informática aplicada al comercio.

La situación se desarrolla en Chile, en el contexto de la empresa Corona, una entidad clave del sector retail. La falta de eficiencia afecta tanto a los empleados de las tiendas, quienes enfrentan dificultades para gestionar los pedidos de manera ágil, como a los clientes, que esperan tiempos de respuesta rápidos y eficientes. Esto genera un impacto negativo en la productividad interna y en la satisfacción del cliente.

El aporte de valor del proyecto radica en mejorar la logística interna y la eficiencia en los procesos de recolección y entrega de pedidos, contribuyendo directamente a la satisfacción del cliente y a la operatividad de la empresa. A través de la implementación de tecnologías modernas y la optimización de los flujos de trabajo, se busca aumentar la competitividad de Corona en el mercado, mejorando tanto la experiencia del cliente como la eficiencia de los procesos internos de negocio.

# **Objetivos**

## Objetivo general

El objetivo general del proyecto Backstore es implementar desde cero un sistema de administración y logística para gestionar los pedidos bajo las modalidades Ship From Store (SFS) y Click & Collect (C&C) en la empresa Corona en Chile. Esta solución tecnológica tiene como propósito establecer un proceso eficiente de recolección, preparación y entrega de pedidos, con el fin de mejorar la productividad de los empleados y ofrecer a los clientes una experiencia de compra moderna y conveniente.

El proyecto se enfoca en crear la infraestructura tecnológica necesaria para permitir a la empresa incorporar estas modalidades de despacho, logrando así adaptarse a las tendencias actuales del retail y satisfacer la demanda de un servicio más flexible y rápido.

## Objetivos específicos.

**Diseñar la arquitectura tecnológica del sistema:**

* Definir y documentar la infraestructura requerida, incluyendo componentes de backend, frontend, y bases de datos, para soportar la implementación de SFS y C&C.

**Desarrollar el módulo de administración de pedidos:**

* Crear un sistema que permita gestionar los pedidos de manera centralizada, integrando funcionalidades para recibir, procesar y asignar pedidos a las distintas tiendas.

**Implementar una solución de integración móvil:**

* Desarrollar una aplicación móvil que permita a los empleados de tienda gestionar los pedidos asignados, desde la recolección hasta la preparación para el despacho.

**Crear una interfaz amigable para la gestión de órdenes por parte de los encargados de tienda:**

* Desarrollar un portal web que facilite a los encargados de tienda la asignación y seguimiento de órdenes de SFS y C&C, con un enfoque en la usabilidad y eficiencia.

**Integrar el sistema con las plataformas existentes en Corona:**

* Asegurar que el sistema desarrollado pueda comunicarse eficientemente con las plataformas actuales de la empresa, tales como inventarios y sistemas de facturación, para evitar inconsistencias y mejorar la eficiencia de los procesos.

**Desarrollar una solución para la sincronización en tiempo real:**

* Implementar mecanismos que permitan la sincronización en tiempo real entre la aplicación móvil, el portal web y la base de datos, para mantener siempre la información actualizada y evitar errores en la preparación y entrega de pedidos.

**Probar e iterar la solución para asegurar un despliegue exitoso:**

* Realizar pruebas piloto en una o varias sucursales para verificar el correcto funcionamiento de los módulos implementados, recibir retroalimentación de los usuarios, y ajustar las funcionalidades según las necesidades detectadas.

**Capacitar al personal de la empresa en el uso de las nuevas herramientas:**

* Diseñar y ejecutar un plan de capacitación para que los empleados de Corona puedan usar de manera efectiva el sistema y las herramientas de SFS y C&C.

# **Metodología**

Para el desarrollo del proyecto Backstore utilizamos la metodología Scrum, que se caracteriza por un enfoque iterativo e incremental, ideal para proyectos de alta complejidad y con requisitos que pueden evolucionar durante el proceso. Scrum nos permitió dividir el proyecto en sprints de trabajo manejables, cada uno con objetivos claros y medibles, manteniendo un enfoque constante en la mejora continua y la entrega de valor incremental.

