**Plataforma Integral de Gestión de Clientes y Órdenes de Transporte.**

|  |  |  |
| --- | --- | --- |
| Integrantes | : | Martín Hidalgo  Jonatan Sandoval  Vanessa Salazar |
| Carrera | : | Ingeniería en informática |
| Profesor | : | Jonathan López Acevedo |
| Fecha de entrega | : | Diciembre 2024 |

Contenido

[Resumen ejecutivo (español). 5](#_Toc183471828)

[Introducción. 7](#_Toc183471829)

[Descripción del problema. 8](#_Toc183471830)

[Flujo actual SetraLog 9](#_Toc183471831)

[Solución al problema. 10](#_Toc183471832)

[Objetivo general. 13](#_Toc183471833)

[Objetivos específicos. 13](#_Toc183471834)

[Competencias del perfil de egreso. 15](#_Toc183471835)

[Acta de constitución de proyecto. 16](#_Toc183471836)

[Información general 16](#_Toc183471837)

[Propósito del Proyecto 16](#_Toc183471838)

[Aprobaciones 19](#_Toc183471839)

[Asignación de roles. 20](#_Toc183471840)

[Roles en el proyecto. 20](#_Toc183471841)

[Responsables de cada rol. 21](#_Toc183471842)

[Metodología. 22](#_Toc183471843)

[Fase 1 - Levantamiento de Requerimiento y Entendimiento del Negocio 24](#_Toc183471844)

[Fase 2 - Diseño, desarrollo y pruebas. 25](#_Toc183471845)

[Fase 3 – Implementación y marcha blanca. 26](#_Toc183471846)

[Carta Gantt. 27](#_Toc183471847)

[Implementación del proyecto. 30](#_Toc183471848)

[Levantamiento de requerimientos. 30](#_Toc183471849)

[Flujos de Procesos. 32](#_Toc183471850)

[Registro de usuarios/empresas. 32](#_Toc183471851)

[Generación de órdenes de transporte. 33](#_Toc183471852)

[Seguimiento de Orden 34](#_Toc183471853)

[Casos de uso. 35](#_Toc183471854)

[Modelo 3 capas. 45](#_Toc183471855)

[Arquitectura física. 47](#_Toc183471856)

[Product Backlog. 49](#_Toc183471857)

[Desarrollo de tareas. 53](#_Toc183471858)

[Base de datos. 56](#_Toc183471859)

[Microsoft Azure data base. 57](#_Toc183471860)

[Modelo lógico. 58](#_Toc183471861)

[Modelo relacional. 59](#_Toc183471862)

[Código SQL. 60](#_Toc183471863)

[Desarrollo BackEnd. 62](#_Toc183471864)

[FrontEnd. 63](#_Toc183471865)

[Conclusión. 68](#_Toc183471866)

[Mejoras a futuro. 69](#_Toc183471867)

[Acrónimos. 70](#_Toc183471868)

[Bibliografía. 72](#_Toc183471869)

[Anexos. 73](#_Toc183471870)

Índice de ilustraciones

[Ilustración 1- Flujo de proceso actual 9](#_Toc183468041)

[Ilustración 2 - Flujo optimizado de procesos 12](#_Toc183468042)

[Ilustración 3 - Desarrollo incremental 23](#_Toc183468043)

[Ilustración 4 -Diagrama Gantt con actividades 28](#_Toc183468044)

[Ilustración 5 - Diagrama Gantt con línea temporal. 29](#_Toc183468045)

[Ilustración 6 - Flujo registro usuario/empresa 32](#_Toc183468046)

[Ilustración 7- Flujo generación de transporte 33](#_Toc183468047)

[Ilustración 8 - Seguimiento de orden 34](#_Toc183468048)

[Ilustración 9 - Caso de uso -Registro de Usuario 35](#_Toc183468049)

[Ilustración 10 - Caso de uso - Inicio de Sesión 36](#_Toc183468050)

[Ilustración 11- Caso de uso - Recuperación de Contraseña 37](#_Toc183468051)

[Ilustración 12 - Caso de uso - Creación de empresa 38](#_Toc183468052)

[Ilustración 13 - Caso de uso - Edición de Datos de Empresa 39](#_Toc183468053)

[Ilustración 14 - Caso de uso - Generación de Orden de Transporte 40](#_Toc183468054)

[Ilustración 15 - Caso de uso - Seguimiento de Orden 41](#_Toc183468055)

[Ilustración 16 - Caso de uso - Notificaciones 42](#_Toc183468056)

[Ilustración 17 - Caso de uso - Gestión de Órdenes 43](#_Toc183468057)

[Ilustración 18 - Caso de uso - Generación de reportes 44](#_Toc183468058)

[Ilustración 19 - Modelo de 3 capas 46](#_Toc183468059)

[Ilustración 20 - Infraestructura en Microsoft Azure 48](#_Toc183468060)

[Ilustración 21 - implementación de Web APP Services 56](#_Toc183468061)

[Ilustración 22 - Grupo de recursos de Azure 57](#_Toc183468062)

[Ilustración 23 - Modelo lógico de datos 58](#_Toc183468063)

[Ilustración 24 - Modelo relacional de datos 59](#_Toc183468064)

[Ilustración 25 - Generación de tablas mediante DDL 60](#_Toc183468065)

[Ilustración 26 - Estructura de tablas SQL Managent Studio 61](#_Toc183468066)

[Ilustración 27 - FrameWork Express 62](#_Toc183468067)

[Ilustración 28 - FrontEnd en HTML 63](#_Toc183468068)

[Ilustración 29- Estilo visual en CSS 64](#_Toc183468069)

[Ilustración 30 - P ágina web de inició de sesión 65](#_Toc183468070)

Índice de tablas

[Tabla 1- Responsables de cada Rol 21](#_Toc183468896)

[Tabla 2 - Requerimientos funcionales / No Funcionales 30](#_Toc183468897)

[Tabla 3 - Caso de uso - CU-01 35](#_Toc183468898)

[Tabla 4 - Caso de uso - CU-02 36](#_Toc183468899)

[Tabla 5 - Caso de uso - CU-03 37](#_Toc183468900)

[Tabla 6 - Caso de uso - CU-04 38](#_Toc183468901)

[Tabla 7 - Caso de uso - CU-05 39](#_Toc183468902)

[Tabla 8 - Caso de uso - CU-06 40](#_Toc183468903)

[Tabla 9 - Caso de uso - CU-07 41](#_Toc183468904)

[Tabla 10 - Caso de uso - CU-08 42](#_Toc183468905)

[Tabla 11 - Caso de uso - CU-09 43](#_Toc183468906)

[Tabla 12 - Caso de uso - CU-10 44](#_Toc183468907)

[Tabla 13 - Requerimientos 50](#_Toc183468908)

[Tabla 14 - Tareas primer entregable 53](#_Toc183468909)

[Tabla 15 - Tareas segundo entregable 54](#_Toc183468910)

[Tabla 16 - Tareas tercer entregable 55](#_Toc183468911)

Resumen ejecutivo (español).

|  |
| --- |
| **Título del Proyecto:** “Plataforma Integral De Gestión de Cliente y Órdenes de Transporte”.  **Objetivos:** Desarrollar una solución tecnológica basada en una plataforma web, que centralice y automatice la gestión de clientes y órdenes de transporte, para mejorar la eficiencia operativa de Setralog, una empresa dedicada al transporte terrestre de carga.  **Contexto:** Setralog, fundada en 2018, actualmente enfrenta limitaciones significativas debido a la gestión manual de sus operaciones, basadas en hojas de cálculos y correos electrónicos. Esta metodología ha generado ineficiencias como:   * Incremento del 45% en tiempos de respuesta en comparación a la competencia. * Tasa de errores del 18% en la creación de órdenes. * Pérdida de 13 clientes en 2024 debido a la baja calidad del servicio, entre otros.   Estos problemas impactan negativamente en la reputación, productividad y competitividad de la empresa.  **Solución Propuesta**  La Asignatura Portafolio de Título (APT), propone una plataforma web que permitirá:  **Centralizar la Información:** Registro y gestión unificada de la información de datos de clientes y órdenes de transporte.  **Acceso autónomo:** Clientes y colaboradores podrán gestionar solicitudes de manera independiente y sin dificultades.  **Optimización de procesos:** Reducción de errores de registro al 5% y disminución de los tiempos de creación y procesamiento de órdenes al 10%.  **Beneficios Esperados**  Incremento de la satisfacción y fidelización de clientes mediante una interfaz amigable para los usuarios.  Mejora en la trazabilidad y calidad del servicio ofrecido.  Reducción de costos operativos y optimización de recursos humanos.    **Metodología**  El proyecto se desarrollará utilizando la metodología iterativa incrementada, específicamente el marco ágil, que permite dividir el desarrollo en fases entregables.  Este enfoque asegura la entrega gradual de valor, adaptándose a cambios y priorizando funcionalidades críticas.  Fases Principales:   1. **Levantamiento de Requerimiento:** Documentación de necesidades del cliente y análisis de procesos actuales. 2. **Diseño y Desarrollo:** Implementación en incrementos funcionales (gestión de clientes, creación de órdenes y seguimiento automatizado). 3. **Prueba e Implementación:** Validación, marcha blanca y ajustes finales.   **Competencias**  El desarrollo de Proyecto APT permite a los integrantes aplicar competencias técnicas (programación, base de datos, seguridad), habilidades blandas (trabajo en equipo, resolución de problemas) y de gestión (planificación y liderazgo), alineándose con las demandas del mercado y fortaleciendo su perfil profesional. |

Introducción.

El proyecto APT (Asignatura de portafolio de título) se desarrolla en el marco de la carrera de Ingeniería en Informática y responde a la necesidad de implementar una solución tecnológica basada en software que ayude a automatizar y modernizar los procesos de administración de clientes y órdenes de transporte de la empresa Setralog, dedicada al transporte terrestre de carga en general, tales como electrodomésticos, herramientas, textiles, etc. Así también transporta carga especializada, como medicamentos, alimentos refrigerados y químicos no peligrosos.

Setralog, fundada en 2018 por la ingeniera en logística Patricia González, quien actualmente se desempeña como gerente general, inició su trayectoria como una pyme enfrentando diversos desafíos, entre los más significativos se encontraba la gestión eficiente de la información de clientes y órdenes de transporte. En sus primeros años, la administración de la empresa se apoyaba en hojas de cálculo en Excel, mientras que la comunicación con los clientes, especialmente para solicitar órdenes de transporte, se hacía por correo electrónico.

Esta combinación de métodos permitió a Setralog dar sus primeros pasos en el sector, aunque también evidenció la necesidad de optimizar procesos para adaptarse a un entorno empresarial en constante evolución.

En la actualidad, Setralog ha experimentado un crecimiento tanto en capacidad operativa como en capacidad de transporte. Con el paso de los años, ha mantenido una cartera de clientes a los que presta sus servicios (72 clientes en el año 2024), sin embargo, sigue enfrentando desafíos derivados de la gestión manual de órdenes e información de clientes, ya que continúa utilizando hojas de cálculo y correo electrónico para registrar y administrar la información.

