**FASE 2**

**DISEÑO E IMPLEMENTACIÓN DEL PROYECTO**

**CleanFlow: PWA de gestión de productos de aseo.**

Escuela de Informática y Telecomunicaciones

Mes 2025

1. Identificación del Proyecto

|  |
| --- |
| **Nombre de Proyecto** |
| CleanFlow: PWA de gestión de productos de aseo. |

1. Integrantes del Equipo de Trabajo

|  |  |  |  |
| --- | --- | --- | --- |
| **N°** | **Rut** | **Apellidos** | **Nombres** |
| 1 | 21.145.851-8 | Aguilera Pinto | Jorge Maximiliano |
| 2 | 20.541.106-2 | Fernández Muñoz | Lucas Ignacio |
| 3 | 21.535.925-5 | Garrido Sánchez | Jeremy Alejandro |

1. Registro de Control de Cambios

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Revisión** | **Fecha** | **Páginas** | **Descripción del Cambio** | **Autor** |
| 1 | 14-09-25 | Todas | Cambios según primera revisión docente (alcances y objetivos). | DDI |
| 2 | 21-09-25 | n° 17 | Modificación de roles del equipo | DDI |
| 3 | 01-10-25 | n° 17 | Definición de riesgos del proyecto | DDI |

**ÍNDICE DE CONTENIDO**

[LISTA DE TABLAS 4](#_heading=h.yfofi16cgzaz)

[LISTA DE FIGURAS 5](#_heading=h.35l1se2mswgc)

[LISTA DE DIAGRAMAS 7](#_heading=h.pkl45iifq2ts)

[GLOSARIO 8](#_heading=h.x3cwehh8vjtu)

[1 Diseño e Implementación del Proyecto 9](#_heading=h.slk16z75dptp)

[1.1 Resumen 9](#_heading=h.bw0xd24dj5su)

[1.2 Abstract 10](#_heading=h.p42k0q3k22t3)

[1.3 Introducción 11](#_heading=h.ieova6xrckgi)

[1.4 Problemática a solucionar o necesidad a satisfacer 11](#_heading=h.x0j3utueeqpz)

[1.5 Objetivos del Proyecto (general y específicos) 12](#_heading=h.ialjuu2todlp)

[1.6 Competencias del Perfil de Egreso 12](#_heading=h.nalol01dp0l4)

[1.7 Asignación de roles 13](#_heading=h.ty8worqfjmzh)

[1.8 Metodología utilizada en el Proyecto. 14](#_heading=h.6yluqkrpiuyw)

[1.9 Creación de cronograma asociado al Proyecto (Carta Gantt) 16](#_heading=h.od4u9iswl9sp)

[1.10 Riesgos Asociados al Proyecto 17](#_heading=h.fhitv16k0r3k)

[1.11 Implementación del Proyecto 17](#_heading=h.p5lm5w34srb)

[Diseño y Arquitectura de la solución (Caso de uso de Software o plataforma de gestión) 17](#_heading=h.ybf9jn6kiw0y)

[Requerimientos técnicos 17](#_heading=h.vse3ypue1emj)

[Desarrollo de la solución 17](#_heading=h.15f2zdvezndq)

[Resultados de la solución 18](#_heading=h.ky64vvj1hco1)

[1.12 Definición de Recursos y Costos asociados al Proyecto 18](#_heading=h.9zkcztwypx0l)

[CONCLUSIÓN 19](#_heading=h.g2f7wxlpwqhy)

[BIBLIOGRAFÍA 20](#_heading=h.3pybn9qi01u)

[ANEXOS 21](#_heading=h.uipiea1sln7s)

# LISTA DE TABLAS

[Figura 2. Tabla de Riesgos 18](#_Toc210732635)

# LISTA DE FIGURAS

[Figura 1. Carta Gantt 15](#_Toc210732640)

[Figura 4. Prototipo de interfaz 22](file:///C:\Users\jmaxi\OneDrive\Desktop\Capstone\2.2\2.4_GuiaEstudiante_Fase%202_DesarrolloProyecto%20APT.docx#_Toc210732643)

[Figura 5. Prototipo de interfaz 22](file:///C:\Users\jmaxi\OneDrive\Desktop\Capstone\2.2\2.4_GuiaEstudiante_Fase%202_DesarrolloProyecto%20APT.docx#_Toc210732644)

# LISTA DE DIAGRAMAS

[[Figura 3. Diagrama de casos de uso 22](#_Toc210748728)](file:///C:\Users\jmaxi\OneDrive\Desktop\Capstone\2.2\2.4_GuiaEstudiante_Fase%202_DesarrolloProyecto%20APT.docx#_Toc210752278)

[[Figura 6. Diagrama Base de datos 24](#_Toc210748728)](#_Toc210752281)

[[Figura 7. Diagrama Arquitectura de Software 25](#_Toc210748728)](#_Toc210752282)

[Base de datos 24](#_Toc210748728)

# GLOSARIO

**PWA (Aplicación Web Progresiva):** Aplicación web que combina la experiencia de los sitios web y las aplicaciones nativas, usando tecnologías web para ofrecer acceso sin conexión, notificaciones push e instalación en la pantalla de inicio desde un navegador.

**QA:** Equipo o profesional encargado de asegurar que la calidad del producto o proyecto cumpla con los estándares de calidad previamente establecidos mediante pruebas exhaustivas.

**Backend:** Parte de una aplicación web que no es visible para el usuario, la cual se encarga de la lógica, el procesamiento de datos, la conexión a bases de datos y la seguridad del sistema.

**Frontend:** Se encarga de la presentación del contenido, el diseño, la navegación y todos los elementos interactivos como botones y formularios de un sistema, es decir, la interfaz visual.

**Carta Gantt:** Planificación del desarrollo del proyecto a lo largo de un tiempo definido.

**Scrum:** Marco de desarrollo ágil que tiene consigo un conjunto de buenas prácticas para trabajar de manera colaborativa.

**Scrum Master:** Profesional que se encargará de dirigir al equipo, utilizando técnicas de gestión de proyectos. Además será encargado de facilitar la comunicación y colaboración en el equipo de trabajo.