El proyecto se estructuró en siete sprints más dos sprint de entregas finales. por lo que en el momento en que se realiza entrega de estos documentos, el proyecto se encuentra en el sprint 8. Estos sprints fueron distribuidos a lo largo de varias fases. Cada sprint tuvo una duración de dos semanas y se centró en objetivos específicos que abordaban aspectos técnicos y funcionales de la implementación del sistema SFS y C&C para la empresa Corona. A continuación, se describen las principales fases y procedimientos desarrollados durante los sprints:

**Sprint 1 - Preparación del entorno y diseño inicial:**

Se revisaron y ajustaron los prototipos, y se preparó el entorno de desarrollo. Esto incluyó la configuración de Git y Azure DevOps, y la creación de los diagramas de casos de uso, clases y despliegue. Aunque se avanzó significativamente en estos aspectos, hubo tareas retrasadas debido a la complejidad del diseño lógico.

**Sprint 2 - Diseño e implementación del backend:**

Se diseñaron los endpoints y esquemas de datos, y se implementó la versión inicial del backend en Node.js. Además, se comenzó el desarrollo del frontend de picking, lo cual permitió establecer las bases para la sincronización con la parte móvil.

**Sprint 3 - Desarrollo del frontend de picking y gestión de usuarios:**

Este sprint se centró en la implementación del frontend para el módulo de picking, así como en la lógica de gestión de usuarios. A pesar de un periodo corto debido a un fin de semana largo, todos los objetivos fueron completados a tiempo.

**Sprint 4 - Gestión de usuarios y preparación para pruebas de integración:**

En este sprint se desarrolló la funcionalidad de asignación de roles y permisos. Debido a una subestimación del esfuerzo requerido y cambios en los requisitos, se presentaron retrasos que llevaron a una re estimación y redistribución de tareas para los sprints siguientes.

**Sprint 5 - Módulo de picking móvil y MVP:**

Se completaron las tareas críticas pendientes del sprint anterior y se avanzó en el desarrollo del módulo de picking móvil. Al finalizar este sprint, logramos un Producto Mínimo Viable (MVP) que nos permitió tener una visión clara de la funcionalidad general del sistema.

**Sprint 6 - Despliegue en entorno cloud:**

Realizamos el despliegue en el entorno de preproducción de Azure, asegurándonos de que el sistema estuviera listo para pruebas funcionales y de rendimiento. Además, se diseñó la API de entregas y se crearon los esquemas correspondientes en MongoDB.

**Sprint 7 - Pruebas y desarrollo del módulo de transporte:**

El desarrollo del módulo de transporte y la finalización de la API de entregas fueron los objetivos principales. También se realizaron pruebas de rendimiento adicionales en la aplicación móvil y se coordinó la integración para garantizar un despliegue estable.

**Pertinencia de la Metodología Scrum**

La metodología Scrum fue especialmente pertinente para el desarrollo del proyecto, ya que nos permitió:

**Adaptabilidad:** Los requisitos del proyecto fueron evolucionando conforme fuimos trabajando en la implementación, y Scrum nos permitió adaptarnos a esos cambios sin perder la visión general. Durante el sprint 4, por ejemplo, se realizaron ajustes en la lógica de roles y permisos que, de haberse realizado sin un enfoque ágil, hubieran generado retrasos significativos.

**Trabajo Iterativo e Incremental:** Cada sprint nos permitió construir sobre lo realizado anteriormente, entregando pequeñas iteraciones del producto que fueron creciendo en complejidad y funcionalidades. Esto nos permitió, en el sprint 5, alcanzar un Producto Mínimo Viable que nos sirvió como base para continuar avanzando hacia el producto completo.

**Enfoque en la Entrega de Valor:** Cada sprint tenía objetivos claros y medibles. Esto permitió al equipo enfocarse en tareas específicas que aportaban valor directo al sistema, tales como el despliegue en Azure durante el sprint 6, que nos permitió comenzar a probar las funcionalidades completas en un entorno realista.