Esto permitirá que tanto los colaboradores de Setralog y sus clientes, puedan autogestionar sus datos y darse de alta como clientes, es decir que ellos mediante el mismo sistema web puedan rastrear sus órdenes y que a su vez se le permita la creación de las solicitudes de transporte de manera autónoma por parte de los clientes.

Descripción del problema.

La problemática principal que enfrenta SetraLog actualmente se basa en el hecho de que siguen usando Excel y el correo de forma manual para registrar y administrar la información; esta forma de trabajo es lenta, y poco eficaz, propensa a errores, y genera un aumento en los tiempos de respuesta. Todas estas ineficiencias han llevado a la pérdida de 13 clientes en lo que va del 2024. SetraLog hoy en día se ve en la necesidad de optimizar sus procesos operativos.

El enfoque que SetraLog tiene hoy no solo ralentiza las operaciones, generando un 45% más de retraso en los tiempos de entrega en comparación con estándares competitivos, sino que también incrementa el riesgo de errores en el ingreso de datos, lo que resulta en una tasa de error del 18%. Estas deficiencias impactan negativamente en la eficiencia operativa, la calidad del servicio y la satisfacción de los clientes.

Este método carece de unificación y eficiencia, afectando tanto a la empresa como a sus clientes. Lo que ha evidenciado en retraso en la entrega de pedidos, tiempos excesivos en procesos administrativos, errores de despacho y tipeo, pérdida de imagen ante la competencia lo que ha desencadenado en una disminución de clientes.

Flujo actual SetraLog

En este flujo se aprecian los procesos involucrados en una solicitud de transporte.

Ilustración - Flujo de proceso actual

Diagrama

Descripción generada automáticamente

Nota: El flujo representado en la figura muestra todos los procesos que debe pasar el flujo de una orden y los actores involucrados (Elaboración propia 2024)

Detalles de flujo original en el Anexo “[Flujo de proceso original](#Flujo_de_proceso_original)”

Solución al problema.

La propuesta es centralizar todos los datos en un sistema unificado que pueda almacenar y procesar grandes volúmenes de información, lo que contribuirá a la eficiencia operativa y reducirá significativamente los errores derivados del ingreso manual de datos. La unificación y centralización permitirá a Setralog contar con un flujo de trabajo más ágil, disminuir la carga administrativa de los empleados y mejorar la trazabilidad de los procesos logrando así una mejor calidad en los servicios ofrecidos.

El proyecto APT tiene como objetivo abordar la ineficiencia en la gestión de clientes y órdenes de transporte de la empresa Setralog, que actualmente opera con métodos manuales, como hojas de cálculo en Excel y correos electrónicos, para registrar y organizar información. Este enfoque genera altos tiempos de procesamiento y dificulta la optimización operativa.

Esta solución está diseñada para automatizar, optimizar y centralizar los procesos operativos, mejorando significativamente la eficiencia y la calidad del servicio.

La solución cuenta con componentes clave como:

1. **Centralización de datos**

* Implementación de un sistema unificado para almacenar y gestionar información de clientes y órdenes de transporte.
* Sustitución de hojas de cálculo Excel y correos electrónicos por una base de datos centralizada, unificada y segura.

1. **Automatización de Procesos**

* Desarrollo de módulos específicos para**: g**estión de clientes; Registro, actualización y consulta de datos en tiempo real.
* Gestión de órdenes de transporte: creación, asignación y seguimiento automatizado de las órdenes.
* Integración de herramientas en la nube (Azure, AWS, Google Cloud) para garantizar alta disponibilidad y escalabilidad

1. **Interacción Autónoma**

Diseño de una interfaz web intuitiva que permita los clientes:

* Registrarse como usuarios.
* Crear y gestionar sus solicitudes de transporte.
* Consultar el estado de sus órdenes en tiempo real.
* Reducción de la dependencia del soporte administrativo.

1. **Reducción de Errores**

* Implementación de validaciones automáticas en el ingreso de datos para minimizar errores humanos.
* Mejora en la precisión y consistencia de la información registrada.

1. **Mejora en la trazabilidad**

* Sistema de seguimiento en tiempo real para monitorear el estado de los pedidos.
* Registro histórico para análisis y optimización de procesos futuros.

1. **Seguridad y Confidencialidad**

* Autenticación de usuarios.
* Encriptación de Password.
* Uso de protocolo HTTPS y SSL.

Esta solución moderniza y automatiza los procesos de Setralog, permitiendo una gestión más eficiente, precisa y alineada con las demandas del mercado actual. En base a las mejoras el flujo optimizado de los procesos queda de esta manera.

Ilustración - Flujo optimizado de procesos

Gráfico, Diagrama

Descripción generada automáticamente

Nota: El flujo representado en la figura muestra la optimización de los procesos a los cuales se espera llegar con la implementación de esta solución. (Elaboración propia 2024)

Detalles de flujo optimizado en el Anexo “[Flujo de proceso optimizado”](#Flujo_de_proceso_optimizado)

Objetivo general.

Desarrollar una plataforma integral que centralice y automatice la gestión de clientes y órdenes de transporte para mejorar la eficiencia operativa, también mejorar la experiencia del cliente con una interfaz intuitiva y funcional que facilite la interacción entre los clientes y Setralog, mejorando la satisfacción y fidelización de los clientes; aumentando así la eficiencia, mediante una solución tecnológica que haga el uso eficiente de los recursos y mejore los tiempos de respuesta.

Al alcanzar estos objetivos se espera reducir de un 45% a un 10% los tiempos relacionados con la generación y procesamiento de la orden de transporte además de mitigar por completo los errores por ingreso de datos manuales.

Objetivos específicos.

* Desarrollo de módulo de gestión de clientes: Este módulo permite registrar, actualizar y consultar información de los clientes de manera rápida y segura. Esto evitará información errónea, ya que él propio cliente se creará como usuario en el sistema.
* Desarrollo de módulo de orden de transporte: Este módulo permite la creación, asignación y seguimiento de las órdenes de transporte. Esto evitará la alta demora al generar una orden de transporte ya que ahora el propio cliente podrá generar la orden con los datos suministrados por él.
* Integración de plataforma Cloud: implementar las soluciones de almacenamiento y procesamiento de datos en la nube. (Azure, Google Cloud, Amazon Web Services según la factibilidad del cliente).
* Seguridad de la información: Implementar medidas de seguridad para proteger la información de los clientes y órdenes de transporte, utilizando las buenas prácticas de la industria y estándares de seguridad de la información.
* Capacitación y Soporte: Se proveerá capacitación para el uso eficiente de la plataforma, además de prestar soporte técnico al desarrollo en caso de alguna incidencia en su funcionamiento.

Competencias del perfil de egreso.

Los egresados de la carrera de Ingeniería en Informática deben desarrollar un conjunto de competencias fundamentales a lo largo de su formación, que son esenciales para su desempeño profesional y la ejecución exitosa de proyectos. Estas competencias se dividen en tres categorías principales:

1. **Competencias Técnicas:**

* Dominio de lenguajes de programación y desarrollo de software.
* Gestión de bases de datos y administración de sistemas.
* Implementación de soluciones tecnológicas innovadoras.

1. **Competencias Blandas:**

* Habilidades de comunicación efectiva y trabajo en equipo.
* Resolución de problemas y pensamiento crítico.
* Adaptabilidad y aprendizaje continuo.

1. **Competencias de Gestión:**

* Planificación y ejecución de proyectos tecnológicos.
* Gestión de recursos y liderazgo de equipos multidisciplinarios.
* Enfoque en la calidad, seguridad y cumplimiento de objetivos estratégicos.

Estas competencias aseguran que los egresados estén preparados para enfrentar los desafíos del sector tecnológico y contribuir de manera efectiva al desarrollo de soluciones informáticas innovadoras.

Acta de constitución de proyecto.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Proyecto: Proyecto APT “Plataforma Integral de gestión de clientes y órdenes de transporte”  Información general  **Proyecto**: **Plataforma Integral de gestión de clientes y órdenes de transporte.**  **Fecha**: Agosto 2024  **Versión**: 1.0  **Duración Total del Proyecto**: 90 días  **Responsable del Proyecto:** Jonatan Sandoval (Product Owner)  **Cliente:** Setralog (Patricia González, Gerente General)  Propósito del Proyecto  El objetivo de este proyecto es desarrollar una plataforma web integral que centralice y automatice la gestión de clientes y órdenes de transporte para la empresa Setralog, optimizando sus procesos operativos. La solución está diseñada para minimizar errores de registro, mejorar los tiempos de respuesta y proporcionar una experiencia más eficiente tanto para los colaboradores como para los clientes.  Mediante la implementación de tecnologías modernas y un enfoque sólido en la seguridad de la información, este proyecto busca no solo aumentar la competitividad de Setralog en el mercado, sino también fortalecer la fidelización de sus clientes al ofrecer un servicio más confiable y ágil.  Objetivos del proyecto  Objetivo general  Implementar una solución de software basado en web para centralizar, automatizar y unificar la gestión de cliente y órdenes de transporte para mejorar la eficiencia operativa y la experiencia del cliente Setralog.  Objetivos específicos  Desarrollar una solución web integral que incluya un módulo de gestión de clientes, permitiendo registrar, actualizar y consultar información de manera eficiente y centralizada. Además, implementar un módulo autónomo para la creación, seguimiento y asignación de órdenes de transporte, optimizando estos procesos clave. La solución estará basada en la web, con almacenamiento y procesamiento de datos completamente en la nube, lo que garantizará escalabilidad, seguridad y eficiencia en la gestión de la información.  Adicionalmente, se proporcionará capacitación y soporte continuo a los usuarios para asegurar el uso óptimo y eficiente de la plataforma, mejorando la productividad y la calidad del servicio.  Factores de éxito del proyecto  **Facilidad de uso**: Una interfaz sencilla e intuitiva para usuarios y clientes.  **Confiable:** sistema estable y sin interrupciones.  **Escalabilidad:** capacidad de crecer bajo demanda.  **Capacitación:** los usuarios serán capacitados para el uso del aplicativo.  **Costo:** cumple con el presupuesto establecido.  **Valor percibido:** el sistema permite ahorrar tiempo, disminuir errores y aumentar la satisfacción del cliente.  **Tecnología**: el sistema utiliza tecnologías actuales, estables y con amplia documentación de soporte.  **Metodología de desarrollo:** Se utilizará metodología Agile Scrum que permite entregas parcializadas e implementando mejora continua.  Interesados claves   |  |  | | --- | --- | | **Rol asignado** | **Nombre** | | Cliente | Patricia González / Setralog | | Product owner / jefe de proyecto | Jonatan Sandoval | | Scrum Máster/Coordinador del Proyecto | Vanessa Salazar | | Development Team Leader / Líder del Equipo de desarrollo: | Martin Hidalgo |   Riesgos identificados  Los principales riesgos identificados son:   * **Riesgo de retrasos:** Posibles demoras en el desarrollo debido a cambios en los requerimientos o problemas técnicos. * **Riesgo de aceptación de la plataforma:** Baja aceptación por parte de los usuarios, si no se proporciona la capacitación adecuada. * **Riesgo técnico:** Posibles dificultades con la integración de tecnologías en la nube o problemas de seguridad.   Se implementarán medidas preventivas para mitigar estos riesgos y garantizar la entrega del proyecto a tiempo.  Cronograma  Hito/Fase 1  Levantamiento de Requerimiento y Análisis de Negocio.  **Actividades principales**   * Reuniones con Stakeholders para definir alcance, objetivos y requisitos del proyecto. * Levantamiento de requerimiento funcionales y no funcionales. * Análisis de los procesos actuales de Setralog y desarrollo de diagramas de flujo. * Documentación detallada de los casos de uso y necesidades de integración con otros sistemas. * Revisión y validación de los requisitos con el cliente para asegurar el alineamiento con las expectativas.   **Duración estimada**   * 20 días   Hito/Fase 2  Levantamiento de Requerimiento y Análisis de Negocio.  **Actividades principales**   * Diseño de la arquitectura del sistema y elecciones de tecnologías. * Diseño de la base de datos y creación de los modelos lógicos y relaciones. * Desarrollo de prototipos de la interfaz de usuario y presentación para retroalimentación. Desarrollo del backend y frontend, integración con la base de datos. * Pruebas unitarias y de integración, corrección de errores detectados.   **Duración estimada**  55 días  Hito/Fase 3  Levantamiento de Requerimiento y Análisis de Negocio.  **Actividades principales**   * Preparación del entorno de producción y configuración de la infraestructura de la nube. * Carga inicial de datos y pruebas de integración del sistema en el entorno de producción. Marcha blanca. * Capacitación del personal de Setralog en el uso de la plataforma. Recopilación de Feedback y ajustes finales del sistema. * Finalización de la documentación técnica y de usuario. * Evaluación del proyecto, revisión de objetivos alcanzados y cierre formal   **Duración estimada**  15 días  Presupuesto  El presupuesto asignado para este proyecto es de 20 millones. Aprobaciones Esta acta de constitución es aprobada por las siguientes partes:  **Cliente**  Patricia González (Gerente General Setralog)  **Responsable del Proyecto**  Jonatan Sandoval  **Scrum Máster**  Vanessa Salazar  **Líder del Equipo de Desarrollo**  Martin Hidalgo |