**Product Owner:** Es el encargado de que el producto final tenga un alto valor para el cliente y el mercado. Además será el que cuente con conocimientos sobre el negocio.

**Developer Team:** Conjunto de equipos el cual será encargado del desarrollo de las funcionalidades del sistema.

**Hardware:** Se refiere a las partes físicas y tangibles de un sistema informático, sus componentes eléctricos, electrónicos y electromecánicos.

**Software:** Conjunto de algoritmos y partes visuales, la cual nos permitirá la interacción con dispositivos electrónicos.

**API:** Conjunto de reglas y protocolos que permite que dos aplicaciones de software se comuniquen entre sí, intercambiando datos y funcionalidades de una manera estandarizada.

**Base de datos:** Colección organizada de datos (números, palabras, imágenes, etc.) almacenados en un entorno local o servidor, que permite su fácil recuperación, gestión y uso mediante un sistema de gestión de bases de datos.

# Diseño e Implementación del Proyecto

## Resumen

Donde Gino es una empresa dedicada a la comercialización de artículos de limpieza para clientes minoristas y mayoristas. Antes del proyecto, operaba con información dispersa en planillas y procesos manuales que dificultaban el control del catálogo, la consistencia de precios y la visibilidad del inventario por bodega. Esta realidad ocasionaba errores frecuentes en el valor de los productos, quiebres no detectados, demoras en la atención y ausencia de indicadores oportunos para decidir las diferentes reposiciones y compras, lo que afectaba tanto la experiencia de los clientes como la eficiencia operativa de la empresa.

A partir de esa problemática se propuso una solución integral que unificó datos y flujos en una plataforma web. La propuesta contempló un catálogo centralizado con categorías y precios sellados al momento de la venta, un flujo completo de emisión de boletas con sus detalles, registro de pagos, control de inventario por bodega con kardex de movimientos y trazabilidad, y un módulo de gestión de usuarios y roles para administrar permisos y responsabilidades. Además, se definieron reportes diarios de ventas, existencias y productos más rotados para entregar visibilidad inmediata a los responsables de operación y compras.

El proyecto se abordó en etapas siguiendo un enfoque ágil. Primero se levantaron y priorizaron requerimientos con los equipos de ventas, bodega y administración, y se registraron reglas clave del negocio como manejo de IVA, estados de documentos y políticas de stock. Luego se diseñó la experiencia de usuario y se modeló la base de datos relacional que sustenta catálogo, ventas, pagos, inventario y seguridad. Posteriormente se desarrolló de forma incremental el backend y el frontend tipo PWA, se integraron validaciones de negocio y el motor de inventario con movimientos de ingreso, salida y ajuste, y se incorporó la autenticación con asignación de roles. Finalmente se realizaron pruebas funcionales y de rendimiento con datos reales, se ejecutó una migración inicial de productos y clientes, se capacitó a los usuarios y se puso en marcha la solución con acompañamiento cercano para la adopción.

Con la implementación, Donde Gino evolucionó para así poder contar con procesos estandarizados y datos consistentes en una única plataforma. La empresa redujo retrabajos y errores en precios e inventario, evitó quiebres gracias a existencias y movimientos actualizados en línea, y aceleró la atención en el punto de venta al simplificar el registro de boletas y pagos. La trazabilidad por producto y por documento permitió auditar operaciones y resolver incidencias con rapidez, mientras que los reportes diarios facilitaron decisiones de reposición y negociación con proveedores. En términos de impacto, la organización mejoró la eficiencia operativa, disminuyó costos administrativos, optimizó la rotación de inventario y ofreció una experiencia de compra más consistente para sus clientes, consolidando una base tecnológica sobre la cual proyectar nuevas capacidades como promociones, listas de precios por segmento y despacho a domicilio.

## Abstract

Donde Gino is a company dedicated to selling cleaning supplies to retail and wholesale customers. Before the project, it operated with scattered information in spreadsheets and manual processes that made it difficult to control the catalog, maintain price consistency, and maintain inventory visibility by warehouse. This reality led to frequent errors in product pricing, undetected stockouts, delays in customer service, and a lack of timely indicators for deciding on different replenishments and purchases, which affected both the customer experience and the company's operational efficiency.

Based on this problem, a comprehensive solution was proposed that unified data and workflows in a web platform. The proposal included a centralized catalog with categories and prices sealed at the time of sale, a complete invoice issuance workflow with details, payment registration, inventory control by warehouse with a transaction log and traceability, and a user and role management module to manage permissions and responsibilities. In addition, daily reports on sales, stock, and most frequently rotated products were defined to provide immediate visibility to operations and purchasing managers.

The project was approached in stages following an agile approach. Requirements were first identified and prioritized with the sales, warehouse, and administration teams, and key business rules such as VAT management, document statuses, and stock policies were recorded. The user experience was then designed and the relational database supporting the catalog, sales, payments, inventory, and security was modeled. The PWA-style backend and frontend were then incrementally developed, business validations and the inventory engine with input, output, and adjustment transactions were integrated, and authentication with role assignment was incorporated. Finally, functional and performance tests were conducted with real data, an initial migration of products and customers was executed, users were trained, and the solution was launched with close support for adoption.

With the implementation, Donde Gino evolved to have standardized processes and consistent data on a single platform. The company reduced rework and errors in pricing and inventory, avoided stock shortages thanks to updated stock and transaction information online, and accelerated service at the point of sale by simplifying the recording of invoices and payments. Traceability by product and document enabled operations to be audited and incidents resolved quickly, while daily reports facilitated replenishment decisions and supplier negotiations. In terms of impact, the organization improved operational efficiency, reduced administrative costs, optimized inventory turnover, and offered a more consistent shopping experience for its customers, consolidating a technological foundation on which to develop new capabilities such as promotions, segment-based price lists, and home delivery.

