**Plataforma Integral de gestión de clientes y ordenes de transporte.**

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

Índice

Contenido

[Resumen ejecutivo (español) 3](#_Toc182951980)

[Introducción 4](#_Toc182951981)

[Descripción del problema 6](#_Toc182951982)

[Problema por resolver 7](#_Toc182951983)

[Solución al problema. 8](#_Toc182951984)

[Justificación 9](#_Toc182951985)

[Objetivo general 10](#_Toc182951986)

[Objetivos específicos 10](#_Toc182951987)

[Competencias Técnicas. 11](#_Toc182951988)

[Competencias Blandas. 11](#_Toc182951989)

[Competencias de Gestión. 12](#_Toc182951990)

[Acta de constitución de proyecto. 12](#_Toc182951991)

[Asignación de roles 12](#_Toc182951992)

[Metodología 12](#_Toc182951993)

[Fase 1 - Levantamiento de Requerimiento y Entendimiento del Negocio 13](#_Toc182951994)

[Fase 2 - Diseño, desarrollo y pruebas. 13](#_Toc182951995)

[Fase 3 – Implementación y marcha blanca. 14](#_Toc182951996)

[Carta Gantt 15](#_Toc182951997)

[Implementación del proyecto 16](#_Toc182951998)

[Base de datos 16](#_Toc182951999)

[Microsoft Azure data base. 16](#_Toc182952000)

[Modelo lógico 17](#_Toc182952001)

[Modelo relacional 18](#_Toc182952002)

[Código SQL 19](#_Toc182952003)

[Factibilidad económica 24](#_Toc182952004)

[Flujo de caja 24](#_Toc182952005)

[VAN valor actual neto 24](#_Toc182952006)

[TIR tasa interna de retorno 24](#_Toc182952007)

[Conclusión en español 24](#_Toc182952008)

[Conclusión en inglés 24](#_Toc182952009)

[Acrónimos 24](#_Toc182952010)

[Bibliografía 25](#_Toc182952011)

[Anexos 25](#_Toc182952012)

Resumen ejecutivo (español)

**Titulo del Proyecto:** Plataforma Integral De Gestión de Cliente y Ordenes de Transporte.

**Objetivos:** Desarrollar una solución tecnológica que centralice y automatice la gestión de clientes y ordenes de trasporte, para mejorar la eficiencia operativa de Setralog, una empresa dedicada al trasporte 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 frente a la competencia.

Tasa de errores del 18% en la creación de órdenes.

Perdida 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**

El proyecto APT propone una plataforma basada en la web que permitirá:

**Centralizar la Información:** Registro y gestión unificada de la información de datos de clientes y ordenes 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 ordenes 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 bajo un marco ágil “Scrum”, el cual estará divido en tres fases:

Levantamiento de requerimientos y análisis de negocios (20 días)

Diseño, desarrollo y pruebas del sistema (55 días)

Implementación, marcha blanca y capacitación (15 días)

**Competencias**

El desarrollo de Proyecto APT permite a los integrantes aplicar competencias técnicas (programación, base de datos, seguridad), blandas (trabajo en equipo, resolución de problemas) y de gestión (planificación y liderazgo), alineándose con las demandas del mercado y fortaleciendo su perfil profesional.

**CONCLUSION NO SE SI SE PONDRA. JONAAAAAAA**

Introducción

El proyecto APT 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 fue fundada en 2018 por la ingeniera en logística Patricia González, quien actualmente ocupa el cargo de gerente general. Al comenzar como una pyme, Setralog enfrentó varios desafíos, entre ellos la gestión eficiente de la información de clientes y órdenes de transporte. La administración de la empresa se realiza mediante hojas de cálculo en Excel, y el medio de comunicación con los clientes para solicitar órdenes de transporte se basa principalmente en correo electrónico.

En la actualidad, Setralog ha experimentado un crecimiento tanto en capacidad operativa como en la 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 problemas 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.

Esta forma de trabajo es lenta y poco eficaz, propensa a errores, y genera un aumento en los tiempos de respuesta de hasta un 45% en comparación con la competencia, además de una tasa de errores cercana al 18% en la generación de órdenes de transporte. Todas estas ineficiencias han llevado a la pérdida de 13 clientes en lo que va del 2024.

Si bien pueden parecer problemas menores en una operación diaria, la suma de estos se traduce en costos significativos al final de cada mes y en una experiencia de servicio deficiente para los clientes, afectando tanto a la productividad a la reputación de la empresa.