Asignación de roles.

Es esencial definir roles dentro de un proyecto para garantizar el correcto funcionamiento del equipo a lo largo del desarrollo de distintas etapas. Estos roles definen las responsabilidades de cada miembro del equipo, lo que facilita la colaboración y asegura que las tareas se lleven a cabo de manera eficiente. Una clara asignación de roles es crucial para alcanzar los objetivos del proyecto y garantizar su éxito.

Roles en el proyecto.

* **Stakeholders / Interesados / Cliente:** Son los patrocinadores del proyecto, que pueden incluir al cliente directo o a otros inversionistas interesados en su éxito. Su papel principal es proporcionar los requisitos, expectativas, prioridades y retroalimentación sobre el producto. También validan que el proyecto cumpla con las necesidades del negocio y colaboran estrechamente con el Product Owner para asegurar que las prioridades del proyecto se alineen con sus intereses.
* **Product Owner / jefe de Proyecto:** Es el responsable del proyecto desde la perspectiva de la empresa ejecutora y actúa como representante de los interesados del cliente final. Su rol principal es gestionar el Product Backlog, priorizando las tareas en función del valor que aportan al negocio. Además, se asegura de que el equipo Scrum comprenda correctamente los requisitos, tareas y metas del proyecto, y mantiene una comunicación constante con el cliente a través de la colaboración con los stakeholders.
* **Facilitador / Coordinador de Proyecto:** Actúa como facilitador y líder de servicio para el equipo Scrum. Su función es eliminar los impedimentos que puedan bloquear el progreso del equipo, promover la comunicación y fomentar la mejora continua dentro del equipo. Además, trabaja en estrecha colaboración con el Product Owner y el Development Team para asegurar que el proceso Scrum se cumpla correctamente.
* **Development Team / Equipo de Desarrollo:** Este equipo es el encargado de la construcción del proyecto o producto. Está compuesto por profesionales especializados en áreas como bases de datos, diseño, arquitectura o programación en diversos lenguajes. El equipo se autoorganiza y divide el trabajo en los entregables que se deseen generar para mejorar y completar el proyecto.

Responsables de cada rol.

Bajo este proyecto los responsables de cada rol quedan definida bajo la siguiente estructura.

Tabla - Responsables de cada Rol

|  |  |
| --- | --- |
| **Rol** | **Responsable** |
| StakeHolder/ Cliente | Equipo Setralog / Patricia González |
| Product Owner / jefe de proyecto | Jonatan Sandoval |
| Facilitador / coordinador de proyecto | Vanessa Salazar |
| Development Team Leader / líder equipo de desarrollo | Martín Hidalgo |

Nota: La tabla muestra el detalle de los roles y sus asignaciones (Elaboración propia 2024)

Metodología.

La metodología para el desarrollo del Proyecto APT se basará en la metodología iterativo-incremental.

¿Qué es el desarrollo iterativo?

El desarrollo iterativo es una metodología de desarrollo de software en la que el proyecto se divide en pequeñas iteraciones o ciclos. Cada iteración es un proceso de planificación, diseño, implementación y pruebas que se realiza en un periodo de tiempo específico. Al final de cada iteración, se entrega el software para que pueda ser probado y evaluado por los usuarios.

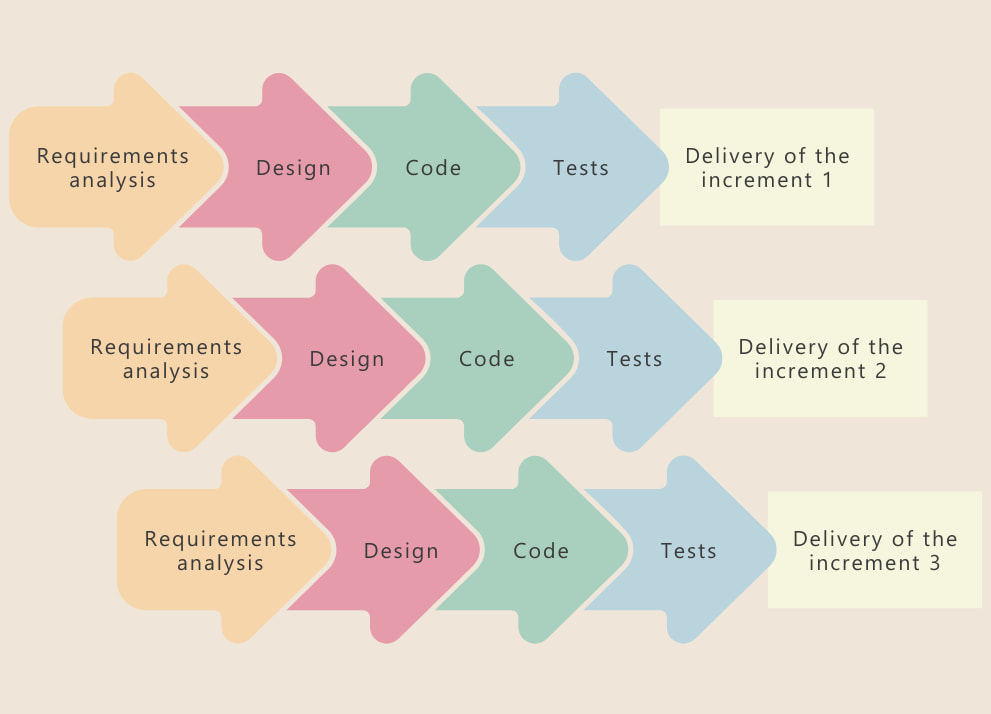
La idea detrás del desarrollo iterativo es que es más fácil identificar y corregir los errores y problemas en etapas tempranas del proyecto, en lugar de descubrirlos al final. Cada iteración proporciona una oportunidad para que el equipo de desarrollo y el cliente revisen y ajusten los requisitos y objetivos del proyecto en función de lo aprendido en la iteración anterior. De esta manera, el proyecto se va adaptando y evoluciona según las necesidades (Diferencias entre el Desarrollo Iterativo e Incremental, 2023/06/21).

El desarrollo iterativo es una metodología de desarrollo de software que organiza el proyecto en pequeñas iteraciones o ciclos. Cada iteración abarca un proceso completo que incluye planificación, diseño, implementación y pruebas, y se desarrolla en un periodo definido. Al finalizar cada iteración, el software se entrega a los usuarios para que puedan evaluarlo y proporcionar retroalimentación que permita ajustarlo según las necesidades identificadas.

La principal ventaja del desarrollo iterativo radica en su capacidad para identificar y corregir errores o problemas en las primeras etapas del proyecto. Esto evita que los inconvenientes se acumulen y sean descubiertos al final, cuando podrían ser más costosos de resolver. Además, cada iteración brinda la oportunidad de revisar y adaptar los requisitos y objetivos del proyecto en función de los aprendizajes obtenidos en el ciclo anterior.

De esta manera, el desarrollo iterativo permite que el proyecto evolucione de forma constante y se adapte progresivamente a las necesidades del cliente. Este enfoque garantiza que el producto final sea más ajustado a los objetivos iniciales y a las expectativas cambiantes, reduciendo riesgos y promoviendo una colaboración más efectiva entre los equipos de desarrollo y los usuarios.

Ilustración - Desarrollo incremental



Nota: La Ilustración muestra como es el ciclo del software basado en la metodologia incremental (Fuente: <https://intelequia.com/es/blog/post/qu%C3%A9-modelo-de-ciclo-de-vida-del-software-es-el-m%C3%A1s-adecuado-para-tu-proyecto,> 2024)

Fase 1 - Levantamiento de Requerimiento y Entendimiento del Negocio

1.1 Reuniones de Kick-Off con Stakeholders: Se realizará reuniones iniciales con los interesados clave, incluidos los usuarios finales, el Producto Owner y el equipo de desarrollo, para discutir el alcance y los objetivos principales del proyecto. Estas reuniones permitirán al equipo comprender las expectativas del cliente y obtener una visión clara de los requisitos.

1.2 Levantamiento de Requerimientos Funcionales y No Funcionales: Se organizarán sesiones para documentar detalladamente los requisitos del sistema. Esto incluye la identificación de funcionalidades críticas y restricciones.

1.3 Levantamiento de flujos y Procesos y Caso de Uso: Se desarrollarán diagramas que representan los procesos operativos actuales y cómo interactuarán los usuarios con el sistema. Los casos de uso servirán para definir claramente las interacciones previstas con el sistema y garantizar que las funcionalidades cubran todas las necesidades identificadas

Fase 2 - Diseño, desarrollo y pruebas.