**Colaboración Continua:** Gracias a las ceremonias de Scrum, como las reuniones diarias (dailies) y las retrospectivas, se mantuvo una comunicación constante que fue fundamental para resolver problemas de forma efectiva. La colaboración resultó clave durante el sprint 7, donde todos los miembros trabajaron en conjunto para estabilizar el sistema antes de su lanzamiento.

Scrum no solo facilitó una gestión más organizada del trabajo, sino que también nos permitió reducir los riesgos asociados a la implementación de nuevas tecnologías como Azure y Flutter. Este enfoque ágil nos brindó la capacidad de asegurar una entrega de alta calidad, siempre alineada con los objetivos y expectativas de la empresa Corona.

# **Desarrollo**

El proyecto Backstore se desarrolló a lo largo de varias etapas clave, distribuidas en nueve sprints que cubrían la implementación de las modalidades Ship From Store (SFS) y Click & Collect (C&C) para la empresa Corona. Las principales etapas del proyecto fueron:

**Etapa de Diseño y Preparación del Entorno (Sprint 1 - Sprint 2):**

Revisión de Prototipos y Ajustes de Wireframes: En esta etapa se definieron los prototipos de la aplicación, los flujos de trabajo y los requerimientos funcionales. También se preparó el entorno de desarrollo, incluyendo la integración con Git y Azure DevOps.

Diseño de la API de Picking e Implementación Inicial del Backend: Se diseñaron los endpoints de la API y se comenzó a desarrollar el backend en Node.js para la lógica de picking.

**Etapa de Desarrollo de Funcionalidades (Sprint 3 - Sprint 5):**

Desarrollo del Frontend de Picking y Gestión de Usuarios: Se implementó el frontend para el módulo de picking y la lógica de gestión de usuarios, así como las funcionalidades de marcado de productos como recogidos y la asignación de roles.

Módulo de Picking Móvil y Producto Mínimo Viable (MVP): Se desarrolló el módulo de picking móvil en Flutter, y al finalizar el sprint 5, se logró un Producto Mínimo Viable que incluía las funcionalidades básicas del sistema para empezar las pruebas iniciales.

**Etapa de Pruebas e Integración (Sprint 6 - Sprint 7):**

Despliegue en Entorno de Preproducción en Azure: Se desplegó el sistema en Azure para realizar pruebas funcionales y de rendimiento, garantizando que todas las funcionalidades estuvieran integradas y funcionando de manera adecuada.

Pruebas del Módulo de Transporte y Finalización de la API de Entregas: Se desarrolló el frontend para la gestión de órdenes de transporte y se completaron las pruebas de integración para la API de entregas.

**Etapa de Despliegue y Finalización (Sprint 8 - Sprint 9):**

Despliegue en Ambiente de Pruebas: Se realizaron pruebas en un entorno controlado para validar la estabilidad y la robustez del sistema antes de su lanzamiento en producción.

Revisión Final y Entrega del Producto: Se realizaron los ajustes finales y se completó la entrega del sistema, listo para el despliegue en producción.

**Historias de Usuario del Proyecto**

*\* Este apartado tiene documentación asociada en el anexo de este entregable “****Evidencias/Scrum/Backstore Backlog.xlsx****” \*.*

**Facilitadores:**

Uso de Azure y Git para la Integración Continua: La integración con Azure DevOps y Git nos permitió mantener un control adecuado de versiones y llevar a cabo el despliegue continuo, lo que facilitó enormemente la colaboración y la gestión del código.

Metodología Scrum: Nos permitió adaptarnos a cambios en los requisitos y trabajar de manera incremental, asegurando una entrega continua de valor y la capacidad de revisar y ajustar las tareas durante cada sprint.

Colaboración y Comunicación Constante: Las reuniones diarias (dailies) y las retrospectivas después de cada sprint facilitaron la comunicación y la identificación temprana de problemas.

**Dificultades:**

Complejidad en la Lógica de Roles y Permisos: Durante el sprint 4, la implementación de la lógica de roles y permisos resultó ser más compleja de lo anticipado. Esto retrasó el desarrollo y requirió una reestimación de los esfuerzos para los sprints siguientes.