Dado el contexto el APT y el tiempo para su desarrollo, la solución tiene como objetivo desarrollar un sistema para centralizar la información relacionada con el cliente y además de un sistema de solicitudes y gestión de ordenes de transporte. Esta solución basada en código web permitirá el acceso mediante internet no solo del personal de Setralog, sino que también de los clientes, generando a si un sistema de gestión integral y unificada de información

Esto permitirá que tanto los colaboradores de Setralog y sus clientes, puedan autogestionar sus datos y darse de alta como clientes y a su vez permita la creación de las solicitudes de transporte de manera autónoma por parte de los clientes.

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 también se enmarca en un contexto laboral en constante evolución, donde las tecnologías de la información y comunicación son claves para un trabajo sólido y que genere competitividad. En este sentido, la solución ofrece a los futuros profesionales de Ingeniería en Informática la oportunidad de aplicar y fortalecer las competencias adquiridas en un caso real, abarcando diversas técnicas y una gestión bien hecha, aplicando conocimientos en diversos ramos.

Descripción del problema

La problemática se centra en la optimización de procesos de la empresa Setralog, una empresa de transporte de carga especializada en productos de diversos tipos. Actualmente, Setralog gestiona sus operaciones de forma manual, utilizando hojas de cálculo Excel para registrar clientes y pedidos, lo que implica una gestión lenta que se traduce en un 45% más de retraso en los tiempos de entrega, además de estar propenso a cometer errores en el momento del ingreso de los datos lo que se traduce en una tasa de error de un 18%.

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, perdida de imagen ante la competencia lo que ha desencadenado la pérdida de clientes.

En este flujo se puede apreciar los procesos que actualmente están involucrados en el proceso de una solicitud de transporte.

Diagrama

Descripción generada automáticamente

Este proceso de puede apreciar de mejor manera en el anexo xxxx

Problema por resolver

El problema que busca resolver el proyecto APT es la ineficiencia en la gestión de clientes y órdenes de transporte de la empresa Setralog, que actualmente utiliza métodos manuales, como hojas de cálculo en Excel y correo electrónico, para registrar y organizar la información, lo cual demanda una gran cantidad de tiempo para sus registros. Para un pedido estándar el tiempo de lectura de un correo va desde los 30 minutos a 4 horas. Luego pasa por la generación de solicitud de transporte en el cual se deben verificar que estén todos los datos del cliente, lugar de retiro, lugar de despacho, tipo y cantidad de carga. Todo este proceso puede tomar entre 30 a 90 minutos. Para que el pedido quede a la espera de ser asignado a un vehículo de despacho.

Este sistema no solo es lento y propenso a errores, sino que también carece de centralización y unificación, lo que dificulta el acceso rápido a datos y seguimientos de pedidos obligando a los clientes a tener que consultar ya sea por correo o teléfono para saber el estado de su orden de transporte.

Esta falta de agilidad y precisión afecta tanto a los empleados como a los clientes de la empresa, limitando la calidad del servicio y la competitividad de Setralog en el mercado.

Solución al problema.

El proyecto APT aborda la ineficiencia en la gestión de cliente y ordenes de trasporte de Setralog mediante la creación de una plataforma integral basada en la web, diseñada para automatizar, optimizar y unificar los procesos operativos. La solución cuenta con componentes clave como:

1. **Centralización de datos**

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

1. **Automatización de Procesos**

* **Desarrollo de módulos específicos para:**
* Gestión de clientes: Registro, actualización y consulta de datos en tiempo real.
* Gestión de órdenes de trasporte: 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.
* ---------------------------------

Esta solución moderniza y automatiza los procesos de Setralog, permitiendo una gestión mas eficiente, precisa y alineada con las demandas del mercado actual.

Justificación

La justificación del proyecto APT radica en la necesidad de mejorar la eficiencia operativa y la precisión en la gestión de cliente y órdenes de trasporte de Setralog, una empresa dedicada al transporte de productos sensibles.

Dados que los procesos que hoy por hoy maneja Setralog afectan en la calidad del servicio y la satisfacción del cliente, el desarrollo de una plataforma de gestión integral permitirá a Setralog optimizar y automatizar toda la información que se ingresa en sistema además de mejorar los tiempos de procesamiento de estos.

Esto contribuirá a reducir errores de registro y mejorar los tiempos de respuesta. Además de una reducción en los costos de operación no teniendo que generar horas extras para solucionar los errores y facilitando la tarea a los administrativos, disponiendo de más tiempo para destinar a otras tareas. Esto sumando a la trazabilidad de los pedidos, hace que Setralog se alinee a las demandas actuales del mercado de trasporte, sino que también reforzara su posición competitiva, generando valor tanto para la empresa como sus trabajadores.