2.1 Diseño de la Arquitectura del Sistema: Se crearán diagramas UML (diagrama de clases, de secuencia, de componentes, etc.) para representar las estructuras y la interacción de los diferentes componentes del sistema. Además, se diseñarán la base de datos, tanto en su modelo lógico como físico, optimizando la eficiencia y seguridad de las consultas.

2.2 Prototipo de la Interfaz de Usuario: Se desarrollarán prototipos de que reflejarán la interfaz del usuario. Estos se presentarán a los Stakeholders para obtener retroalimentación y realizar ajustes antes de pasar el desarrollo completo.

2.3 Product Backlog y Prioridades: Se creará un Product Backlog que contendrá todas las tareas necesarias para desarrollar el sistema. Las tareas se priorizarán según su importancia y se discutirán con el Product Owner para asegurarse de que las funcionalidades más críticas se desarrollan primero.

2.4 División de Sprint: El proyecto se dividirá en 3 sprint de 3 semanas cada uno. Cada sprint incluye la planificación, ejecución, revisión y retrospectiva. El equipo de desarrollo trabajara en un conjunto de tareas seleccionadas para cada sprint, asegurando que se entreguen incrementos funcionales del producto.

2.5 Desarrollo de la solución: Se codificarán los módulos principales del sistema, como la gestión de clientes y pedidos, integrando tecnologías en la nube (como Azure, AWS o Google cloud) para optimizar el almacenamiento y procesamiento de datos. También se implementarán buenas prácticas de programación para asegurar un código limpio, eficiente y fácil de mantener.

2.6 Pruebas operación y seguridad: Se realizarán pruebas exhaustivas para identificar fallas y vulnerabilidades en el sistema al final de cada sprint. Cualquier hallazgo será corregido y documentado antes de avanzará a las fases siguiente.

Fase 3 – Implementación y marcha blanca.

3.1 Implementación: se preparar ambiente de producción para ejecutar la solución ya programada, se cargan los datos necesarios y configuraciones, de esta manera queda lista para el paso a producción.

3.2 Marcha blanca: Una vez en producción se realiza una marcha blanca en donde el equipo de programación en conjunto con el cliente, monitorean el uso de software, se verifican los datos en caso de haber inconsistencias y se presta el soporte necesario a los usuarios para el uso del software.

Carta Gantt.

La carta Gantt es una herramienta crucial para controlar el desarrollo del proyecto. En ella se aprecian las tareas y duración del proyecto, además de servir de control para monitorear el avance del proyecto.

Es este caso el proyecto contempla un periodo de 90 días de inicio a fin, separado en 3 entregables o hitos cada uno con sus correspondientes tareas y tiempos de ejecución.

* Entregable 1 – 20 días
* Entregable 2 – 55 días
* Entregable 3 – 15 días

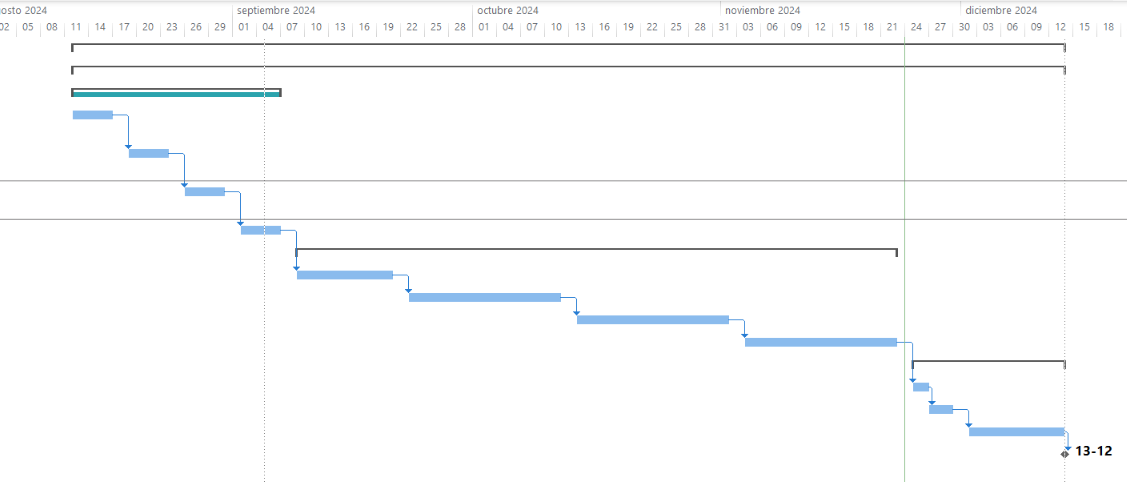
Ilustración 4 -Diagrama Gantt con actividades

Interfaz de usuario gráfica, Aplicación, Tabla, Excel

Descripción generada automáticamente

Nota: La ilustración muestra la cronología del proyecto según sus tareas y tiempos de ejecución (fuente: Elaboración propia, 2024)

Ilustración 5 - Diagrama Gantt con línea temporal.



Nota: la ilustración muestra la línea temporal de las tareas especificadas en la carta Gantt (fuente: Elaboración propia, 2024)

Implementación del proyecto.

Levantamiento de requerimientos.

Levantar los requerimientos es fundamentar para la planificación del proyecto. Los requerimientos son definidos en 2 categorías. Requerimientos Funcionales que describe las funciones que debe realizar el sistema para cumplir con las necesidades y el negocio. Y los requerimientos no funcionales, que describen las características de técnicas que debe tener el sistema las cuales impactan directamente en su desempeño, seguridad y escalabilidad.

A continuación, se enlista la lista de requerimientos funcionales y no funcionales que tener el sistema a desarrollar.

Tabla - Requerimientos funcionales / No Funcionales

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Requerimiento** | **Tipo** | **Actores Relacionados** | **Descripción** | **Estado** |
| RF-01 | Registro de nuevos usuarios | Funcional | Usuario | Permite a nuevos usuarios crear una cuenta para acceder al sistema. | Pendiente |
| RF-02 | Inicio de sesión seguro | Funcional | Usuario | Los usuarios pueden acceder al sistema con credenciales encriptadas. | Pendiente |
| RF-03 | Recuperación de contraseña | Funcional | Usuario | Facilita a los usuarios recuperar o cambiar su contraseña olvidada. | Pendiente |
| RF-04 | Creación de empresas clientes | Funcional | Usuario | Permite registrar a las empresas como clientes con su información relevante. | Pendiente |
| RF-05 | Gestión de empresas | Funcional | Usuario | Permite editar o eliminar información de las empresas registradas. | Pendiente |
| RF-06 | Generación de órdenes | Funcional | Usuario | Los usuarios pueden crear automáticamente órdenes de transporte. | Pendiente |
| RF-07 | Seguimiento de órdenes | Funcional | Usuario | Permite consultar el estado y detalles de las órdenes de transporte. | Pendiente |
| RF-08 | Notificaciones | Funcional | Usuario | Envía notificaciones al usuario cuando se crea o actualiza una orden. | Pendiente |
| RF-09 | Panel de control | Funcional | Usuario, Administrador | Proporciona un resumen de órdenes y estadísticas relacionadas. | Pendiente |
| RNF-01 | Encriptación de contraseñas | No Funcional | Sistema | Las contraseñas deben ser encriptadas para garantizar la seguridad. | Pendiente |
| RNF-02 | Comunicación segura (HTTPS) | No Funcional | Sistema | Todo el tráfico de datos debe utilizar HTTPS para proteger la información. | Pendiente |
| RNF-03 | Respuesta rápida | No Funcional | Sistema | El sistema debe responder en ≤ 2 segundos para consultas y acciones comunes. | Pendiente |
| RNF-04 | Escalabilidad | No Funcional | Sistema | Debe soportar un aumento en usuarios y órdenes sin pérdida de rendimiento. | Pendiente |
| RNF-05 | Alta disponibilidad | No Funcional | Sistema | Garantizar un tiempo de actividad del sistema del 99.9%. | Pendiente |
| RNF-06 | Compatibilidad | No Funcional | Sistema | Accesible desde navegadores modernos y dispositivos móviles o de escritorio. | Pendiente |
| RNF-07 | Almacenamiento eficiente | No Funcional | Sistema | Optimización del uso de la base de datos Azure SQL para un desempeño óptimo. | Pendiente |
| RNF-08 | Backup automático | No Funcional | Sistema, Administrador | Realizar copias de seguridad diarias para proteger los datos. | Pendiente |
| RNF-09 | Recuperación ante fallos | No Funcional | Sistema, Administrador | Permite restaurar datos en caso de fallos críticos o pérdida de información. | Pendiente |

Nota: tabla muestra las tareas funcionales y no funcionales levantadas en el proceso de análisis de requerimientos (fuente: elaboración propia, 2024)

Flujos de Procesos.

Los flujos de procesos nos ayudan a identificar y documentar las actividades pasos y decisiones necesarias para llevar a cabo un proceso dentro de un sistema. En base a los requerimientos levantados podemos establecer los siguientes flujos para el nuevo sistema.

Registro de usuarios/empresas.

Flujo: Inicio del usuario → Completa formulario → Validación de datos → Creación de cuenta → Notificación de éxito.

Ilustración - Flujo registro usuario/empresa

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Generación de órdenes de transporte.

Flujo: Usuario inicia sesión → Selecciona cliente → Completa detalles de la orden (origen, destino, carga) → Sistema autogenera número de orden → Guardar orden → Notificación al cliente

Ilustración - Flujo generación de transporte

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Seguimiento de Orden

Flujo: Cliente inicia sesión → Introduce número de orden → Consulta estado (en tránsito, entregado) → Visualiza detalles.

Ilustración - Seguimiento de orden

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Casos de uso.

Un caso de uso es una descripción de cómo un usuario interactúa con un sistema para lograr un objetivo específico. Detalla los pasos principales, actores involucrados y las condiciones necesarias para completar la interacción. Ayuda a identificar requisitos funcionales y diseñar procesos claros.

A continuación, se enlistan los casos de uso, de los análisis de la solución a implementar.