**Problemas de Sincronización de la Aplicación Móvil:** La implementación de la funcionalidad de sincronización en Flutter enfrentó desafíos técnicos relacionados con el rendimiento y la sincronización en tiempo real.

**Cambio en Requisitos:** Hubo modificaciones en los requisitos durante el desarrollo, lo cual afectó el avance de algunas tareas, especialmente durante los sprints 4 y 5, requiriendo una replanificación.

**Ajustes Realizados para Abordar las Dificultades**

**Reestimación de Esfuerzos y Redistribución de Recursos:**

Durante el sprint 4, reestimamos las tareas pendientes y redistribuimos los puntos planeados para el sprint 5, asegurando que las tareas críticas se completaran.

**Extensión del Sprint y División de Tareas Complejas:**

En el sprint 5, aumentamos la cantidad de tareas asociadas por persona y dividimos las tareas más grandes en subtareas manejables, lo cual nos permitió cumplir con las funcionalidades necesarias para lograr el Producto Mínimo Viable.

**Pruebas y Despliegue Controlado en Azure:**

En el sprint 6, realizamos un despliegue en un entorno de preproducción en Azure para llevar a cabo pruebas funcionales y de rendimiento antes del despliegue final, lo cual ayudó a identificar y corregir errores críticos a tiempo.

**Asignación de Recursos Adicionales:**

Durante los sprints donde se identificaron retrasos importantes, asignamos más recursos a las tareas críticas para asegurar su finalización y minimizar el impacto en los plazos de entrega.

**Monitoreo Constante y Enfoque en Prioridades:**

Para evitar más retrasos, durante el sprint 5 implementamos un monitoreo constante con reuniones diarias más largas y un enfoque en priorizar tareas clave que eran esenciales para cumplir los objetivos establecidos.

Este enfoque estructurado y adaptativo permitió al equipo lidiar con las dificultades y garantizar que el proyecto cumpliera con los objetivos planteados. Al implementar de forma iterativa los módulos y al ajustar las prioridades según las necesidades, logramos no solo un sistema funcional, sino también una solución robusta que satisface las expectativas del negocio.

# **Evidencias**

Adjunta evidencias que permitan dar cuenta del desarrollo del Proyecto APT y sus resultados finales.

## Épicas e Historias de Usuario

*\* Este apartado tiene documentación asociada en el anexo de este entregable “****Evidencias/Scrum/Backstore Backlog.xlsx****” \*.*

## Matriz módulo o funciones del producto versus Historias de usuario

*\* Este apartado tiene documentación asociada en el anexo de este entregable “****Evidencias/Scrum/Backstore Backlog.xlsx****” \*.*

## Product Backlog

*\* Este apartado tiene documentación asociada en el anexo de este entregable “****Evidencias/Scrum/Backstore Backlog.xlsx****” \*.*

## Sprint Backlog

*\* Este apartado tiene documentación asociada en el anexo de este entregable “****Evidencias/Scrum/Backstore Backlog.xlsx****” \*.*

## Burndown Chart

*\* Este apartado tiene documentación asociada en el anexo de este entregable “****Evidencias/Scrum/Backstore Backlog.xlsx****” \*.*

## Evidencias del desarrollo

1. Código fuente **web** versión final (link github):   
     
   <https://github.com/alonsillo88/coronaduoc> (Rama **backstore-frontend-react**)
2. Código fuente **aplicación móvil** versión final (link github)

<https://github.com/alonsillo88/coronaduoc> (Rama **AppMovil**)

1. Script base de datos (link github).

Se adjuntan colecciones de mongoDB para la creación de la base de datos en los anexos del documento.

1. Código fuente **backend** versión final (link github)

<https://github.com/alonsillo88/coronaduoc> (Rama **backstore-backend-api**)

## Implementación de ambientes

Se utilizó Microsoft Azure como la nube para el proyecto, ya que es la plataforma utilizada por Corona, y ofrece seguridad, escalabilidad y disponibilidad de los servicios. Además, se utilizó MongoDB Atlas, la nube oficial de MongoDB, para gestionar la base de datos de manera eficiente, asegurando alta disponibilidad y escalabilidad para los datos críticos del sistema.