Objetivo general

* Desarrollar una plataforma integral que centralice y automatice la gestión de clientes y ordenes de transporte para mejorar la eficiencia operativa.
* 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.
* Aumentar 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 disminuir al menos de un 18% a un 5 % la tasa de errores en la generación de ordenes de transporte y reducir de un 45% a un 10% los tiempos de relacionado con el generación y procesamiento de la orden de trasporte. Esto se traduce en una mejor satisfacción al cliente y aumenta la competitividad ante el mercado.

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 evitara errores ya que el propio cliente se dará de alta en el sistema con los datos que el indique.

* Desarrollo de módulo de orden de transporte: Este módulo permite la creación, asignación y seguimiento de las ordenes de transporte. Esto evitará la alta demora al generar una orden de trasporte 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, Amazón Web Services según la factibilidad del cliente).
* Seguridad de la información: Implementar medidas de seguridad para proteger la información de los clientes y ordenes 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

Competencias Técnicas.

* ***Programación de aplicaciones:*** *Propramacación en distintos lenguajes (JavaScript, Python, HTTP, CSS, etc.)*
* ***Manejo de base de datos:*** *Diseño, implementación, optimización de consultas, mantenimiento y respaldo de bases de datos.*
* ***Seguridad informática:*** *Implementación de medidas de seguridad, pruebas de penetración, gestión de seguridad, y aplicación de normativas y estándares de seguridad de la información.*

Competencias Blandas.

* ***Resolución de problemas:*** *Análisis y evaluación de situaciones complejas, aplicando creatividad e innovación en las soluciones.*
* ***Trabajo en equipo:*** *Colaboración con otros profesionales.*
* ***Adaptabilidad:*** *capacidad de ajustarse y responder de manera efectiva a cambios situaciones o desafíos imprevistos.*
* ***Comunicación efectiva:*** *Habilidades de presentación y redacción, explicación de conceptos técnicos a público no técnico, escucha activa y empática.*

Competencias de Gestión.

* ***Gestión de Proyecto:*** *Planificación, organización, gestión de recursos, presupuesto, monitoreo y control de proyectos.*
* **Liderazgo:** Motivación, dirección de equipos, evaluación de desempeño, Delegación de tareas y responsabilidades.
* **Gestión del cambio:** implementación de estrategias para el cambio en las organizaciones, imparto, comunicación y gestión de la resistencia a los cambios.
* **Conocimiento del negocio:** comprensión de la operación de negocio, entendimiento de procesos. Conocimiento del mercado.

Acta de constitución de proyecto.

----------------------------Pendiente----------------------------

Asignación de roles

----------------------------Pendiente----------------------------

Metodología

La metodología para desarrollar el Proyecto APT estará basada en el marco de proyecto ágiles Scrum, adecuado para tecnologías debido a su enfoque en la flexibilidad, la colaboración y la entrega iterativa. A continuación, se explica detalladamente como se implementará cada fase de la metodología, desde la planificación hasta el despliegue y la marcha blanca del sistema.

A continuación, se detallan las fases de este proyecto desacuerdo a la carta Gantt de planificación.

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 Steakholders 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 codificaran 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 puede apreciar las tareas definidas en el proyecto y duración de estas, además de servir como una herramienta 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 faces o hitos cada uno con sus correspondientes tareas y tiempos de ejecución.

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

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

El anexo 1 se puede apreciar la carta Gantt en mayor escala para su interpretación.

