 **PROYECTO:**

“ANÁLISIS DEL PROCESO OPERATIVO DE CAPTURA DE DATOS Y DESARROLLO DE UN SISTEMA DIGITAL WEB PARA LA GESTIÓN EFICIENTE DE REPORTES PPV/CANCELS DE LA EMPRESA AMETEK LAMB MOTORES S de RL de CV”

**REPORTE DE ESTADÍA PARA OBTENER EL**

**TÍTULO DE:**

**INGENIERIA EN DESARROLLO Y GESTIÓN DE SOFTWARE**

**PRESENTA:**

**IAN ALBERTO CASTRO LOPEZ**

**CD. REYNOSA, TAMAULIPAS AGOSTO 2025.**

Dedicatoria.

Agradecimientos.

Resumen.

Abstract.

Índice

**Tema Página**

[**Dedicatoria**. i](#_Toc191406933)

[**Agradecimientos**. ii](#_Toc191406934)

[**Resumen**. iii](#_Toc191406935)

[**Abstract**. iv](#_Toc191406936)

[**Índice** v](#_Toc191406937)

[**Índice** **de** **figuras**. vi](#_Toc191406938)

[**Índice** **de** **gráficas**. vii](#_Toc191406939)

[**Índice** **de** **tablas**. vii](#_Toc191406940)

[**Capítulo** **1. Introducción** 8](#_Toc191406941)

[1.1. Antecedentes 9](#_Toc191406942)

[1.2. Definición del problema 10](#_Toc191406943)

[1.3. Justificación 11](#_Toc191406944)

[1.4. Objetivo 13](#_Toc191406945)

[1.5. Pregunta 13](#_Toc191406946)

[**Capítulo** **2. Marco Teórico**. 14](#_Toc191406947)

[2.1. Gestión de Reportes 14](#_Toc191406948)

[2.2. Mantenimiento de Equipos de Cómputo 15](#_Toc191406949)

[2.3. Monitoreo de Equipos de Cómputo 15](#_Toc191406950)

[2.4. Aplicaciones Web para la Gestión de Reportes 16](#_Toc191406951)

[2.5. Automatización de Procesos 16](#_Toc191406952)

[2.6. Beneficios de un Sistema de Gestión de Reportes 16](#_Toc191406953)

[2.7. Herramientas Tecnológicas para la Gestión de Reportes 17](#_Toc191406954)

[**Capítulo** **3.** **Procedimiento**. 18](#_Toc191406955)

[3.1. Requerimientos de las necesidades de las partes interesadas y los usuarios. 18](#_Toc191406956)

[3.1.1. Requerimientos funcionales. 18](#_Toc191406957)

[3.1.2. Requerimientos no funcionales. 22](#_Toc191406958)

[3.2. Método de la propuesta 24](#_Toc191406959)

Índice de figuras.

**Tema Página**

[Figura 1. Resumen de las etapas de la propuesta 24](#_Toc191406898)

Índice de gráficas.

**Tema Página**

Índice de tablas.

**Tema Página**

[Tabla 1. Requerimiento funcional RF-001 Login. 19](#_Toc191406881)

[Tabla 2. Requerimiento funcional RF-002 Reportes 20](#_Toc191406882)

[Tabla 3. Requerimiento funcional RF-003 Consulta 20](#_Toc191406883)

[Tabla 4. Requerimiento funcional RF-004 Administración 21](#_Toc191406884)

[Tabla 5. Requerimiento funcional RF-005 Administración de Usuarios y Laboratorios 21](#_Toc191406885)

[Tabla 6. Requerimiento funcional RF-006 Historial de reportes 22](#_Toc191406886)

[Tabla 7. Requerimiento no funcional RNF-001 Dispositivos compatibles 22](#_Toc191406887)

[Tabla 8. Requerimiento no funcional RNF-002 Facilidad de uso 23](#_Toc191406888)

[Tabla 9. Requerimiento no funcional RNF-003 Protección de Datos 23](#_Toc191406889)

[Tabla 10. Requerimiento no funcional RNF-004 Escalabilidad 24](#_Toc191406890)

Capítulo 1. Introducción

La empresa Ametek Lamb Motores es una empresa global manufacturera de instrumentos electrónicos y electromecánicos. Fundada en 1930, la compañía mantiene su sede en Berwyn, Pennsylvania. Ametek opera alrededor de 220 plantas en todo el mundo y sirve a varias industrias, incluyendo aerospacial, energía, salud y semiconductores. Actualmente cuenta con dos plantas en Reynosa; la planta 1 ubicada en el Parque del Norte, y la planta 2 ubicada en el Parque Villa Florida.