Tabla - Caso de uso - CU-01

|  |  |
| --- | --- |
| **Caso N°** | CU-01 |
| **Nombre** | Registro de Usuario |
| **Actor** | Usuario |
| **Precondición** | El usuario no debe estar registrado previamente. |
| **Postcondición** | El usuario queda registrado y puede iniciar sesión. |
| **Flujo principal** | |
| 1- El usuario accede a la página de registro. 2- Completa el formulario con nombre, correo electrónico, contraseña, y otros datos necesarios. 3- El sistema valida los datos ingresados. 4- Se guarda la información en la base de datos con la contraseña encriptada. 5- El sistema confirma el registro exitoso con un mensaje o correo electrónico. | |
|  |
|  |
|  |
|  |
|  |
|  |

Ilustración - Caso de uso -Registro de Usuario

Diagrama

Descripción generada automáticamente

Tabla - Caso de uso - CU-02

|  |  |
| --- | --- |
| **Caso N°** | CU-02 |
| **Nombre** | Inicio de Sesión |
| **Actor** | Usuario |
| **Precondición** | El usuario debe estar registrado previamente. |
| **Postcondición** | El usuario tiene acceso a las funcionalidades del sistema |
| **Flujo principal** | |
| 1- El usuario accede a la página de inicio de sesión. 2- Ingresa su correo electrónico y contraseña. 3- El sistema verifica las credenciales. 4- Si son válidas, el usuario accede al sistema. | |
|  |
|  |
|  |
|  |
| **Flujo alternativo** | |  |
| 1-Si las credenciales son incorrectas, el sistema muestra un mensaje de error | |  |
|  |
|  |

Ilustración - Caso de uso - Inicio de Sesión

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Tabla - Caso de uso - CU-03

|  |  |
| --- | --- |
| **Caso N°** | CU-03 |
| **Nombre** | Recuperación de Contraseña |
| **Actor** | Usuario |
| **Precondición** | El usuario debe estar registrado y tener acceso al correo electrónico proporcionado. |
| **Postcondición** | El usuario puede iniciar sesión con la nueva contraseña. |
| **Flujo principal** | |
| 1- El usuario solicita la recuperación desde la página de inicio de sesión. 2- Ingresa su correo electrónico. 3- El sistema envía un enlace para restablecer la contraseña. 4- El usuario accede al enlace y establece una nueva contraseña. 5- El sistema actualiza la base de datos con la nueva contraseña encriptada. | |
|  |
|  |
|  |
|  |
|  |

Ilustración - Caso de uso - Recuperación de Contraseña

Diagrama

Descripción generada automáticamente

Tabla - Caso de uso - CU-04

|  |  |
| --- | --- |
| **Caso N°** | CU-04 |
| **Nombre** | Creación de Empresa |
| **Actor** | Usuario |
| **Precondición** | El usuario debe estar autenticado. |
| **Postcondición** | La empresa queda registrada y disponible para asociarla con órdenes. |
| **Flujo principal** | |
| 1- El usuario accede a la sección "Registrar Empresa". 2- Completa el formulario con datos de la empresa (nombre, dirección, contacto). 3- El sistema valida los datos ingresados. 4- Se guarda la información de la empresa en la base de datos. 5- El sistema confirma la creación exitosa. | |
|  |
|  |
|  |
|  |
|  |

Ilustración - Caso de uso - Creación de empresa

Diagrama

Descripción generada automáticamente con confianza media

Tabla - Caso de uso - CU-05

|  |  |
| --- | --- |
| **Caso N°** | CU-05 |
| **Nombre** | Edición de Datos de Empresa |
| **Actor** | Usuario |
| **Precondición** | La empresa debe estar registrada previamente. El usuario debe tener permisos para editar. |
| **Postcondición** | Los datos de la empresa se actualizan correctamente. |
| **Flujo principal** | |
| 1- El usuario selecciona una empresa existente. 2- Realiza las modificaciones necesarias en el formulario. 3- El sistema valida los cambios. 4- Los datos actualizados se guardan en la base de datos. 5- El sistema confirma los cambios. | |
|  |
|  |
|  |
|  |
|  |

Ilustración - Caso de uso - Edición de Datos de Empresa

Imagen que contiene animal

Descripción generada automáticamente

Tabla - Caso de uso - CU-06

|  |  |
| --- | --- |
|  |  |
| **Caso N°** | CU-06 |
| **Nombre** | Generación de Orden de Transporte |
| **Actor** | Usuario |
| **Precondición** | El usuario debe estar autenticado. La empresa cliente debe estar registrada. |
| **Postcondición** | La orden queda registrada y disponible para seguimiento. |
| **Flujo principal** | |
| 1- El usuario accede a la sección "Nueva Orden". 2- Selecciona una empresa cliente. 3- Completa detalles de la orden (origen, destino, tipo de carga, fecha). 4- El sistema valida los datos ingresados. 5- Genera un número único de orden. 6- Guarda la orden en la base de datos. 7- El sistema confirma la creación y envía una notificación. | |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

Ilustración - Caso de uso - Generación de Orden de Transporte

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Tabla - Caso de uso - CU-07

|  |  |
| --- | --- |
| **Caso N°** | CU-07 |
| **Nombre** | Seguimiento de Orden |
| **Actor** | Usuario |
| **Precondición** | La orden debe estar registrada en el sistema. |
| **Postcondición** | El usuario obtiene la información actualizada de la orden. |
| **Flujo principal** | |
| 1- El usuario accede a la sección "Seguimiento". 2- Ingresa el número de la orden o selecciona una de la lista. 3- El sistema muestra el estado actual 4- El usuario consulta los detalles. | |
|  |
|  |
|  |
|  |

Ilustración - Caso de uso - Seguimiento de Orden

Diagrama

Descripción generada automáticamente

Tabla - Caso de uso - CU-08

|  |  |
| --- | --- |
| **Caso N°** | CU-08 |
| **Nombre** | Notificaciones |
| **Actor** | Sistema |
| **Precondición** | Debe ocurrir un evento relevante |
| **Postcondición** | Los usuarios reciben información oportuna sobre las órdenes. |
| **Flujo principal** | |
| 1- El sistema detecta el evento (ejemplo: creación de una orden). 2- Genera un mensaje con los detalles del evento. 3- Envía la notificación al correo electrónico del usuario o cliente | |
|  |
|  |
|  |

Ilustración - Caso de uso - Notificaciones

Imagen que contiene Diagrama

Descripción generada automáticamente

Tabla - Caso de uso - CU-09

|  |  |
| --- | --- |
| **Caso N°** | CU-09 |
| **Nombre** | Gestión de Órdenes |
| **Actor** | Usuario, Administrador |
| **Precondición** | Las órdenes deben estar registradas. |
| **Postcondición** | La orden se actualiza según las acciones realizadas. |
| **Flujo principal** | |
| 1- El usuario o administrador accede a la sección "Órdenes". 2- Visualiza una lista de órdenes activas, finalizadas o en proceso. 3- Puede buscar órdenes específicas por número, cliente o estado. 4- Realiza acciones como actualizar o cancelar una orden. | |
|  |
|  |
|  |

Ilustración - Caso de uso - Gestión de Órdenes

Diagrama

Descripción generada automáticamente con confianza media

Tabla - Caso de uso - CU-10

|  |  |
| --- | --- |
| **Caso N°** | CU-10 |
| **Nombre** | Generación de Reportes |
| **Actor** | Administrador |
| **Precondición** | El administrador debe tener permisos de acceso. |
| **Postcondición** | El administrador obtiene información consolidada para la toma de decisiones. |
| **Flujo principal** | |
| 1- El administrador accede a la sección "Reportes". 2- Selecciona los parámetros del reporte (fechas, cliente, estado). 3- El sistema genera un reporte con los datos seleccionados. 4- El reporte se presenta en pantalla | |
|  |
|  |
|  |

Ilustración - Caso de uso - Generación de reportes

Imagen que contiene Diagrama

Descripción generada automáticamente

Modelo 3 capas.

Para este proyecto la arquitectura utilizada es el modelo de 3 capas. Esta arquitectura divide un diseño de software en 3 capas separadas lo que permite modularidad, escalabilidad y mantenibilidad. Es un modelo muy usado para el desarrollo de aplicaciones en entorno web.

**Capa de presentación:** Es la que está del lado del usuario y se encarga de la interacción entre el usuario y el sistema, utiliza elementos visuales como formularios, páginas web, botones y otros, y para este proyecto utiliza lenguaje HTML, CSS y JavaScript para su desarrollo.

**Capa de negocio:** Es la que contiene la lógica central de la aplicación, se encarga del procesamiento de los datos y los cálculos necesarios para el funcionamiento de la aplicación, también actúa de puente entre la capa de presentación y la capa de datos. Para este proyecto utiliza Node.JS y JavaScript como base para la ejecución de los procesos que necesita el aplicativo.

**Capa de datos:** Administra el acceso y la gestión de los datos, actúa con la base de datos u otros servicios para almacenar la información. Para este proyecto se utiliza Azure SQL para el almacenado y respaldo de los datos de la aplicación.

Ilustración - Modelo de 3 capas

Imagen que contiene Texto

Descripción generada automáticamenteNota: Modelo de 3 capas, Presentación, Negocios Datos. (fuente: elaboración propia, 2024)

Arquitectura física.

Este proyecto adoptará una arquitectura basada en la nube para mejorar la escalabilidad, optimizar los costos operativos, garantizar la alta disponibilidad del sistema y facilitar el acceso remoto seguro desde cualquier ubicación. Se elegirá Microsoft Azure como proveedor de servicios en la nube debido a su robusta gama de servicios integrados, alta confiabilidad, características de seguridad avanzadas y su capacidad para soportar aplicaciones empresariales. Entre las herramientas clave de Azure, se utilizarán Azure SQL para la gestión de bases de datos y Azure App Services para el alojamiento de aplicaciones, lo que asegurará un entorno seguro, eficiente y adaptable a las necesidades del proyecto.

Servicios utilizados en la nube.

**Azure App Services:** es una plataforma de Microsoft para crear y alojar aplicaciones web y APIs. Permite escalar automáticamente y ofrece seguridad integrada. Soporta varios lenguajes como .NET, Java, y Node.js. Facilita el desarrollo sin la necesidad de tener infraestructura física.

**Azure SQL:** es un servicio de base de datos en la nube que ofrece una versión administrada de SQL Server. Permite almacenar y gestionar datos de manera escalable, segura y con alta disponibilidad. Incluye características de copias de seguridad automáticas, recuperación ante desastres y escalabilidad según la necesidad. Suele usarse para aplicaciones que requieren bases de datos relacionales.

Ilustración - Infraestructura en Microsoft Azure

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Nota: Diagrama de infraestructura de los servicios implementados en Microsoft Azure (fuente: elaboración propia, 2024)

Product Backlog.

Con el objetivo de organizar mejor el proyecto y permitir iteraciones para alcanzar los objetivos deseados, se definió un orden para las tareas identificadas como necesarias para el desarrollo de la plataforma. Tal como se hace en el marco de trabajo Scrum, en el cual se establece el product backlog y se priorizan ciertas tareas, pero ahora dentro del contexto de la metodología incremental.

El Product Backlog es una lista priorizada y dinámica de todos los requisitos, tareas o características necesarias para desarrollar un producto en proyectos que utiliza la metodología ágil Scrum.

El product Backlog ayuda a:

* **Gestionar el alcance del Proyecto:** Define que debe incluirse en el producto final.
* **Establecer prioridades:** Ayuda al equipo y al cliente a enfocarse en los elementos de mayor valor
* **Planificar el trabajo:** Durante las reuniones de planificación de sprint, el equipo selecciona elementos del backlog para desarrollar en el ciclo actual.
* **Facilitar la comunicación:** Asegura que todos los involucrados (desarrollares, Product Owner y Stakeholders) estén alineados en cuanto a las prioridades y objetivos.