## Introducción

El proyecto CleanFlow surge para modernizar y digitalizar la gestión de Donde Gino, empresa dedicada a la comercialización de artículos de limpieza para el hogar y el rubro institucional, con ventas a clientes minoristas y mayoristas. El crecimiento dejó en evidencia limitaciones de procesos manuales: inventario en planillas, registro parcial de boletas y pagos, datos de clientes dispersos y escasa trazabilidad de entradas y salidas. Esto se tradujo en inconsistencias de precios, quiebres por falta de visibilidad de stock, demoras en el punto de venta y dificultades para obtener reportes confiables que apoyaran la toma de decisiones.

CleanFlow aborda este contexto mediante una plataforma web tipo PWA (Progressive Web App) que centraliza catálogo y categorías, clientes, inventarios por bodega con kardex de movimientos, emisión de boletas con sus detalles, registro de pagos y administración de usuarios y roles. Al ser PWA, la solución se instaló como app ligera en navegadores de escritorio y móviles, habilitando acceso uniforme sin despliegues complejos, mejorando la experiencia en terreno y permitiendo capacidades como funcionamiento resiliente con conectividad intermitente y notificaciones para eventos clave (por ejemplo, reposición o confirmación de venta).

El proyecto se ejecutó de forma incremental para minimizar interrupciones: primero consolidación del catálogo y modelado relacional, luego flujo de ventas y pagos, y finalmente control de inventario y reportes operacionales. Con esta base tecnológica, Donde Gino estandarizó procesos, mejoró la calidad de datos y sentó condiciones para incorporar de manera ordenada futuras capacidades (promociones, listas de precio por segmento y módulo de despacho), fortaleciendo su eficiencia interna y la experiencia de compra de sus clientes.

## Problemática a solucionar o necesidad a satisfacer

Con la realidad descrita en la introducción, Donde Gino arrastraba una problemática estructural ligada a la falta de gestión integrada de sus operaciones. El control de productos, clientes, boletas y pagos se realizaba en planillas y archivos separados, mantenidos por distintas personas, lo que derivaba en duplicidad de información, inconsistencias de precios y demoras en el registro de ventas. El inventario se actualizaba de forma manual y con rezago; no existía trazabilidad de entradas, salidas ni ajustes por bodega, y la disponibilidad se estimaba “a ojo”, lo que ocasionaba quiebres o sobrestock sin una señal temprana.

A medida que crecieron el catálogo y la base de clientes, estas limitaciones se hicieron más visibles. Faltaban reportes consolidados para identificar productos más vendidos, rotación por categoría, puntos de reorden y márgenes reales considerando descuentos e impuestos. La ausencia de perfiles y permisos diferenciados impedía delegar con seguridad y elevaba el riesgo de ediciones no autorizadas o errores involuntarios; tampoco se contaba con auditoría de acciones para reconstruir el origen de una inconsistencia. En la operación diaria, esto se traducía en tiempos de atención más largos, cuadraturas frecuentes y toma de decisiones basada en datos incompletos.

Ante ese escenario, la necesidad fue disponer de un sistema único que centralizara el catálogo y las reglas de negocio, asegurara la emisión correcta de boletas y el registro de pagos, incorporara un kardex de inventario por bodega con actualizaciones en tiempo real y ofreciera reportes operacionales diarios. Además, se requirió establecer seguridad por roles y trazabilidad de cambios para resguardar la integridad de la información. Esa brecha operativa y de información justificó el desarrollo de CleanFlow como plataforma web PWA orientada a ordenar los datos, reducir errores y habilitar decisiones oportunas sobre información confiable.

## Objetivos del Proyecto (general y específicos)

**Objetivo General**

Desarrollar e implementar un sistema de gestión llamado CleanFlow, que optimice los procesos relacionados con productos, clientes, inventarios y ventas en la empresa Donde Gino, con el fin de mejorar la eficiencia operativa, reducir errores y proporcionar información confiable para la toma de decisiones estratégicas dentro de la empresa.

**Objetivos Específicos**

* Garantizar la integridad y consistencia de la información de productos, categorías, clientes, boletas y pagos, evitando duplicidad de datos y reduciendo errores en los registros.
* Facilitar la administración y trazabilidad de las operaciones de ventas, inventario y gestión de clientes mediante módulos funcionales integrados que optimizaron el flujo de trabajo.
* Aumentar la seguridad y confiabilidad de la información a través de un sistema de control de accesos y roles que permitió diferenciar permisos según las funciones de cada usuario.
* Mejorar la toma de decisiones estratégicas mediante la incorporación de reportes automatizados y dashboards que entregaron información en tiempo real sobre ventas, stock y desempeño de productos.

## Competencias del Perfil de Egreso

A través de este proyecto se permitió potenciar competencias propias del perfil de egreso de la carrera de Ingeniería en Informática, ya que el desarrollo del sistema exigió aplicar conocimientos de modelamiento de datos, programación y construcción de soluciones tecnológicas de acuerdo con los requerimientos de la organización. En este sentido, las competencias fortalecidas fueron:

* **Diseña un modelo relacional normalizado que permita satisfacer los requerimientos de información de la organización en forma escalable**. Esta competencia se reflejó en el diseño de la base de datos que estructuró los distintos módulos del sistema, garantizando la integridad y consistencia de los datos.
* **Implementa el modelo relacional en la base de datos generando las condiciones para el almacenamiento, procesamiento, manipulación y consulta de datos de acuerdo al modelo conceptual planteado**. Su aplicación permitió construir una base sólida que facilitó el registro y la administración de la información en tiempo real.
* **Construir programas y rutinas de variada complejidad para dar solución a requerimientos de la organización, acordes a tecnologías de mercado y utilizando buenas prácticas de codificación.** Esta competencia se evidenció en el desarrollo de los módulos funcionales, asegurando que cada uno respondiera a las reglas de negocio establecidas.
* **Construye aplicaciones web en un lenguaje de programación que soporten los requerimientos de la organización**. Gracias a esta competencia fue posible implementar una interfaz práctica y accesible, que facilitó la gestión de los procesos de la empresa.