Ametek cuenta con múltiples unidades de negocio, las cuales se encargan de manufacturar y ensamblar ciertas partes y ensamblajes para su posterior envío al destinatario. Las unidades de negocio de Ametek requieren en su totalidad de comunicación con el departamento de Supply Chain. Dentro de este departamento se lleva a cabo la organización y gestión de los recursos involucrados en la creación y venta de los productos, desde el envío de materia prima por parte del proveedor, hasta su eventual entrega al usuario final.

El departamento de Supply Chain ubicado en la planta 1 lleva a cabo diferentes actividades que requieren la creación y gestión de reportes, estos reportes son variados y en su mayoría son solicitados con base a la demanda por parte de corporativo. Entre los reportes llevados a cabo se hace énfasis en dos que son realizados semanalmente, lo que provoca una tarea repetitiva que obstruye con actividades de mayor urgencia y que requieren de más tiempo de análisis, ya que en su mayoría estos reportes manejan presupuestos, pronósticos de ahorros y pérdidas.

Por ello, es importante automatizar estos procesos repetitivos para priorizar las actividades financieras y lograr llevar a cabo planes de acción para mitigar las posibles contingencias que puedan ocurrir. En este contexto, un alumno de la carrera de Tecnologías de la Información de la Universidad Tecnológica de Tamaulipas Norte ha identificado una necesidad importante dentro de la empresa: la ausencia de un sistema web eficaz para gestionar los reportes PPV/Cancels.

La falta de una herramienta adecuada para el seguimiento de los reportes puede conllevar a realizar fallos manuales en cálculos debido a procesos repetitivos, errores al utilizar maros debido a formatos obsoletos y pérdida de tiempo por tareas cuya importancia no es urgente.

* 1. Antecedentes

El departamento de Supply Chain se encarga de apoyar a distintas unidades de negocio, cada una de ellas realiza compras de partes y posteriormente las ensambla. Para comprar partes es necesario contactar con los proveedores, una vez obtenidos se debe de realizar el ensamblado de estos componentes.

Mediante un reporte del sistema MRP se lleva el registro de cada parte ensamblada y comprada, este reporte es enviado al correo electrónico del empleado para posterior gestión. Este reporte MRP contiene información de las compras y ensambles de las partes de manufacturero.

Una vez filtrada la información, se crea uno completamente nuevo que cuenta con información de cada estatus de los ensambles o MAKES. Cada ensamble tiene un estatus diferente; y en el reporte creado por el empleado se centra en los CANCELS, que son aquellos ensambles los cuales se han cancelado por diferentes motivos.

Así mismo, reportes de órdenes de compra son compartidos al empleado, estos contienen todas las unidades de negocio con sus respectivos compradores y proveedores. Debido a esto, se realiza otro en donde se toman los proveedores de cada unidad de negocio para hacer un resumen de estas compras en los distintos proveedores, esto, con el objetivo de identificar cortos y posibles ganancias por cada uno y gestionar los gastos realizados entre estos.

Los problemas más comunes para reportar son los siguientes:

* **Captura manual de los datos:** El empleado docente debe inspeccionar cada uno de los reportes para capturar sus datos y posteriormente crear el reporte correspondiente.

**Formato obsoleto:** El formato de los reportes es obsoleto y se debe de actualizar manualmente, para poder manipular su información cómodamente.

1.2. Definición del problema

El departamento de Supply Chain enfrenta una probemática que afecta su operativa diaria: la gestión y creación ineficiente de los reportes PPV y cancels. Actualmente, el proceso de captura de estos reportes se lleva a cabo manualmente, lo que ha demostrado ser poco efectivo y muy propenso a errores.

Esta situación se debe principalmente a la falta de una aplicación web o un sistema digital que permita llevar un control adecuado y centralizado de estos registros. Como consecuencia, el empleado, a pesar de realizar estos reportes manualmente, entrado archivo por archivo y unirlos en uno solo, esto da como resultado que las tareas se vuelvan redundantes y provoquen pérdida de tiempo que a su vez se traduce en dinero, pues estos reportes contienen información crucial para llevar planes de acción con el propósito de mitigar estas pérdidas, así como también encontrar vías viables para el ahorro de cada inversión.