Tabla - Requerimientos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Título** | **Descripción Extendida** | **Prioridad** | **Actor Principal** | **Criterio de Aceptación** |
| 1 | Registro de Usuario | Los usuarios pueden registrarse proporcionando datos como nombre, correo electrónico, contraseña, y teléfono. El sistema valida que los datos ingresados sean correctos y no estén duplicados. | Alta | Usuario | - Los datos se almacenan correctamente en la base de datos. - El correo electrónico ingresado es único. - Se envía un correo de bienvenida al usuario registrado. |
| 2 | Inicio de Sesión | Los usuarios pueden acceder al sistema ingresando su correo electrónico y contraseña. El sistema valida las credenciales, generando un token JWT en caso de éxito. | Alta | Usuario | - El usuario accede al sistema con credenciales válidas. - El token JWT tiene una vigencia configurada. - Las credenciales incorrectas generan un mensaje claro |
| 3 | Recuperación de Contraseña | Los usuarios pueden solicitar un restablecimiento de contraseña. El sistema envía un enlace al correo electrónico registrado para permitir cambiar la contraseña. | Media | Usuario | - El enlace enviado tiene un token válido y caduca después de cierto tiempo. - El usuario puede establecer una nueva contraseña encriptada. |
| 4 | Registro de Empresas | Los usuarios pueden registrar información de empresas clientes, como nombre, dirección, teléfono, y RUT (ID fiscal). El sistema verifica que los datos no estén duplicados. | Alta | Usuario | - La empresa es creada con datos válidos. - La información es accesible desde un listado. - No se permite registrar una empresa con datos repetidos. |
| 5 | Edición de Empresas | Los usuarios pueden actualizar la información de las empresas previamente registradas. Cambios como dirección, teléfono o nombre son posibles. | Media | Usuario | - Los cambios se reflejan en la base de datos. - El sistema valida que el nuevo nombre o RUT no existan ya registrados. |
| 6 | Generación de Órdenes de Transporte | Permitir a los usuarios crear órdenes de transporte. Ingresan datos como origen, destino, tipo de carga, fecha, y cliente. El sistema genera automáticamente un número único para la orden. | Alta | Usuario | - La orden se guarda en la base de datos con un ID único. - El sistema notifica al cliente registrado. - Los campos requeridos son validados antes de la creación. |
| 7 | Seguimiento de Órdenes | Los usuarios pueden visualizar el estado actual de sus órdenes de transporte, que incluyen detalles como ubicación, estado (en tránsito, entregado) y fecha estimada de entrega. | Alta | Usuario | - El usuario ve un estado actualizado. - Cambios en el estado generan notificaciones al cliente (opcional). - La interfaz es clara y muestra un historial del pedido. |
| 8 | Notificaciones por Correo | El sistema envía correos electrónicos automáticos para confirmaciones de registro, creación de órdenes, actualizaciones de estado, y otras interacciones clave. | Media | Sistema | - Los correos son enviados en los momentos esperados. - Se incluye un formato claro y profesional. - Correos no válidos generan un registro de error. |
| 9 | Gestión de Usuarios | Los administradores pueden gestionar usuarios del sistema. Esto incluye creación, eliminación, actualización de datos, y asignación de roles (admin, usuario). | Baja | Administrador | - Los administradores pueden listar todos los usuarios existentes. - Los cambios en los roles se reflejan de inmediato. - No se permite borrar usuarios activos. |
| 10 | Generación de Reportes | Los administradores pueden generar reportes detallados sobre las órdenes de transporte, usuarios registrados, y clientes. Los reportes son exportables a formatos como PDF y Excel. | Media | Administrador | - El reporte incluye filtros como rango de fechas y estados. - Los datos exportados coinciden con los visibles en la interfaz. |
| 11 | Diseño de Interfaz de Usuario | La aplicación debe contar con una interfaz visual amigable y responsiva, diseñada para adaptarse a dispositivos móviles, tabletas, y pantallas de escritorio. | Alta | Usuario | - La interfaz funciona correctamente en diferentes dispositivos y navegadores. - Los formularios son fáciles de completar y tienen mensajes claros para errores. |
| 12 | Seguridad de Contraseñas | Todas las contraseñas se almacenan en formato encriptado en la base de datos usando un algoritmo seguro como bcrypt. | Alta | Sistema | - Las contraseñas en texto plano no son visibles ni recuperables. - El sistema previene ataques como fuerza bruta y solicita contraseñas fuertes. |
| 13 | Integración con Azure SQL | Configuración de la base de datos en Azure SQL para almacenar la información del sistema (usuarios, empresas, órdenes, etc.), garantizando integridad y rendimiento. | Alta | Sistema | - La base de datos se conecta exitosamente con la capa de negocio. - Las consultas y escrituras son rápidas y confiables. |
| 14 | Hosting en Azure | Configuración de la infraestructura en Azure para hospedar la aplicación web. Esto incluye el backend en Azure App Services y la base de datos en Azure SQL Database. | Alta | Sistema | - La aplicación es accesible mediante una URL pública. - La infraestructura es escalable y permite monitoreo mediante Azure Monitor. |
| 15 | Pruebas Funcionales | Realización de pruebas para garantizar que todas las funcionalidades del sistema funcionan correctamente según los requisitos definidos. | Alta | Equipo de Desarrollo | - Todas las funcionalidades críticas son probadas y validadas. - Se documentan los errores encontrados y las soluciones implementadas. |
| 16 | Documentación del Sistema | Creación de manuales de usuario y documentación técnica para describir el uso y mantenimiento del sistema. | Baja | Equipo de Desarrollo | - La documentación está actualizada y cubre todos los aspectos del sistema. - Los usuarios pueden seguir el manual para realizar tareas básicas. |

Desarrollo de tareas.

Primer entregable.

El desarrollo del primer entregable involucra el desarrollo de las siguientes tareas en un periodo de 20 días.

Tabla - Tareas primer entregable

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Título** | **Descripción** | **Actor Principal** | **Estimación (Puntos)** | **Criterio de Aceptación** |
| 1 | Registro de Usuario | Permitir a los usuarios registrarse en el sistema. | Usuario | 3 | Usuarios pueden registrarse y recibir un correo de bienvenida. |
| 2 | Inicio de Sesión | Implementar login con validación de credenciales y generación de token JWT. | Usuario | 5 | Usuarios pueden iniciar sesión con credenciales válidas. |
| 3 | Recuperación de Contraseña | Habilitar la opción de restablecer contraseñas. | Usuario | 3 | El enlace para restablecer contraseñas funciona correctamente. |
| 4 | Seguridad de Contraseñas | Encriptar contraseñas usando bcrypt. | Sistema | 5 | Las contraseñas están almacenadas de forma segura en la base de datos. |
| 13 | Integración con Azure SQL | Configurar la base de datos en Azure SQL para gestionar usuarios y empresas. | Sistema | 8 | La base de datos almacena correctamente los datos. |
| 14 | Hosting en Azure | Configurar la infraestructura en Azure para el hospedaje del sistema. | Sistema | 8 | La aplicación es accesible a través de una URL pública. |

Segundo entregable.

El desarrollo del segundo entregable involucra el desarrollo de las siguientes tareas en un periodo de 55 días.

Tabla - Tareas segundo entregable

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Título** | **Descripción** | **Actor Principal** | **Estimación (Puntos)** | **Criterio de Aceptación** |
| 4 | Registro de Empresas | Permitir a los usuarios registrar empresas clientes. | Usuario | 5 | Los datos de las empresas son almacenados correctamente. |
| 5 | Edición de Empresas | Habilitar la edición de la información de empresas. | Usuario | 3 | Los datos actualizados se reflejan en el sistema. |
| 6 | Generación de Órdenes de Transporte | Crear órdenes con detalles como origen, destino, tipo de carga y fecha. | Usuario | 8 | Las órdenes se generan con un ID único y se guardan correctamente. |
| 8 | Notificaciones por Correo | Enviar notificaciones por correo al crear una orden. | Sistema | 3 | Los correos son enviados al cliente cuando se crea una orden. |
| 11 | Diseño de Interfaz de Usuario | Mejorar la interfaz de usuario para las secciones de empresas y órdenes. | Usuario | 13 | La interfaz es clara, funcional y responsiva. |

Tercer entregable.

El desarrollo del tercer entregable involucra el desarrollo de las siguientes tareas en un periodo de 15 días

Tabla - Tareas tercer entregable

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Título** | **Descripción** | **Actor Principal** | **Estimación (Puntos)** | **Criterio de Aceptación** |
| 7 | Seguimiento de Órdenes | Permitir a los usuarios consultar el estado de sus órdenes en tiempo real. | Usuario | 8 | El usuario puede ver estados actualizados y detalles históricos de sus órdenes. |
| 10 | Generación de Reportes | Generar reportes detallados con datos de órdenes y empresas. | Administrador | 8 | Los reportes son exportables en PDF y Excel y muestran datos correctos. |
| 9 | Gestión de Usuarios | Habilitar la gestión de usuarios (roles, permisos). | Administrador | 5 | Los administradores pueden asignar roles a los usuarios existentes. |
| 15 | Pruebas Funcionales | Realizar pruebas para garantizar el correcto funcionamiento del sistema. | Equipo de Desarrollo | 13 | Todas las funcionalidades principales son probadas y validadas. |
| 16 | Documentación del Sistema | Crear manuales para usuarios y documentación técnica para mantenimiento del sistema. | Equipo de Desarrollo | 5 | La documentación está actualizada y es fácil de entender. |

Plataforma de aplicación.

Azure Web App Services, proporciona la plataforma para ejecutar nuestro proyecto. En ella se alojará el backend, donde se gestionan las funcionalidades del proyecto, y el frontend, que interactúa directamente con el usuario.

Ilustración - implementación de Web APP Services

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Base de datos.