Implementación del proyecto

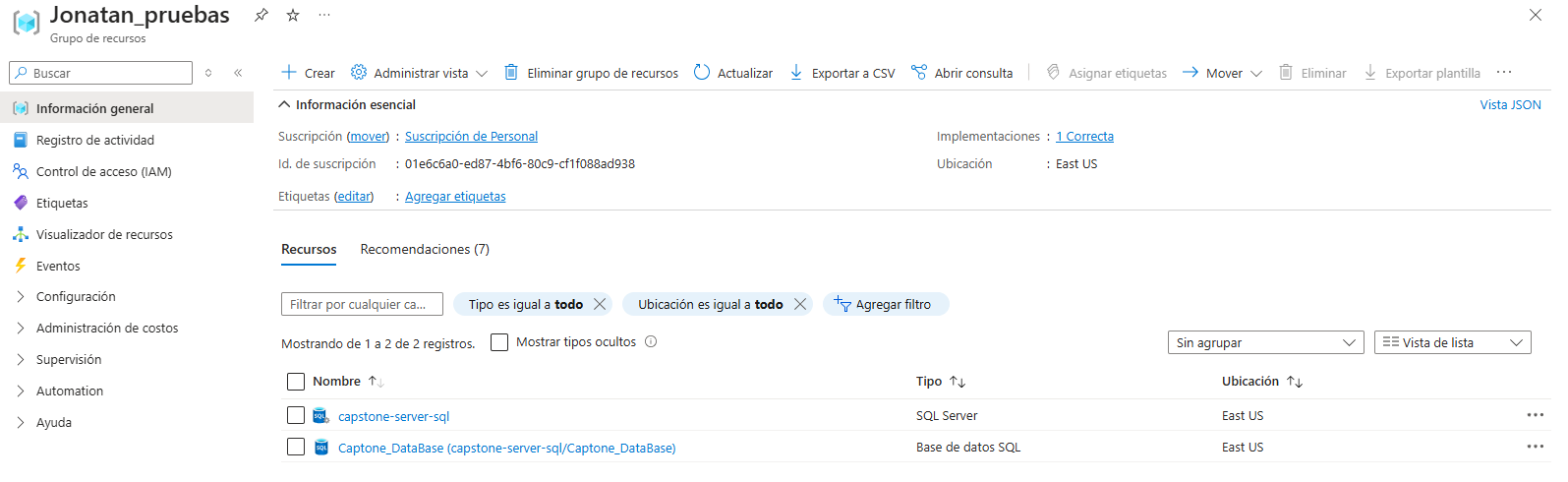
Base de datos

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.



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.

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.

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.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Detalles de código en anexo xxxx

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

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 todo lo que el usuario puede apreciar visualmente mediante un navegador web y contiene el diseño visual con el cual se 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 son otorgadas mediante el lenguaje JavaScript el cual añade interactividad y funcionalidad a la estructura de la página web, además de realizar peticiones hacia los servidores para el procesamiento de la información.

HTML

Texto

Descripción generada automáticamente

CSS

Texto

Descripción generada automáticamente

Página web.

Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente

Factibilidad económica

----------------------------Pendiente----------------------------

Flujo de caja

----------------------------Pendiente----------------------------

VAN valor actual neto

----------------------------Pendiente----------------------------

TIR tasa interna de retorno

----------------------------Pendiente----------------------------

Conclusión en español

----------------------------Pendiente----------------------------

Conclusión en inglés

----------------------------Pendiente----------------------------

Acrónimos

*JavaScript, Python, HTTP, CSS*

*Azure*

*Data Base*

*Microsoft*

*SQL DDL*

*Node J.S*

*FrameWork*

Bibliografía

----------------------------Pendiente----------------------------

Anexos

----------------------------Pendiente----------------------------

Insertar flujo de proceso actual (antes de la solución)

Carta Gannt

Insertar carta Gantt en formato grande.

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\_ordenes (  ordenes\_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\_ordenes ADD CONSTRAINT detalle\_ordenes\_pk PRIMARY KEY ( ordenes\_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\_ordenes (  cod\_estado VARCHAR(1) NOT NULL,  nombre\_estado VARCHAR(15)  );  ALTER TABLE estado\_ordenes ADD CONSTRAINT estado\_ordenes\_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 ordenes (  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\_ordenes\_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 ordenes ADD CONSTRAINT ordenes\_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\_ordenes  ADD CONSTRAINT detalle\_ordenes\_ordenes\_fk FOREIGN KEY ( ordenes\_num\_orden )  REFERENCES ordenes ( 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 ordenes  ADD CONSTRAINT ordenes\_cod\_despacho\_fk FOREIGN KEY ( cod\_despacho\_cod\_retiro )  REFERENCES cod\_despacho ( cod\_despacho );  ALTER TABLE ordenes  ADD CONSTRAINT ordenes\_cod\_despacho\_fkv1 FOREIGN KEY ( cod\_despacho\_cod\_despacho )  REFERENCES cod\_despacho ( cod\_despacho );  ALTER TABLE ordenes  ADD CONSTRAINT ordenes\_direccion\_retiro\_fk FOREIGN KEY ( direccion\_retiro\_cod\_retiro )  REFERENCES direccion\_retiro ( cod\_retiro );  ALTER TABLE ordenes  ADD CONSTRAINT ordenes\_direccion\_retiro\_fkv1 FOREIGN KEY ( direccion\_retiro\_cod\_retiro1 )  REFERENCES direccion\_retiro ( cod\_retiro );  ALTER TABLE ordenes  ADD CONSTRAINT ordenes\_empresas\_fk FOREIGN KEY ( empresas\_rut )  REFERENCES empresas ( rut );  ALTER TABLE ordenes  ADD CONSTRAINT ordenes\_estado\_ordenes\_fk FOREIGN KEY ( estado\_ordenes\_cod\_estado )  REFERENCES estado\_ordenes ( cod\_estado );  ALTER TABLE ordenes  ADD CONSTRAINT ordenes\_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 ); |