*\* Este apartado tiene documentación asociada en el anexo de este entregable, en el directorio “Anexos/****Evidencias/Ambiente Cloud/****” \*, donde se encontrarás las imágenes asociadas a lo realizado para los componentes FrontEnd, Backend y la base de datos en MongoDB Atlas.*

## Comparación con la industria

El proyecto Backstore fue desarrollado tomando en cuenta prácticas y herramientas alineadas con los estándares de la industria, asegurando tanto la calidad del software como la escalabilidad y eficiencia de la solución. A continuación, se presentan las principales comparaciones con la industria:

Uso de **Microsoft Azure** como Plataforma de Nube:

Microsoft Azure es uno de los proveedores de servicios de nube más conocidos y utilizados en la industria del retail, lo que garantiza el cumplimiento de estándares de seguridad, disponibilidad y escalabilidad. La elección de Azure no solo responde a la infraestructura ya utilizada por la empresa Corona, sino que también nos permitió aprovechar sus características avanzadas como Azure DevOps, servicios de CI/CD y Azure App Services.

La utilización de Azure asegura la confiabilidad y escalabilidad del sistema, lo que es clave para una empresa del tamaño de Corona. En la industria, garantizar que una infraestructura pueda crecer conforme aumentan las demandas es fundamental para evitar problemas de sobrecarga o fallos en los servicios.

**Integración Continua y Entrega Continua (CI/CD):**

Implementamos un pipeline de CI/CD con Azure DevOps que está alineado con las mejores prácticas de la industria del software. Esto nos permitió automatizar la integración de cambios y el despliegue de nuevas versiones del sistema de manera rápida y segura, lo cual es un estándar de calidad esencial en los proyectos modernos para minimizar el riesgo de errores durante la integración.

La automatización del ciclo de desarrollo e implementación garantiza una rápida adaptación a los cambios, una menor probabilidad de errores humanos y la posibilidad de iterar sobre la base de retroalimentación continua.

**Arquitectura Escalable y Modular:**

La arquitectura de Backstore fue diseñada siguiendo principios de modularidad, lo cual facilita la escalabilidad y el mantenimiento a largo plazo, características que están alineadas con los estándares de la industria del retail. El diseño de módulos independientes (como Picking, C&C, y Gestión de Usuarios) permite que futuras actualizaciones se realicen de manera localizada sin afectar otras partes del sistema, cumpliendo con la buena práctica de separación de responsabilidades.

La separación de los componentes permite también una mejor integración de nuevas funcionalidades, lo cual es clave en un entorno donde los requerimientos de negocio suelen cambiar con rapidez.

**Seguridad en el Desarrollo y Despliegue:**

Azure nos proporciona servicios de seguridad integrados como el Active Directory para la autenticación y el control de acceso, y los mecanismos de firewall para proteger los servicios desplegados. Cumplir con los estándares de seguridad es fundamental en un contexto como el retail, donde la protección de los datos de los clientes y la integridad del sistema son críticas.

Además, utilizamos prácticas de seguridad durante el desarrollo como la validación de inputs, la gestión de permisos a través de roles, y la protección de endpoints sensibles, todo en concordancia con los estándares de OWASP que dictan las mejores prácticas para el desarrollo seguro de software.

## Diagramas y evidencias del control del proyecto

*\* Este apartado tiene documentación asociada en el anexo de este entregable, en el directorio “****Evidencias/Diagramas/****” \*.*

## Evidencia de funcionamiento (Manual)

*\* Este apartado tiene documentación asociada en el anexo de este entregable, en el directorio “****Evidencias/Videos/****” \*.*

## Pruebas – QA del proyecto

1. **Pruebas Unitarias**

*\* Este apartado tiene documentación asociada en el anexo de este entregable, en el directorio “****Evidencias/Pruebas/****” \*.*

1. **Pruebas de Integración**

*\* Este apartado tiene documentación asociada en el anexo de este entregable, en el directorio “****Evidencias/Pruebas/****” \*.*