La ausencia de un sistema de gestión y creación eficiente tiene un impacto directo en el tiempo de captura cuando se lleva a cabo la creación del reporte semanal. Esta demora no solo afecta el tiempo de reacción para llevar a cabo los planes de acción, sino que además de que no permite que se les dé prioridad a tareas de mayor magnitud como proyectos grandes donde se requiere mayor tiempo de trabajo.

En resumen, la falta de un sistema eficiente para la gestión y creación de reportes PPV y cancels en el departamento genera un contratiempo al hacer los cálculos manualmente mediante tablas dinámicas para posteriormente poblar la plantilla que mantiene la información actualizada semanalmente de cada unidad de negocio.

1.3. Justificación

Supply Chain es un departamento esencial para la toma de acciones en cuestión de finanzas o de control de materiales, ya que permite que las unidades de negocio dentro de la empresa puedan tener el mejor manejo posible de las inversiones de cada ensamble, de cada compra, etc. Para lograr esto, se tienen estos reportes PPV y cancels, sin embargo, estos reportes a pesar de ser de gran ayuda; requieren de mucho trabajo manual, lo que conlleva a errores manuales y estos consideran una inversión de tiempo prolongada que bien podría ser invertido en tareas de mayor importancia.

Actualmente el proceso manual de creación y manejo de reportes carece de un sistema centralizado que garantice su realización eficaz para su posterior envío a revisión por parte de corporativo quien es el departamento encargado de tomar acciones que favorezcan a las unidades de negocio dentro de la división, tal como la reducción y ahorro de inversión por partes, así como también la identificación de cortos.

Al llevar a cabo este proceso de manera manual, se tiene el problema de que se genera un tiempo prolongado para llevar a cabo la compleción de estos reportes, lo que sugiere un problema, puesto a que estos requieren cálculos manuales y a un empleado especializado para llevar a cabo estas tareas que no son sencillas.

La implementación de un sistema digital web para gestionar y crear los reportes de manera eficiente es una solución necesaria. Este sistema permitirá automatizar el registro de los datos, los cálculos y complementar los reportes en un tiempo reducido, logrando que esta tarea mayormente repetitiva y larga se pueda reducir enormemente, así como también ya no requeriría de cálculos manuales, puesto que el propio sistema es quien se encarga de llevar a cabo estos procesos.

Asimismo, el acceso histórico de los reportes permitirá llevar a cabo un control de ellos, permitiendo que los usuarios finales puedan tener una plataforma que les brinde un mejor control de estos reportes y permita una futura incorporación de reportes nuevos y, así mismo promoviendo una oportunidad para que nuevos compañeros lleven a cabo este proceso de manera más simple y controlada.

La justificación de este proyecto radica en la importancia de ofrecer un control de reportes efectivo para que corporativo tome acciones estratégicas para cada unidad de negocio y se tenga un menor índice de cancelaciones, así como también se minoricen las inversiones en órdenes de compra. Un sistema eficiente no solo mejorará la experiencia de los usuarios al reducir los tiempos de llenado, sino que contribuirá a la automatización de su compleción además de mejorar la logística llevada a cabo dentro del departamento.

1.4. Objetivo

Analizar el proceso operativo de la captura de datos y desarrollo de un sistema digital web para automatizar en un 20% la creación y gestión de los reportes PPV/Cancels del departamento de Supply Chain en la empresa AMETEK Lamb Motores S de RL de CV.

1.5. Pregunta

¿De qué manera la Implementación de un sistema digital web se reportes optimizará la creación y gestión de los reportes PPV y Cancels?

Capítulo 2. Marco Teórico.

En este capítulo se plantea dar a conocer de forma clara y precisa sobre el análisis de datos, puesto que son el motivo principal para llevar a cabo la gestión de los reportes, así como de su constante monitoreo y trazabilidad.

2.1. Gestión de Reportes

La gestión de reportes es un proceso sistemático para recopilar, analizar y entregar información clave que apoya la toma de decisiones. Según (Camacho, 2021), se define como “una herramienta de evaluación del desempeño construido con base en la información del negocio que se ha recopilado”.

En entornos corporativos, estos informes permiten monitorear KPIs, detectar desviaciones y establecer correctivas, haciéndolos sumamente importantes para garantizar la eficiencia de los procesos internos dentro de las áreas de desarrollo de informes según (Nuñez, 2022).