## Asignación de roles

En vista de que el tiempo total para trabajar con este proyecto es de aproximadamente 4 meses y será desarrollado por un grupo de 3 integrantes se ha optado por la separación de las funciones que realizará cada uno de los integrantes tomando esto en cuenta, optando por repartir las funciones de desarrollo principalmente en aquellas que abarcan el frontend junto con todos los aspectos de interfaces visuales del sistema y en aquellas que abarcan el backend y toda la lógica por detrás de las funciones del sistema.

Además de esto, y como se abordará en el punto siguiente, la metodología adoptada para el desarrollo de este proyecto será Scrum, por lo cual se ha optado por la designación de las siguientes funciones/roles para cada uno de los integrantes de este equipo:

**Lucas Fernández – Scrum Master / Desarrollador Backend**

* Facilitar reuniones Scrum: Daily Meeting, Sprint Planning, Review y Retrospective.
* Monitorear el progreso del sprint y remover impedimentos.
* Desarrollar APIs y lógica de negocio del backend (productos, inventario, ventas, integración Mercado Pago).
* Documentar endpoints y procesos del backend.
* Implementación de seguridad (roles, encriptación, auditoría).

**Jorge Aguilera – Product Owner / Desarrollador Frontend**

* Mantener y priorizar el Product Backlog.
* Escribir historias de usuario claras y criterios de aceptación.
* Validar funcionalidades con criterios de negocio y aprobar entregables del sprint.
* Desarrollar el frontend web: catálogo, carrito, checkout, características PWA (manifest, service worker).

**Jeremy Garrido – Desarrollador Fullstack / QA**

* Participar activamente en todas las tareas de desarrollo: frontend, backend y base de datos según necesidad.
* Configurar base de datos, consultas y rutinas para reportes e inventario.
* Apoyar en el desarrollo e implementación de APIs.
* Realizar QA y pruebas unitarias/integración, asegurando que el sistema funcione correctamente.
* Apoyar en integración de notificaciones push, dashboards y PWA offline.

## Metodología utilizada en el Proyecto.

Para el desarrollo del proyecto CleanFlow se implementará la metodología ágil **Scrum**, la cual se centra en la entrega incremental de valor, la colaboración constante con el cliente y la flexibilidad para adaptarse a cambios en los requerimientos. Scrum divide el trabajo en ciclos cortos y repetitivos llamados sprints, que permiten entregar funcionalidades utilizables en periodos breves, favoreciendo el aprendizaje continuo y la mejora progresiva del sistema.

|  |  |  |
| --- | --- | --- |
| **Característica** | **Metodología Cascada** | **Metodología Scrum** |
| Enfoque | Secuencial y rígido, cada fase debe completarse antes de iniciar la siguiente. | Iterativo e incremental, el producto se desarrolla en ciclos cortos (sprints). |
| Flexibilidad | Baja: los cambios son costosos y difíciles de incorporar una vez iniciada la ejecución. | Alta: permite adaptar el desarrollo a cambios en los requerimientos durante el proyecto. |
| Entrega de valor | El producto se entrega completo al final del ciclo, lo que retrasa la validación con el cliente. | Se entregan incrementos funcionales en cada sprint, generando valor constante. |
| Riesgo | Elevado: si al final el producto no satisface al cliente, todo el esfuerzo previo puede perderse. | Reducido: la retroalimentación temprana permite corregir el rumbo en cada sprint. |
| Comunicación con el cliente | Limitada, generalmente solo al inicio y al final del proyecto. | Constante, con revisiones frecuentes y participación activa del Product Owner. |
| Adaptación a proyectos cambiantes | Poco recomendable en proyectos con requerimientos poco definidos o sujetos a cambios. | Ideal para proyectos donde las necesidades del cliente pueden evolucionar. |

En el caso del sistema desarrollado para la empresa Donde Gino, se determinó que la metodología Scrum era más adecuada debido a que el proyecto podía experimentar cambios en los requerimientos durante su desarrollo, especialmente en módulos críticos como la gestión de boletas, pagos e inventario. Con Cascada, por ejemplo, estos ajustes habrían significado rehacer fases completas y aumentar significativamente los tiempos y costos.

Scrum, en cambio, permitió dividir el proyecto en sprints de corta duración, entregar avances funcionales de manera periódica y recibir retroalimentación constante de la empresa. Esto favoreció que la solución se fuera ajustando a las necesidades reales del negocio y que el cliente tuviera siempre visibilidad del progreso.

Las fases principales de Scrum y sus funciones dentro del proyecto son:

**1. Planificación del Producto (Product Backlog)**

* Función: recopilar, priorizar y mantener todas las funcionalidades, requerimientos y mejoras que se esperan del sistema.
* En esta fase, el Product Owner define las historias de usuario, criterios de aceptación y prioridades de acuerdo con las necesidades del negocio.

**2. Planificación del Sprint (Sprint Planning)**

* Función: seleccionar, desde el Product Backlog, los elementos que serán trabajados durante el sprint (generalmente de 2 a 4 semanas).
* El equipo acuerda un objetivo de sprint y planifica cómo alcanzarlo, definiendo tareas concretas y estimando su esfuerzo.

**3. Ejecución del Sprint (Sprint Execution)**

* Función: desarrollo de las funcionalidades seleccionadas para el sprint.
* Los integrantes trabajan de forma colaborativa en frontend, backend y base de datos, aplicando integración continua y revisando el progreso diariamente.

**4. Reuniones Diarias (Daily Scrum)**

* Función: sincronizar al equipo, identificar impedimentos y planificar el trabajo de las próximas 24 horas.
* Cada integrante responde brevemente: ¿Qué hice ayer?, ¿Qué haré hoy?, ¿Qué obstáculos tengo?.

**5. Revisión del Sprint (Sprint Review)**

* Función: presentar al cliente y a los interesados el incremento desarrollado durante el sprint.
* Se reciben comentarios, sugerencias y retroalimentación, que pueden incorporarse en el Product Backlog.