1. **Pruebas Funcionales**

*\* Este apartado tiene documentación asociada en el anexo de este entregable, en el directorio “****Evidencias/Pruebas/****” \*.*

1. **Pruebas de Regresión**

*\* Este apartado tiene documentación asociada en el anexo de este entregable, en el directorio “****Evidencias/Pruebas/****” \*.*

1. **Pruebas de Usuario (si aplica)**

*\* Este apartado tiene documentación asociada en el anexo de este entregable, en el directorio “****Evidencias/Pruebas/****” \*.*

1. **Pruebas de Rendimiento**

*\* Este apartado tiene documentación asociada en el anexo de este entregable, en el directorio “****Evidencias/Pruebas/****” \*.*

# **Intereses y proyecciones profesionales**

**Alonso González:**

Durante el proyecto APT Backstore, Alonso se centró en el desarrollo del backend, utilizando Node.js con NestJS e integrando la base de datos con MongoDB Atlas. Este trabajo le permitió profundizar sus conocimientos en infraestructura de la nube, gestionando el despliegue en Microsoft Azure y comprendiendo mejor los procesos de seguridad y escalabilidad. Gracias a esta experiencia, Alonso ha reafirmado su interés en el desarrollo de software backend y en la gestión de infraestructuras escalables. Su interés por las tecnologías en la nube y la optimización de procesos logísticos se ha fortalecido, y se siente motivado a seguir explorando el área de DevOps y la arquitectura de sistemas distribuidos.

**Juan Pablo Fuentes:**   
Juan Pablo fue responsable del desarrollo de la aplicación móvil utilizando Flutter y también trabajó en la integración con el backend mediante GraphQL. Esta experiencia le permitió comprender la importancia de la sincronización en tiempo real y la escalabilidad de las aplicaciones móviles, especialmente al utilizar MongoDB Atlas para la gestión eficiente de datos. Además, participó en el despliegue de los servicios en Azure, lo cual expandió sus habilidades en la gestión de entornos cloud. Luego de finalizar el proyecto, Juan Pablo mantiene su interés en el desarrollo móvil, y desea profundizar sus conocimientos en experiencia de usuario y optimización de aplicaciones para diferentes plataformas.

**Paulo Sandoval:**   
Paulo trabajó principalmente en el frontend utilizando React, y también ejerció el rol de Scrum Master, coordinando las actividades del equipo. El uso de Azure para el despliegue le permitió entender mejor cómo interactúan los diferentes servicios en un entorno cloud. Además, la integración con MongoDB Atlas le ayudó a conocer más sobre la gestión eficiente de datos desde la perspectiva del frontend. Su participación en la coordinación de equipos con Scrum y en la gestión de infraestructura ha reafirmado su interés tanto en el desarrollo de interfaces dinámicas como en el rol de facilitador de equipos. Tras concluir el proyecto, Paulo se siente motivado a continuar con una trayectoria híbrida, combinando el desarrollo frontend con la gestión ágil de proyectos.

# **Proyecciones laborales a partir de Proyecto APT.**

**Alonso González**:   
Luego de completar el proyecto APT Backstore, Alonso se ha dado cuenta de lo importante que es la infraestructura en la nube para mantener un sistema confiable y escalable. Le interesa explorar más a fondo el área de DevOps, particularmente trabajando con Microsoft Azure y otras plataformas de nube, para continuar especializándose en la automatización de despliegues y la gestión de infraestructura. A futuro, se proyecta en una posición donde pueda liderar equipos técnicos y enfocar sus esfuerzos en la arquitectura de sistemas distribuidos, ayudando a empresas a resolver problemas logísticos con soluciones eficientes y escalables.

**Juan Pablo Fuentes**:   
Durante el proyecto, Juan Pablo fortaleció su interés por el desarrollo móvil. Después de trabajar en la integración con Flutter y experimentar con la sincronización en tiempo real, ahora quiere seguir profundizando en el desarrollo de aplicaciones móviles híbridas y la experiencia del usuario (UX). A corto plazo, se ve participando en proyectos donde pueda seguir creando aplicaciones atractivas y funcionales, aplicando sus conocimientos en MongoDB Atlas para garantizar un rendimiento eficiente. Le gustaría trabajar en un ambiente donde pueda aprender más sobre cómo optimizar la experiencia de usuario y hacer que las aplicaciones móviles sean más intuitivas y fluidas.