Por ello, es fundamental implementar herramientas que permitan automatizar y agilizar este proceso, asegurando que los reportes lleguen a las áreas correspondientes de manera inmediata y también los empleados del departamento prioricen otras tareas.

2.2. Uso de reportes

Según, (Nuñez, 2022), “Un informe de gestión es un documento corporativo de mucha utilidad para el mantenimiento y crecimiento de una empresa. Contiene información sobre proyectos y áreas operativas y financieras, lo cual permite monitorear y evaluar el rendimiento de un negocio. Entre los datos de un informe de gestión, se incluyen los presupuestos asignados, el flujo de caja, la rentabilidad, el balance financiero, el rendimiento de los equipos, los problemas operativos, la respuesta del cliente, etc.”.

Los informes se pueden destinar a niveles gerenciales y juegan un papel clave en la coherencia estratégica, pues su función es evaluar el rendimiento en diferentes áreas para tomar decisiones correctas y fundamentadas. Se considera que los informes ayudan a tomar las decisiones correctas a los responsables de estas áreas, casi siempre siendo el área financiera quien toma el mando.

2.3. Aplicaciones Web para la Gestión de Reportes

Las aplicaciones web proporcionan una plataforma centralizada para generar, almacenar y visualizar reportes de forma accesible y en tiempo real. Permiten la centralización de información y trazabilidad de acciones desde cualquier dispositivo.

“Una app para informes no solo mejora la eficiencia operativa, sino que también proporciona información muy valiosa para tomar decisiones empresariales. Y es que, gracias a la digitalización, los responsables de área pueden acceder a métricas clave de rendimiento en tiempo real, como el número de tareas asignadas, resueltas y pendientes, los tiempos de desplazamiento y el tiempo de resolución promedio, entre otros.” (ABARCA, 2023)

2.4. Automatización de Procesos

La automatización de procesos consiste en sustituir actividades manuales y repetitivas por acciones controladas por software. Esto no solo mejora la eficiencia, sino que también disminuye significativamente la probabilidad de errores humanos.

Dentro de la automatización se encuentran los siguientes procesos: Lectura y validación de archivos, extracción y filtrado de datos, clasificación automática y generación y exportación del documento final. Automatizar este flujo permite que los reportes estén listos en minutos en lugar de horas, liberando así tiempo para tareas de mayor magnitud.

2. 5. Beneficios de un Sistema de Gestión de Reportes

Implementar un sistema web para la gestión automatizada de reportes trae consigo múltiples beneficios, dentro de los cuales la primera es que reduce el tiempo de elaboración de reportes en un 15%, elimina los errores humanos derivados de las operaciones manuales, cuenta con una mayor trazabilidad, permitiendo saber quién genera cada reporte y cuándo este es realizado.

El historial y respaldo automático de la aplicación, así como la estandarización de los formatos pueden mejorar el flujo de la información entre departamentos y facilitar la toma de decisiones en tiempo real, lo que permite que el equipo corporativo tenga acceso inmediato a la información sin necesidad de esperar a que los empleados lo completen desde cero.

2.6. Herramientas Tecnológicas para la Gestión de Reportes

Para llevar a cabo la implementación de la aplicación se utilizan herramientas modernas del entorno de desarrollo web, como lo son JavaScript, cuya biblioteca de JavaScript permite la construcción de interfaces de usuario dinámicas; Node.Js y Express permiten crear un servidor *backend* que procese los archivos y maneje la lógica del servidor.

SQL Server es la base de datos SQL que permite guardar el historial de reportes generados, usuarios y configuraciones personalizadas.

2.7. *Backend*

“Backend, conocido como “el lado del servidor”, se refiere a la parte de una aplicación web o sitio web que no es visible para el usuario final, pero que es esencial para su funcionamiento. En él se configuran todos los aspectos lógicos de una página web o aplicación; abarca la lógica, el almacenamiento de datos y las funciones de seguridad necesarias para que una aplicación funcione correctamente y sea fiable, de tal forma que todas las acciones solicitadas en la página web sean ejecutadas de manera correcta.” (Ken, 2023)

2.8. *Frontend*

“Frontend, también conocido como el lado del cliente, es la parte visual de un sitio web, incluyendo la estructura del sitio, estilos, colores, fondos, tamaños, animaciones y efectos. Es la parte de la página con la que interactúan los usuarios de esta, todo lo que ve y experimenta de forma directa.