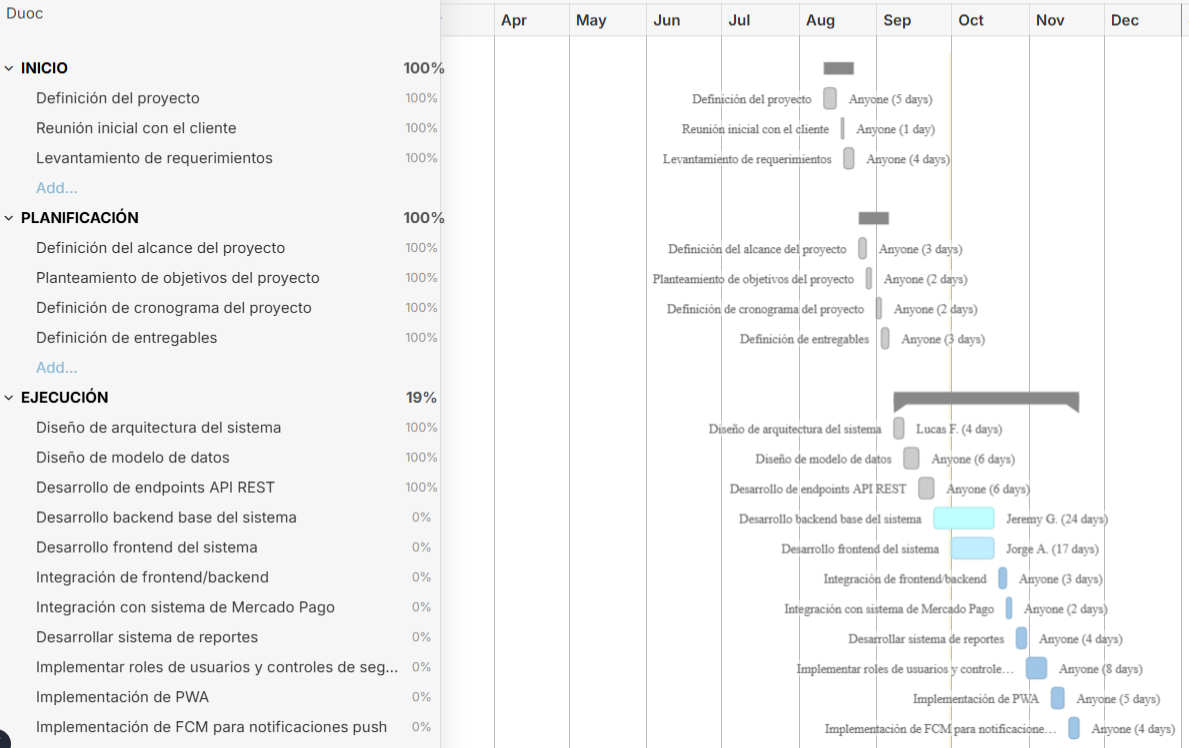
**6. Retrospectiva del Sprint (Sprint Retrospective)**

* Función: analizar cómo trabajó el equipo durante el sprint, identificar mejoras en la forma de colaborar y definir acciones concretas para el próximo ciclo.
* Favorece la mejora continua del proceso de trabajo.

La aplicación de la metodología **Scrum** en CleanFlow permitirá al equipo de desarrollo adaptarse a los cambios, mantener una comunicación constante con el cliente y entregar valor de forma temprana y continua, asegurando que el resultado final cumpla con los objetivos del proyecto y las expectativas del negocio.

## Creación de cronograma asociado al Proyecto (Carta Gantt)

Con el fin de planificar de manera estructurada las actividades del proyecto CleanFlow, se elaboró una carta Gantt que refleja las principales fases de ejecución, control, monitoreo y cierre. En este cronograma se establecieron los plazos estimados de cada tarea, los responsables asignados y las dependencias entre actividades, lo que permitió visualizar el flujo de trabajo y asegurar que cada etapa se cumpliera en los tiempos acordados.



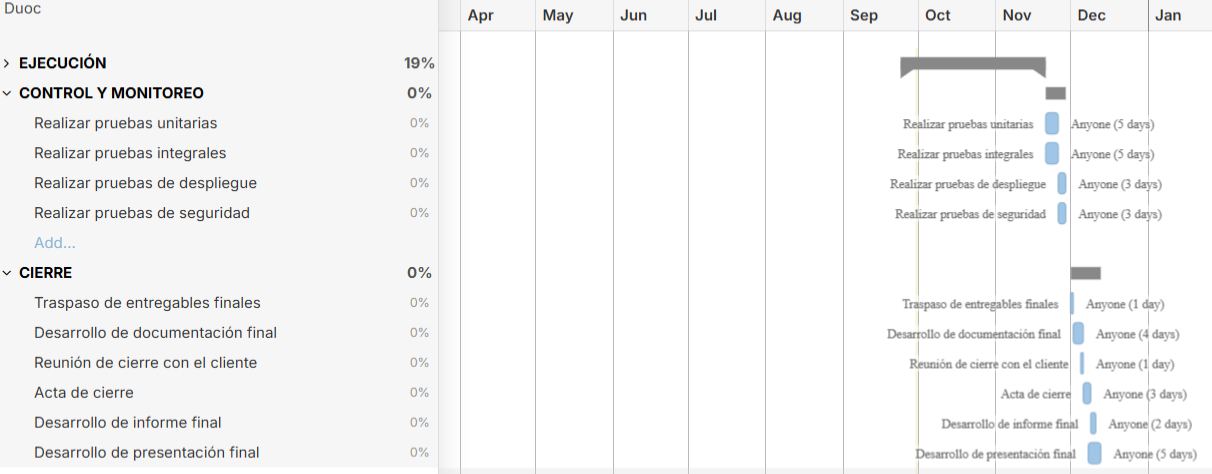
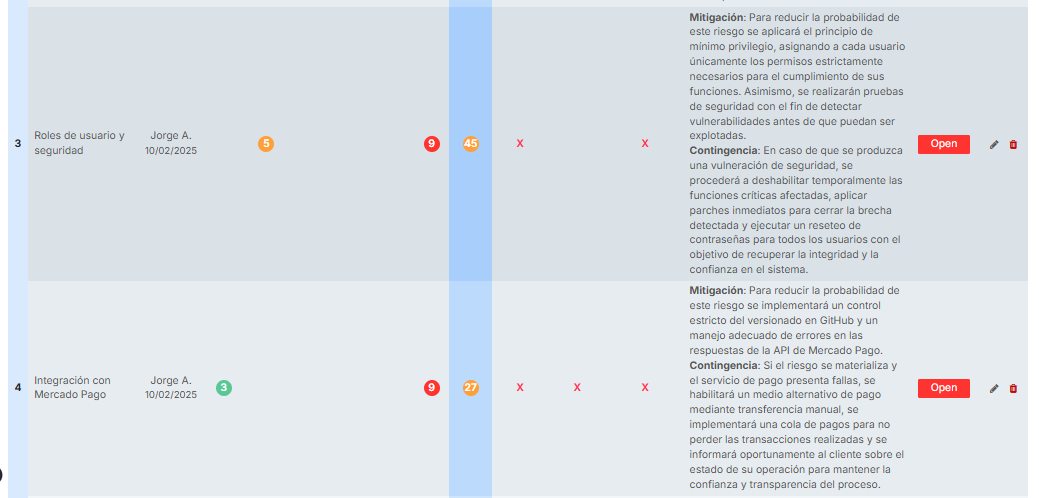