**Paulo Sandoval**:   
Paulo disfrutó especialmente de su rol como Scrum Master en el proyecto, lo cual le permitió ayudar al equipo a coordinarse y avanzar de manera efectiva. Al mismo tiempo, el desarrollo del frontend con React despertó su interés por mejorar en el diseño y la optimización de interfaces. Le gustaría seguir desempeñando una función dual, donde pueda contribuir tanto en la gestión de proyectos como en el desarrollo técnico. Paulo se proyecta trabajando en empresas donde la gestión ágil sea fundamental para el éxito del proyecto, permitiéndole aplicar sus habilidades tanto en el desarrollo de software como en la facilitación de equipos hacia sus objetivos.

# **Bibliografía**

Microsoft Azure Documentation. Microsoft Azure. Consultado para implementar infraestructura en la nube, administración de recursos y despliegue de servicios utilizando Azure DevOps y Azure App Services. https://azure.microsoft.com.

MongoDB Atlas Documentation. MongoDB. Utilizado para la gestión de base de datos, asegurando escalabilidad y alta disponibilidad para los datos del sistema Backstore. [https://www.mongodb.com/atlas.](https://www.mongodb.com/atlas)

NestJS Documentation. NestJS Framework. Referencia utilizada para la implementación del backend en el proyecto APT, optimizando la creación de APIs con una arquitectura modular y escalable. [https://nestjs.com.](https://nestjs.com)

React Documentation. Meta (Facebook). Consultado para el desarrollo del frontend web del proyecto, enfocado en interfaces dinámicas y responsivas. [https://reactjs.org.](https://reactjs.org)

Flutter Documentation. Google. Utilizado en la construcción de la aplicación móvil, permitiendo el desarrollo multiplataforma con una sola base de código. [https://flutter.dev.](https://flutter.dev)

Scrum Guide. Schwaber, K. & Sutherland, J. (2020). Scrum: The Art of Doing Twice the Work in Half the Time. Scrum.org. Referencia utilizada para la aplicación de la metodología ágil en la planificación, ejecución y gestión del proyecto. [https://scrumguides.org.](https://scrumguides.org)

OWASP Top 10 Security Practices. OWASP Foundation. Consultado para garantizar la seguridad del software y la protección de datos sensibles durante el desarrollo del proyecto Backstore. [https://owasp.org/www-project-top-ten/.](https://owasp.org/www-project-top-ten/)

Azure DevOps Documentation. Microsoft Azure. Utilizado para la planificación y gestión de tareas, así como para la integración y entrega continua (CI/CD). [https://azure.microsoft.com/en-us/services/devops/.](https://azure.microsoft.com/en-us/services/devops/)

GraphQL Documentation. GraphQL Foundation. Consultado para la integración entre el backend y las aplicaciones móviles y web, permitiendo una mejor gestión y consulta de datos. [https://graphql.org.](https://graphql.org)

JIRA Software Documentation. Atlassian. Utilizado para el seguimiento de tareas y la gestión del trabajo en equipo, alineado con la metodología Scrum. [https://www.atlassian.com/software/jira.](https://www.atlassian.com/software/jira)

Azure App Service Documentation. Microsoft. Consultado para el despliegue de la aplicación web, utilizando servicios cloud para garantizar la disponibilidad y escalabilidad. [https://docs.microsoft.com/en-us/azure/app-service/.](https://docs.microsoft.com/en-us/azure/app-service/)

# **Anexos**

Los archivos de este informe se pueden encontrar en la carpeta Anexo/Evidencias:

9.1 -Anexo A: Ambiente Cloud

9.2 -Anexo B: Diagramas  
9.3 -Anexo C: Pruebas  
9.4 -Anexo C: Scrum  
9.5 -Anexo D: Videos