El desarrollo frontend, consiste en darle estructura a los datos que aparecen en el sitio web, su finalidad es optimizar la experiencia del usuario, encargándose de dar definición a la jerarquía de la información que se visualiza, así como su distribución: márgenes, colores, recursos, audiovisuales, animaciones, entre otros; dando como resultado una página web.” (Ken, gluo, 2023)

2.9. Base de datos

“Una base de datos es una recopilación de información que se almacena digitalmente en una computadora, en un servidor o en la nube. Las bases de datos pueden ser muy simples o bastante complicadas, dependiendo de la estructura y organización de los datos. El sistema de bases de datos generalmente se compone de la información en sí y un sistema de gestión de bases de datos (DBMS), lo que permite a los usuarios acceder, actualizar, analizar y gestionar fácilmente la información.” (Nutanix, 2023)

Base de datos relacional

Según (Oracle, 2021), “Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí. Las bases de datos relacionales se basan en el modelo relacional, una forma intuitiva y directa de representar datos en tablas. En una base de datos relacional, cada fila en una tabla es un registro con una ID única, llamada clave. Las columnas de la tabla contienen los atributos de los datos y cada registro suele tener un valor para cada atributo, lo que simplifica la creación de relaciones entre los puntos de datos.”

Base de datos no relacional

“Una base de datos no relacional, también conocida como base de datos NoSQL (Not Only SQL), es un tipo de sistema de gestión de bases de datos diseñado para manejar y almacenar datos de manera flexible y escalable, sin seguir el modelo de tablas y relaciones de las bases de datos relacionales tradicionales. A diferencia de las bases de datos relacionales, las bases de datos no relacionales permiten almacenar y recuperar datos de forma más libre y menos estructurada. Están diseñadas para manejar grandes volúmenes de datos no estructurados o semi estructurados, como documentos, gráficos, clave-valor o columnas.” (fpsuperiorufv, n.d.)

Capítulo 3. Procedimiento.

En este capítulo se describen los requerimientos funcionales y no funcionales que se quieren implementar en el proyecto, donde se detallan las necesidades del cliente y de los usuarios que utilizan la aplicación. Es fundamental establecer estos requerimientos, ya que proporcionan una guía clara para el desarrollo del sistema, aseguran que se cumplan las expectativas de todos y facilitan la planificación y gestión del proyecto.

3.1. Requerimientos de las necesidades de las partes interesadas y los usuarios.

Lo que necesitan los usuarios y partes interesadas del sistema. Para ellos, es importante que la aplicación permita acceder de forma segura, generar reportes detallados. También es clave que la aplicación sea fácil de usar, para que cualquiera pueda navegar sin problemas.

3.1.1. Requerimientos funcionales.

Establecer los requerimientos funcionales es muy importante ya que le dan claridad y enfoque a las ideas y necesidades que se piensan realizar y así todos los involucrados en este proyecto pueden comprender mejor los objetivos y las funcionalidades que se están por desarrollar. Al tener claro estos requerimientos se reduce el riesgo de cambios constantes y malentendidos a lo largo del desarrollo.

El requerimiento funcional 001 menciona que la aplicación debe permitir a los usuarios iniciar sesión de manera segura mediante un sistema de autenticación. Esto quiere garantizar que solo los usuarios autorizados puedan acceder a las funciones y datos de la aplicación, protegiendo la información sensible y asegurando el control de acceso al sistema.

El requerimiento funcional 001 menciona que la aplicación debe permitir a los usuarios iniciar sesión de manera segura mediante un sistema de autenticación. Esto quiere garantizar que solo los usuarios autorizados puedan acceder a las funciones y datos de la aplicación, protegiendo la información sensible y asegurando el control de acceso al sistema.

|  |  |
| --- | --- |
| Identificación de requerimiento | RF-001 |
| Nombre del requerimiento | *Login* |
| Descripción del requerimiento | Los usuarios pueden acceder a su cuenta en la aplicación web a través de credenciales de usuario. |
| Prioridad de requerimiento | Alta |

Tabla 1. Requerimiento funcional RF-001 Login.