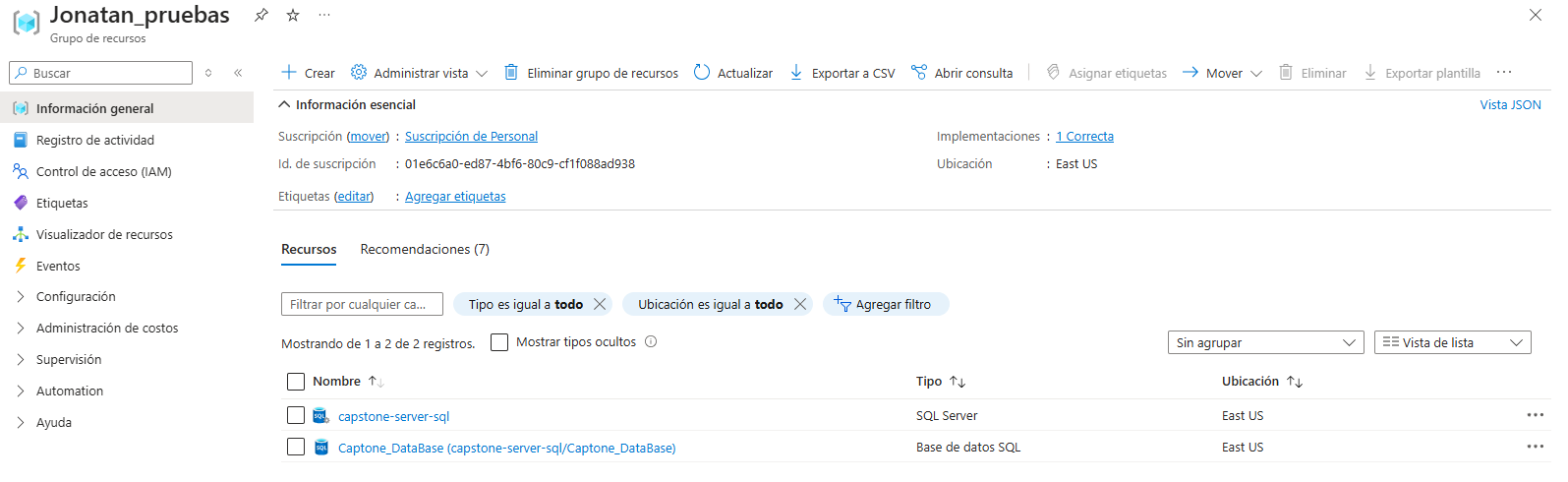
La base de datos es un pilar fundamental en cualquier proyecto de software, ya que permite almacenar, organizar y gestionar la información requerida para el correcto funcionamiento del sistema. Además, asegura que los datos estén disponibles de manera confiable, eficiente y segura, facilitando las operaciones y procesos críticos del aplicativo.

Microsoft Azure data base.

Para este proyecto, se utiliza Microsoft Azure Database como la solución principal para almacenar los datos necesarios, garantizando que la información se gestione y guarde de manera eficiente.

Como primer paso, es necesario crear un grupo de recursos en Azure, que servirá como contenedor para todos los servicios requeridos por la aplicación. En este caso, se configurarán los servicios de Azure Database necesarios para soportar las funcionalidades del sistema.

Ilustración - Grupo de recursos de Azure



Modelo lógico.

El modelo lógico es un diseño fundamental que describe cómo se estructuran y relacionan los datos dentro de un sistema. Emplea conceptos como entidades, relaciones y atributos, siendo independiente de la tecnología. Actúa como un puente entre el diseño conceptual y su implementación física, proporcionando una visión clara y general sobre cómo interactúan los datos entre sí.

Ilustración - Modelo lógico de datos

Diagrama

Descripción generada automáticamente

Modelo relacional.

El modelo relacional es un enfoque técnico que organiza los datos en tablas, filas y columnas. Cada fila representa un registro único, y las tablas incluyen claves primarias, que identifican de manera exclusiva cada registro, y claves foráneas, que vinculan las tablas entre sí. Este modelo permite definir el tipo de dato de cada columna (texto, número, decimal, etc.). Además, se utiliza para generar código SQL, el cual se ejecuta en el motor de base de datos para crear las tablas con sus tipos de datos específicos y establecer sus relaciones.

Ilustración - Modelo relacional de datos

Diagrama

Descripción generada automáticamente

Código SQL.

El código SQL DDL se utiliza para crear y configurar en el motor de base de datos todas las estructuras necesarias, como tablas, relaciones y otros objetos, requeridos para el proyecto. Una vez ejecutado, la base de datos queda lista para integrarse al proyecto y almacenar la información necesaria para el funcionamiento del aplicativo que se implementará.

Ilustración - Generación de tablas mediante DDL

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Detalles de código en anexo “[Código DDL](#index_código_DDL)”

A continuación, podemos ver la estructura de la base de datos mediante la herramienta de SQL Management Studio.

Ilustración - Estructura de tablas SQL Managent Studio

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Desarrollo BackEnd.

El entorno de trabajo utilizado para el desarrollo de este proyecto es Node JS. Utiliza JavaScript para crear aplicaciones del lado del servidor. Es rápido, eficiente y excelente para manejar muchas conexiones al mismo tiempo, como chats o servicios en tiempo real.

Junto con Node.JS es necesario poder utilizar un framework como es el caso de express el cual nos permite desarrollar entornos de aplicaciones de manera robusta y bibliotecas complementarias que ayudaran a ejecutar los distintos componentes para el desarrollo de la aplicación, como es el caso de mssql el cual establece la conexión entre la base de datos y el proyecto.

A continuación, se puede apreciar el desarrollo en donde están los parámetros principales de la aplicación.

Ilustración - FrameWork Express

Texto

Descripción generada automáticamente

FrontEnd.

El FrontEnd es crucial para el proyecto, es la parte de la aplicación que interactúa directamente con el usuario. Es lo que el usuario puede apreciar visualmente con un navegador web y contiene el diseño visual con el que interactúa con la aplicación.

Para la estructura principal del FrontEnd se utiliza HTML se utiliza para los encabezados, párrafos, imágenes, formularios, etc.

Para complementar el estilo visual de la aplicación se utiliza CSS que es un lenguaje que se utiliza para el diseño web y controla los aspectos visuales como colores, márgenes y distribución de los elementos.

Las funcionalidades se otorgan con el lenguaje JavaScript, que añade interactividad y funcionalidad a la estructura de la página web, además de peticiones a los servidores para procesar la información.

HTML

Ilustración - FrontEnd en HTML

Texto

Descripción generada automáticamente

CSS

Ilustración - Estilo visual en CSS

Texto

Descripción generada automáticamente

Página web.

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente.

Ilustración 30 - Página web de inició de sesión

Factibilidad económica.

El modelo de negocio utilizado para este proyecto será en modo de arriendo/suscripción por el periodo mínimo de 2 años a Setralog. El flujo de caja es una herramienta esencial para evaluar la viabilidad financiera de cualquier proyecto.

En este caso, presentamos un análisis del flujo de caja asociado a nuestro proyecto, considerando ingresos, costos operativos, inversiones iniciales y proyecciones futuras. Este análisis permite determinar si los flujos netos generados son suficientes para cubrir las inversiones realizadas y garantizar la sostenibilidad del proyecto a lo largo del tiempo. El objetivo principal es establecer, con base en datos concretos, si el proyecto es financieramente viable y rentable.

Flujo de caja.

Ingresos esperados por concepto suscripción

Tabla

Descripción generada automáticamente

Inversión

|  |  |
| --- | --- |
| Detalle | |
| Miembros equipo | 3 |
| Horas trabajadas | 65 |
| Valor Hora UF | 2,5 |
| Total UF | 487,5 |
| Total CLP | $18.608.850 |

Tabla

Descripción generada automáticamente

Costos

Captura de pantalla de un celular

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Tabla

Descripción generada automáticamente

VAN valor actual neto.

El VAN del proyecto es de 29.200.583

TIR tasa interna de retorno.

El TIR del proyecto es de 30%

Conclusión.

El proyecto APT ofrece una solución integral basada en una plataforma web, diseñada para abordar las ineficiencias operativas de Setralog mediante la implementación de tecnologías modernas. Esta herramienta centraliza la gestión de clientes y órdenes de transporte, optimizando los procesos y reduciendo significativamente los errores y tiempos de respuesta.

La propuesta, basada en metodologías ágiles, no solo mejora la experiencia del cliente a través de una interfaz intuitiva, sino que también fortalece la competitividad de Setralog en el mercado actual. La adopción de tecnologías en la nube asegura escalabilidad, seguridad y flexibilidad, mientras que la automatización de procesos incrementa la precisión y eficiencia operativa.

El proyecto tiene un impacto positivo en la empresa, y permite a los integrantes del equipo aplicar y desarrollar las competencias técnicas, blandas y de gestión adquiridas en Ingeniería en Informática, consolidando así su preparación profesional.

Mejoras a futuro.

Debido a las limitaciones de tiempo en el desarrollo del proyecto, no fue posible incluir módulos adicionales que podrían agregar más valor. Sin embargo, a continuación, se presentan algunas ideas de mejora que podrían optimizar aún más el software desarrollado y maximizar los beneficios a largo plazo:

* Optimización de rutas
* Rastreo de los vehículos de carga con GPS mediante tecnología IOT.
* Cubicación automática de los camiones de carga.
* Conexión con SII para validación de datos de clientes.
* Dashboard interactivo con alguna herramienta BI para análisis de rendimiento.
* Módulo de contabilidad
* Facturación electrónica integrada.
* Módulo de inventario (en caso de que cliente final solicite el servicio de almacenaje y transporte)
* Recepción electrónica de factura de proveedores.
* Generación de órdenes de compra.
* Control de mantenciones de floja vehicular.

Acrónimos.

A

APT

Asignatura de portafolio de título 4, 5, 6, 7, 10, 11, 13, 18, 25, 60, 62

**Aws**

Amazon Web Services 62

**Azure**

Es la plataforma Microsoft que ofrece servicios en la nube 3, 11, 15, 27, 33, 44, 45, 48, 49, 52, 63

B

**Backend**

Son datos e infraestructura que permiten que una aplicación funcione 62

C

**Carta Gantt**

Herramienta grafica que demuestra el tiempo para cumplir con diversas tareas 2, 29, 62

**CSS**

Se utiliza para definir y dar estilo a las páginas web 15, 43, 58, 63

D

**Data Base**

Herramienta para recopilar y organizar información 63

**Development Team**

Grupo de personas encargadas de desarrollar un proyecto 19, 22, 24, 62

E

**Excel**

Programa de hoja de cálculos 6, 9, 10, 11, 48, 50, 62

F

**Framework**

Marco de trabajo que ofrece una estructura de base para elaborar un proyecto 63

G

**Google Cloud**

Plataforma Informática utilizada para almacenamiento de datos 11, 15, 62

H

**HTTP**

Protocolo de trasferencia de hipertexto 15, 63

I

**Iterativo**

Repetir varias veces un proceso para alcanzar la meta deseada 62

J

**JavaScript**

Lenguaje de programación que se utiliza para crear páginas web interactivas y dinámicas 15, 43, 56, 58, 62

K

**Kick-Off**

Reunión inicial de un proyecto 26, 62

M

**Management**

Gestión o Administración 55, 62

**Microsoft**

Corporación Tecnológica 3, 44, 45, 52, 63

N

**Node.Js**

Entorno en tiempo de ejecución multiplataforma, de código abierto 62

P

**Product Backlog**

Lista ordenada de tareas que se utilizan para desarrollar un producto 3, 27, 46, 62

**Product Owner**

Persona que toma decisiones sobre un proyecto 18, 22, 24, 27, 46, 62

**Pyme**

Pequeña y Mediana Empresa 62

**Python**

Lenguaje de programación de alto nivel 15, 62

R

**Rol**

Funciones que debe realizar una determinada persona en su entorno laboral 19, 24, 62

S

**Scrum Master**

Encargado de facilitar y agilizar proyectos 19, 21, 22, 24, 62

**Setralog**

Nombre de Cliente para proyecto APT 4, 6, 7, 9, 10, 11, 12, 13, 14, 18, 19, 20, 21, 24, 60, 62

**Software**

Programa Informático 62

**Sprint**

Ciclo de ejecución de un proyecto 3, 25, 27, 49, 50, 62

**Sql**

Lenguaje de Consulta Estructurado 62

**SSL**

Secure Sockets Layer 12, 63

**Stakeholder**

Persona o grupo que poseen un interés en un negocio 62

U

**UML**

Lenguaje Unificado de Modelo 26, 62

W

**Web**

Sistema de información que se comparte por Internet 15, 52, 62

Bibliografía.