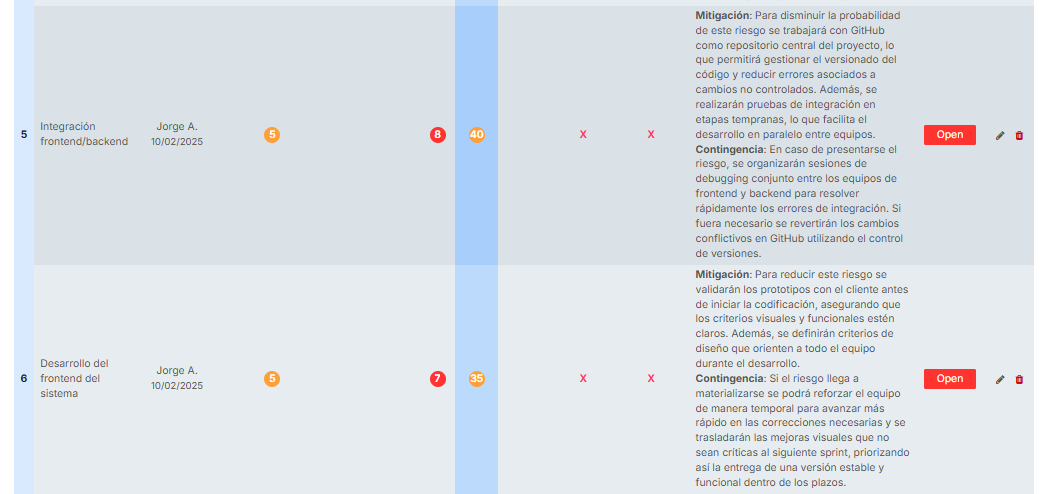
Figura 1. Carta Gantt

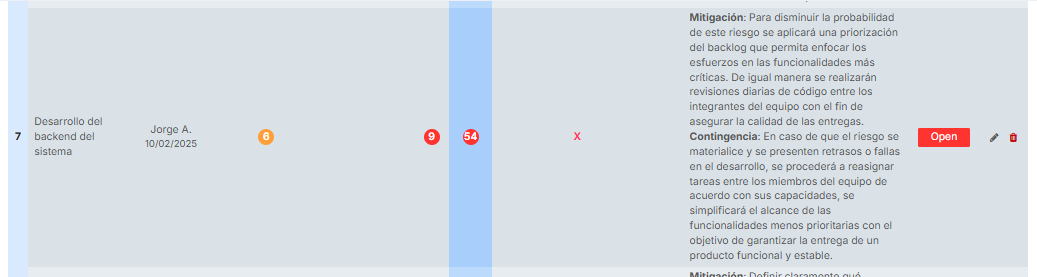
## Riesgos Asociados al Proyecto

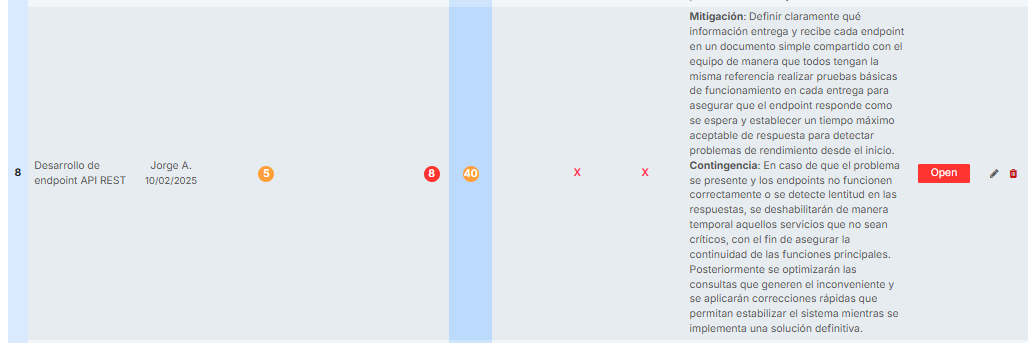
Junto con la planificación temporal del proyecto, se identificaron los principales riesgos asociados al desarrollo e implementación del sistema en la empresa **Donde don Gino**. Para ello se elaboró una matriz de riesgos que contempla la descripción de cada riesgo, su probabilidad de ocurrencia, el impacto que podría generar en el proyecto, las medidas de mitigación y el plan de contingencia en caso de materializarse. Esta herramienta permitió anticiparse a posibles problemas y definir estrategias que garantizaron la continuidad del trabajo y la entrega de un sistema confiable y alineado con los objetivos de la organización.