El requerimiento funcional 002 menciona que la aplicación debe permitir a los usuarios crear reportes sobre fallas en los equipos de los laboratorios. Esto quiere garantizar que los reportes sean realizados de manera eficiente mediante la carga de documentos por parte del usuario.

|  |  |
| --- | --- |
| Identificación de requerimiento | RF-002 |
| Nombre del requerimiento | *Carga de documentos* |
| Descripción del requerimiento | Los usuarios deben poder agregar archivos a la aplicación. |
| Prioridad de requerimiento | Alta |

Tabla 2. Requerimiento funcional RF-002 Reportes

El requerimiento funcional 003 menciona que la aplicación debe permitir a los usuarios visualizar los reportes generados. Esto quiere garantizar que los empleados puedan estar informados sobre los reportes generados en su entorno, promoviendo la transparencia y mejorando la comunicación dentro del mismo.

|  |  |
| --- | --- |
| Identificación de requerimiento | RF-003 |
| Nombre del requerimiento | *Consulta de Reportes* |
| Descripción del requerimiento | Los usuarios podrán visualizar los reportes a gestionar, incluyendo detalles como el autor y la fecha de creación. |
| Prioridad de requerimiento | Alta |

Tabla 3. Requerimiento funcional RF-003 Consulta

El requerimiento funcional 004 menciona que la aplicación debe permitir a los usuarios y administradores acceder a un historial de reportes anteriores. Esto quiere garantizar que la información de reportes pasados esté disponible para su consulta, con el objetivo de analizar datos y mejorar la gestión de incidencias.

|  |  |
| --- | --- |
| Identificación de requerimiento | RF-004 |
| Nombre del requerimiento | *Historial de Reportes* |
| Descripción del requerimiento | Tanto los usuarios como los administradores podrán acceder a un historial de reportes previos, con la posibilidad de filtrar por fechas, estatus y laboratorio. |
| Prioridad de requerimiento | Media |

Tabla 4. Requerimiento funcional RF-004 Historial

3.1.2. Requerimientos no funcionales.

También es importante mencionar los requerimientos no funcionales ya que son fundamentales para definir características esenciales como la disponibilidad, rendimiento, escalabilidad y usabilidad. Los requerimientos no funcionales también son esenciales para el mantenimiento y la escalabilidad del sistema.

El requerimiento no funcional 001 menciona que la aplicación debe estar diseñada exclusivamente para su uso en computadoras de escritorio y laptops.

|  |  |
| --- | --- |
| Identificación de requerimiento | RNF-001 |
| Nombre del requerimiento | *Dispositivos compatibles* |
| Descripción del requerimiento | La aplicación debe estar optimizada para su uso exclusivo en computadoras de escritorio y laptops, sin soporte para dispositivos móviles. |
| Prioridad de requerimiento | Alta |

Tabla 5. Requerimiento no funcional RNF-001 Dispositivos compatibles

El requerimiento no funcional 002 menciona que la aplicación debe contar con una interfaz intuitiva y fácil de usar. Esto debe garantizar que cualquier usuario, sin necesidad de capacitación, pueda navegar en la plataforma de manera sencilla, accediendo a las funciones principales sin dificultades.

|  |  |
| --- | --- |
| Identificación de requerimiento | RNF-002 |
| Nombre del requerimiento | *Facilidad de uso* |
| Descripción del requerimiento | La interfaz debe ser clara y sencilla, permitiendo a los usuarios navegar fácilmente por la aplicación y acceder a sus funciones sin necesidad de capacitación adicional. |
| Prioridad de requerimiento | Alta |

Tabla 6. Requerimiento no funcional RNF-002 Facilidad de uso

El requerimiento no funcional 003 menciona que la aplicación debe garantizar la seguridad de los datos almacenados. Esto quiere garantizar que la información de los usuarios y reportes esté protegida mediante cifrado de contraseñas y conexiones seguras con la base de datos, evitando accesos no autorizados y asegurando la integridad de los datos.

|  |  |
| --- | --- |
| Identificación de requerimiento | RNF-003 |
| Nombre del requerimiento | *Protección de datos* |
| Descripción del requerimiento | La aplicación debe garantizar la seguridad de los datos mediante el uso de cifrado para contraseñas y conexiones seguras con Mongo DB. |
| Prioridad de requerimiento | Alta |