Atlassian. (s. f.-b). Qué es scrum y cómo empezar. Recuperado el 18 de noviembre del 2024 de <https://www.atlassian.com/es/agile/scrum>

IBM Product Master 12.0.0. (s. f.). Definición de casos de uso. Recuperado el 19 de noviembre del 2024 de <https://www.ibm.com/docs/es/product-master/12.0.0?topic=processes-defining-use-cases>

Sentrio (2023, 21 junio). Diferencias entre el desarrollo iterativo e incremental. Recuperado 20 de noviembre de 2024, de <https://sentrio.io/blog/diferencias-entre-desarrollo-iterativo-e-incremental/>

Intelequia (s.f). ¿Qué modelo de ciclo de vida del software es el más adecuado para tu proyecto? Recuperado el 20 de noviembre del 2024 de <https://intelequia.com/es/blog/post/qu%C3%A9-modelo-de-ciclo-de-vida-del-software-es-el-m%C3%A1s-adecuado-para-tu-proyecto>

Anexos.

**Flujo de proceso original**

Diagrama

Descripción generada automáticamente

**Flujo de proceso optimizado**

Gráfico, Diagrama

Descripción generada automáticamente

**Código DDL**

|  |
| --- |
| CREATE TABLE ciudad (  cod\_cuidad INTEGER NOT NULL,  nombre\_ciudad VARCHAR (40) NOT NULL  );  ALTER TABLE ciudad ADD CONSTRAINT ciudad\_pk PRIMARY KEY ( cod\_cuidad );  CREATE TABLE cod\_despacho (  cod\_despacho INTEGER NOT NULL,  direccion VARCHAR (80) NOT NULL,  region\_cod\_region INTEGER NOT NULL,  ciudad\_cod\_cuidad INTEGER NOT NULL,  comunas\_cod\_comuna INTEGER NOT NULL  );  ALTER TABLE cod\_despacho ADD CONSTRAINT cod\_despacho\_pk PRIMARY KEY ( cod\_despacho );  CREATE TABLE comunas (  cod\_comuna INTEGER NOT NULL,  nombre\_comuna VARCHAR (40) NOT NULL  );  ALTER TABLE comunas ADD CONSTRAINT comunas\_pk PRIMARY KEY ( cod\_comuna );  CREATE TABLE detalle\_órdenes (  órdenes\_num\_orden INTEGER NOT NULL,  id\_detalle INTEGER NOT NULL,  producto VARCHAR (60),  cantidad INTEGER,  alto INTEGER NOT NULL,  ancho INTEGER NOT NULL,  largo INTEGER NOT NULL,  peso\_kgs FLOAT NOT NULL  );  ALTER TABLE detalle\_órdenes ADD CONSTRAINT detalle\_órdenes\_pk PRIMARY KEY ( órdenes\_num\_orden,  id\_detalle );  CREATE TABLE direccion\_retiro (  cod\_retiro INTEGER NOT NULL,  direccion VARCHAR (80) NOT NULL,  region\_cod\_region INTEGER NOT NULL,  ciudad\_cod\_cuidad INTEGER NOT NULL,  comunas\_cod\_comuna INTEGER NOT NULL  );  ALTER TABLE direccion\_retiro ADD CONSTRAINT direccion\_retiro\_pk PRIMARY KEY ( cod\_retiro );  CREATE TABLE empresas (  rut INTEGER NOT NULL,  digito\_verificador VARCHAR(1) NOT NULL,  nombre\_fantasia VARCHAR (80) NOT NULL,  razon\_social VARCHAR (80) NOT NULL,  giro VARCHAR (90),  direccion\_comercial VARCHAR (60) NOT NULL,  region\_cod\_region INTEGER NOT NULL,  ciudad\_cod\_cuidad INTEGER NOT NULL,  comunas\_cod\_comuna INTEGER NOT NULL,  giro\_id\_giro INTEGER NOT NULL  );  ALTER TABLE empresas ADD CONSTRAINT empresas\_pk PRIMARY KEY ( rut );  CREATE TABLE estado\_órdenes (  cod\_estado VARCHAR (1) NOT NULL,  nombre\_estado VARCHAR (15)  );  ALTER TABLE estado\_órdenes ADD CONSTRAINT estado\_órdenes\_pk PRIMARY KEY ( cod\_estado );  CREATE TABLE giro (  id\_giro INTEGER NOT NULL,  descripcion\_giro VARCHAR (160) NOT NULL  );  ALTER TABLE giro ADD CONSTRAINT giro\_pk PRIMARY KEY ( id\_giro );  CREATE TABLE órdenes (  num\_orden INTEGER NOT NULL,  detalle\_orden VARCHAR (256),  direccion\_retiro VARCHAR (256) NOT NULL,  direccion\_despacho VARCHAR (256) NOT NULL,  empresas\_rut INTEGER NOT NULL,  usuarios\_id\_usuario INTEGER NOT NULL,  estado\_órdenes\_cod\_estado VARCHAR(1) NOT NULL,  direccion\_retiro\_cod\_retiro INTEGER NOT NULL,  cod\_despacho\_cod\_retiro INTEGER NOT NULL,  direccion\_retiro\_cod\_retiro1 INTEGER NOT NULL,  cod\_despacho\_cod\_despacho INTEGER NOT NULL,  correo\_electronico VARCHAR(60) NOT NULL  );  ALTER TABLE órdenes ADD CONSTRAINT órdenes\_pk PRIMARY KEY ( num\_orden );  CREATE TABLE region (  cod\_region INTEGER NOT NULL,  nombre\_region VARCHAR(30) NOT NULL  );  ALTER TABLE region ADD CONSTRAINT region\_pk PRIMARY KEY ( cod\_region );  CREATE TABLE usuario\_empresa (  usuarios\_correo\_electronico VARCHAR(60) NOT NULL,  empresas\_rut INTEGER NOT NULL  );  CREATE UNIQUE INDEX usuario\_empresa\_\_idx ON  usuario\_empresa (  usuarios\_correo\_electronico  ASC);  CREATE TABLE usuarios (  id\_usuario INTEGER NOT NULL,  correo\_electronico VARCHAR (60) NOT NULL,  password VARCHAR (15) NOT NULL,  nombre VARCHAR (60) NOT NULL,  tipo\_usuario VARCHAR (5)  );  ALTER TABLE usuarios ADD CONSTRAINT usuarios\_pk PRIMARY KEY ( correo\_electronico );  ALTER TABLE cod\_despacho  ADD CONSTRAINT cod\_despacho\_ciudad\_fk FOREIGN KEY ( ciudad\_cod\_cuidad )  REFERENCES ciudad ( cod\_cuidad );  ALTER TABLE cod\_despacho  ADD CONSTRAINT cod\_despacho\_comunas\_fk FOREIGN KEY ( comunas\_cod\_comuna )  REFERENCES comunas ( cod\_comuna );  ALTER TABLE cod\_despacho  ADD CONSTRAINT cod\_despacho\_region\_fk FOREIGN KEY ( region\_cod\_region )  REFERENCES region ( cod\_region );  ALTER TABLE detalle\_órdenes  ADD CONSTRAINT detalle\_órdenes\_órdenes\_fk FOREIGN KEY ( órdenes\_num\_orden )  REFERENCES órdenes ( num\_orden );  ALTER TABLE direccion\_retiro  ADD CONSTRAINT direccion\_retiro\_ciudad\_fk FOREIGN KEY ( ciudad\_cod\_cuidad )  REFERENCES ciudad ( cod\_cuidad );  ALTER TABLE direccion\_retiro  ADD CONSTRAINT direccion\_retiro\_comunas\_fk FOREIGN KEY ( comunas\_cod\_comuna )  REFERENCES comunas ( cod\_comuna );  ALTER TABLE direccion\_retiro  ADD CONSTRAINT direccion\_retiro\_region\_fk FOREIGN KEY ( region\_cod\_region )  REFERENCES region ( cod\_region );  ALTER TABLE empresas  ADD CONSTRAINT empresas\_ciudad\_fk FOREIGN KEY ( ciudad\_cod\_cuidad )  REFERENCES ciudad ( cod\_cuidad );  ALTER TABLE empresas  ADD CONSTRAINT empresas\_comunas\_fk FOREIGN KEY ( comunas\_cod\_comuna )  REFERENCES comunas ( cod\_comuna );  ALTER TABLE empresas  ADD CONSTRAINT empresas\_giro\_fk FOREIGN KEY ( giro\_id\_giro )  REFERENCES giro ( id\_giro );  ALTER TABLE empresas  ADD CONSTRAINT empresas\_region\_fk FOREIGN KEY ( region\_cod\_region )  REFERENCES region ( cod\_region );  ALTER TABLE órdenes  ADD CONSTRAINT órdenes\_cod\_despacho\_fk FOREIGN KEY ( cod\_despacho\_cod\_retiro )  REFERENCES cod\_despacho ( cod\_despacho );  ALTER TABLE órdenes  ADD CONSTRAINT órdenes\_cod\_despacho\_fkv1 FOREIGN KEY ( cod\_despacho\_cod\_despacho )  REFERENCES cod\_despacho ( cod\_despacho );  ALTER TABLE órdenes  ADD CONSTRAINT órdenes\_direccion\_retiro\_fk FOREIGN KEY ( direccion\_retiro\_cod\_retiro )  REFERENCES direccion\_retiro ( cod\_retiro );  ALTER TABLE órdenes  ADD CONSTRAINT órdenes\_direccion\_retiro\_fkv1 FOREIGN KEY (direccion\_retiro\_cod\_retiro1)  REFERENCES direccion\_retiro ( cod\_retiro );  ALTER TABLE órdenes  ADD CONSTRAINT órdenes\_empresas\_fk FOREIGN KEY ( empresas\_rut )  REFERENCES empresas ( rut );  ALTER TABLE órdenes  ADD CONSTRAINT órdenes\_estado\_órdenes\_fk FOREIGN KEY ( estado\_órdenes\_cod\_estado )  REFERENCES estado\_órdenes ( cod\_estado );  ALTER TABLE órdenes  ADD CONSTRAINT órdenes\_usuarios\_fk FOREIGN KEY ( correo\_electronico )  REFERENCES usuarios ( correo\_electronico );  ALTER TABLE usuario\_empresa  ADD CONSTRAINT usuario\_empresa\_empresas\_fk FOREIGN KEY ( empresas\_rut )  REFERENCES empresas ( rut );  ALTER TABLE usuario\_empresa  ADD CONSTRAINT usuario\_empresa\_usuarios\_fk FOREIGN KEY ( usuarios\_correo\_electronico )  REFERENCES usuarios ( correo\_electronico ); |