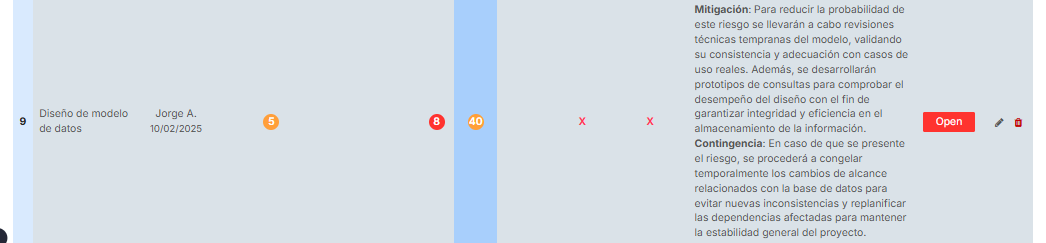


Figura 2. Tabla de Riesgos

## Implementación del Proyecto

### Diseño y Arquitectura de la solución (Caso de uso de Software o plataforma de gestión)

* Describir en profundidad el modelo de datos que permiten soportar los requerimientos de la organización de acuerdo al diseño definido que sea escalable en el tiempo.

### Requerimientos técnicos

* Detallar cada uno de los dispositivos, insumos y herramientas que se utilizaran para la implementación del proyecto.

### Desarrollo de la solución

* Se recomienda insertar capturas de pantalla con una breve explicación de cada uno de los pasos realizados en cada proceso de programación.
* Evidenciar la solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos del proyecto.
* Integrar los distintos componentes de la solución de software utilizando técnicas que permitan sistematizar el proceso de desarrollo y mantenimiento, asegurando el logro de los objetivos del proyecto.

### Resultados de la solución

* En este apartado debe insertar todas las pruebas que fueron ejecutadas para chequear los resultados de la solución. Puede incorporar capturas de pantalla, videos o demostración en tiempo real de la solución.
* Los resultados deben ser mostrados en la exposición final frente a la comisión.

## Definición de Recursos y Costos asociados al Proyecto

# CONCLUSIÓN

Sintaxis final del documento que permite resumir y/o llegar a conclusiones desde la detección del problema o necesidad, pasando por el desarrollo del proyecto hasta las pruebas finales de funcionamiento.

# BIBLIOGRAFÍA

Para la bibliografía considerar Norma APA 7ma Edición.

# 

# ANEXOS

Debe incorporar como anexos en documento como archivo Word, todo tipo de información que complemente el proyecto.

En la entrega final del proyecto, deben enviar todos los archivos utilizados en la implementación y desarrollo del mismo. El docente mencionará unas clases antes de la entrega final dicho listado a entregar.

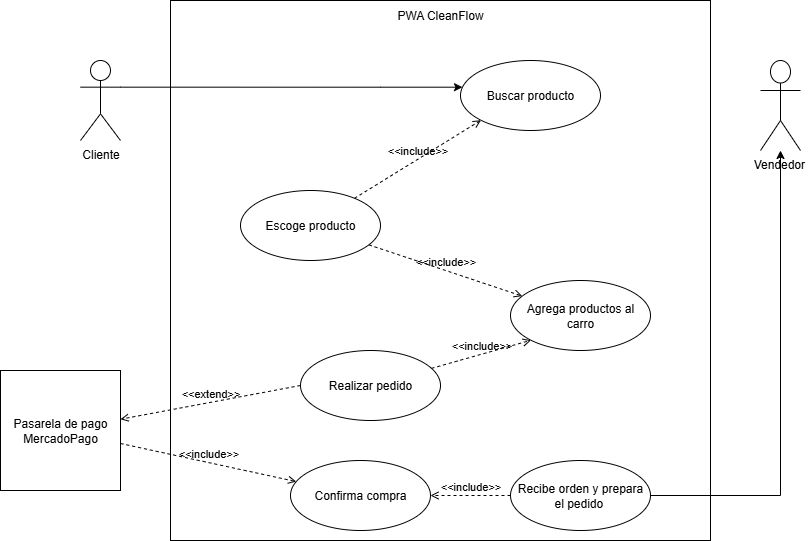


Figura 3. Diagrama de casos de uso

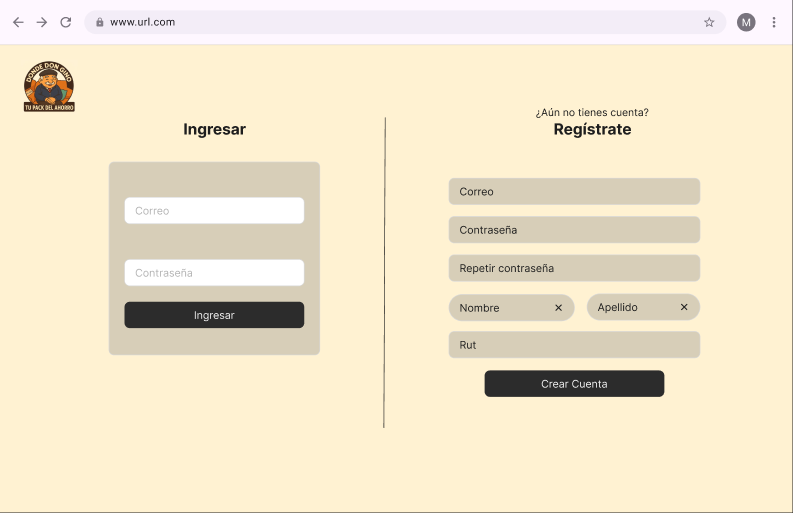
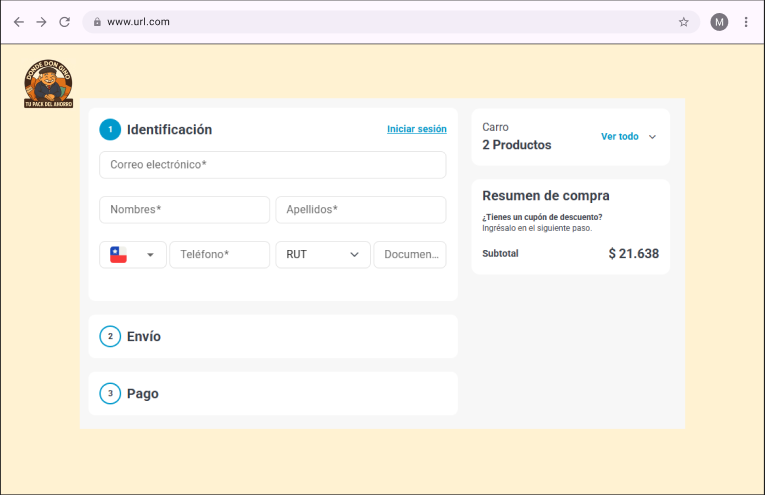