Tabla 7. Requerimiento no funcional RNF-003 Protección de Datos

El requerimiento no funcional 004 menciona que la aplicación debe ser escalable para permitir la incorporación de más laboratorios en el futuro. Esto quiere garantizar que el sistema pueda crecer sin comprometer su rendimiento, permitiendo la adición de nuevos laboratorios, usuarios y reportes sin afectar la velocidad ni la estabilidad de la plataforma.

|  |  |
| --- | --- |
| Identificación de requerimiento | RNF-004 |
| Nombre del requerimiento | *Escalabilidad* |
| Descripción del requerimiento | La aplicación debe ser escalable para soportar la incorporación de más de dos laboratorios sin afectar el rendimiento ni la usabilidad. |
| Prioridad de requerimiento | Media |

Tabla 8. Requerimiento no funcional RNF-004 Escalabilidad

3.2. Método de la propuesta

En este capítulo se explica el desarrollo del proyecto donde se incluyen todas las etapas o fases de desarrollo, mostrando así de forma gráfica y escrita el diseño de cada una de estas etapas, así como todo lo que se está desarrollando de la aplicación web para ser empleado en el área que se quiere mejorar. En la figura 1 se resume la descripción de cada etapa realizada de esta propuesta.



Figura 1. Resumen de las etapas de la propuesta

Dentro del análisis, se realiza la recolección de datos necesarios para definir con claridad los requerimientos del proyecto. Es un paso esencial, ya que permite identificar necesidades, limitaciones y oportunidades que influirán en el éxito del desarrollo.

La planeación, donde se establece un plan detallado que define las acciones específicas y los recursos necesarios para la ejecución del proyecto. Aquí se diseñan estrategias, se asignan roles y se establecen cronogramas que permitirán organizar el flujo de trabajo de manera eficiente.

La fase de desarrollo abarca la preparación del entorno de trabajo necesario para la ejecución del proyecto. Aquí se afinan detalles técnicos, se optimizan configuraciones y se establecen los elementos fundamentales que hacen que la aplicación funcione adecuadamente.

En laimplementación, se lleva a cabo el trabajo planificado previamente, asegurando que los componentes de la aplicación sean configurados y probados adecuadamente para su correcto funcionamiento.

3.2.1. Análisis de la propuesta

El objetivo de esta etapa en el desarrollo de la aplicación web para la gestión de reportes en el departamento es identificar y comprender a fondo la problemática actual para ofrecer una solución eficiente. Esto implica un análisis detallado de los requerimientos y expectativas de los usuarios finales.

Las actividades realizadas en esta fase incluyen la recolección de datos, el análisis de la viabilidad del proyecto, la identificación de riesgos y la definición de objetivos específicos. Para comprender mejor la situación y asegurar que el sistema cumpla con las necesidades de la empresa, se realizaron entrevistas y cuestionarios dirigidos al personal, con el fin de recopilar información sobre las dificultades actuales en el seguimiento de reportes y las mejoras esperadas con la implementación del sistema.

3.2.2. Planeación de la propuesta

En esta fase, se establece un plan detallado que define las acciones y los recursos necesarios para el desarrollo de la aplicación web de gestión de reportes en el departamento de Supply Chain. La planeación se enfoca en coordinar las actividades, asignar tiempos adecuados y distribuir recursos para cada etapa del desarrollo, asegurando que la propuesta se ejecute de manera eficiente y cumpla con los objetivos establecidos.

Para estructurar la planificación del proyecto, se elabora un diagrama de Gantt, donde se identifican las principales tareas y sus plazos de ejecución. Estas actividades se desarrollarán en diferentes fases, permitiendo un flujo de trabajo ordenado y alineado con los requerimientos del proyecto.

**Análisis de proyectos similares**

Esta tarea se lleva a cabo en la etapa inicial del cronograma. Su propósito es investigar y comparar soluciones existentes para la gestión de reportes en entornos empresariales. A través de este análisis, se identifican las mejores prácticas y funcionalidades clave que puedan ser adaptadas a las necesidades específicas de la empresa.

**Definición de requerimientos y estructura del sistema**

Durante esta fase, se recopilan los requerimientos funcionales y no funcionales del sistema mediante entrevistas y encuestas a los empleados del departamento. Se establecen los módulos principales de la aplicación, incluyendo la generación de reportes, la gestión de estados y el acceso al historial de incidencias.

**Diseño e implementación del entorno de desarrollo**

Esta tarea abarca la instalación y configuración del entorno necesario para la programación del sistema. Se seleccionan las tecnologías adecuadas para el desarrollo, considerando la integración con la base de datos de la empresa y la seguridad de la información.