Figura 4. Prototipo de interfaz

Figura 5. Prototipo de interfaz

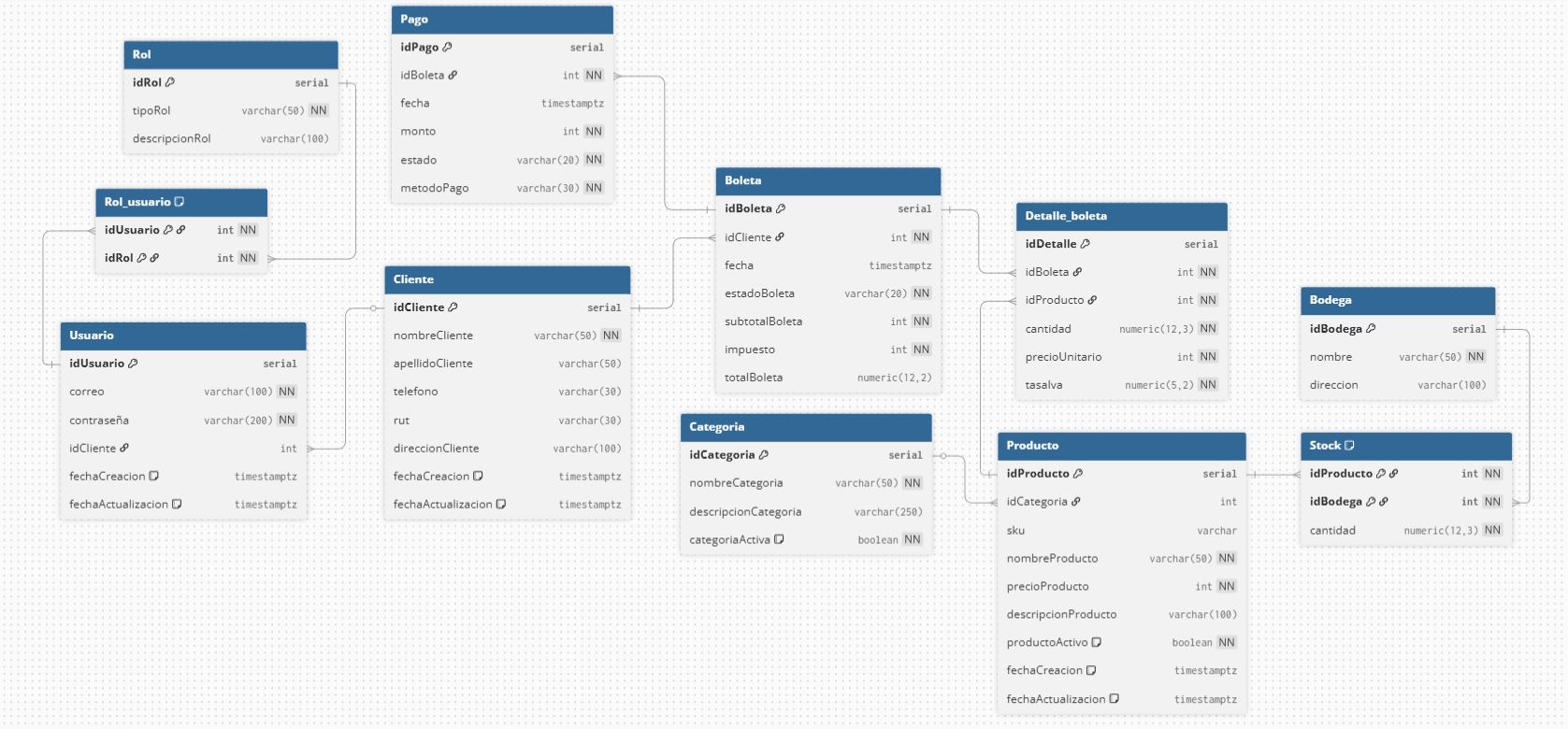


Figura . Diagrama Base de datos

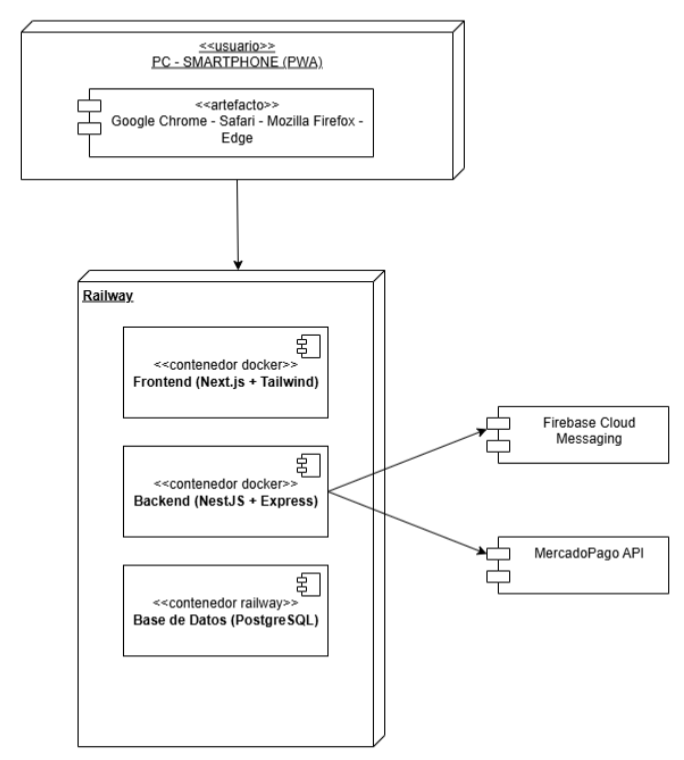


Figura . Diagrama Arquitectura de Software