**Fase de desarrollo**

En esta etapa, se implementan las funcionalidades clave de la aplicación web. Se desarrollarán las interfaces para los distintos roles de usuario (empleados y administradores), asegurando una navegación intuitiva y una gestión eficiente de reportes.

**Pruebas y revisión final**

Antes del despliegue final, se realizan pruebas funcionales y de seguridad para garantizar que el sistema opere sin fallos y cumpla con las expectativas de los usuarios. Se corrigen errores y se realizan ajustes con base en la retroalimentación del personal involucrado.

Con esta planificación estructurada, el proyecto avanza de manera eficiente, asegurando que la aplicación web cumpla con su propósito de automatizar la creación de reportes en el departamento.

3.2.3. Implementación de la propuesta

En esta fase, se ejecutan las actividades necesarias para la implementación de la aplicación web de gestión de reportes en el departamento. La implementación sigue el cronograma establecido, cumpliendo con los requerimientos técnicos y funcionales definidos previamente.

Se inicia con la configuración e instalación del sistema en un entorno de desarrollo adecuado. Se establece la conexión con la base de datos y se implementan los módulos principales, como la gestión de usuarios, la creación y seguimiento de reportes y el historial de incidencias.

Posteriormente, se procede con la integración de usuarios y funcionalidades, definiendo los roles de acceso (empleados y administradores) y asignando los permisos correspondientes. Se implementan funcionalidades clave, como la notificación automática de reportes y la actualización de estados de incidencias en tiempo real.

Para validar el correcto funcionamiento del sistema, se realizan pruebas funcionales, asegurando la captura, almacenamiento y visualización adecuada de los reportes. Además, se verifica la correcta emisión de notificaciones y la actualización de los estados de los reportes según el flujo de trabajo definido. Cualquier error detectado se corrige en esta etapa.

Finalmente, se lleva a cabo el despliegue de la aplicación en el entorno real, asegurando su correcta operación mediante un monitoreo inicial. Durante este período, se recopila retroalimentación de los usuarios para realizar ajustes y optimizar la plataforma.

3.2.4. Desarrollo de la propuesta

El desarrollo de la aplicación web para la gestión de reportes en los laboratorios de cómputo de la institución se lleva a cabo mediante tecnologías como HTML, CSS y JavaScript para el frontend, y Express.js para el backend. Esta fase se enfoca en la implementación de las funcionalidades necesarias para asegurar un sistema eficiente, accesible y seguro.

Preparación del Entorno de Desarrollo

Para garantizar un desarrollo estructurado y eficiente, se realiza la configuración del entorno de trabajo con las siguientes herramientas y tecnologías:

* Configuración del servidor: Se utiliza Node.js con el framework Express.js para manejar las solicitudes y respuestas del sistema.
* Gestión de base de datos: Se emplea SQL Server para el almacenamiento y administración de los reportes.
* Conexión con la base de datos: Se establecen los parámetros de conexión mediante el uso de SQL Server, asegurando la comunicación eficiente entre el backend y la base de datos.
* Definición de roles y permisos: Se implementa un sistema de autenticación y autorización que permite gestionar los accesos según los perfiles de usuario (administradores y empleados).

Desarrollo del Sistema

La construcción del sistema se divide en frontend, backend y gestión de base de datos, asegurando la integración adecuada de los componentes. Se implementa una interfaz web utilizando HTML, CSS y JavaScript, enfocándose en un diseño limpio y accesible.

* Estructura y navegación: Se crea una barra de navegación con enlaces a las secciones principales: "Inicio", "Reportes", e "Historial".
* Estilos y usabilidad: Se diseña la interfaz utilizando CSS puro, aplicando una combinación de colores que facilite la lectura y la interacción con la plataforma.
* Visualización de reportes: Se implementan tarjetas (cards) para mostrar los reportes en un formato visual organizado. Cada tarjeta incluye usuario que lo realizó además de la fecha de creación.
* Mensajes y notificaciones: Se integran alertas visuales y mensajes emergentes (modals) para mejorar la experiencia del usuario.

Backend (Lógica de Negocio)

El servidor, construido con Express.js, maneja las solicitudes de la aplicación y la comunicación con la base de datos.

* Creación de endpoints: Se implementan rutas en Express.js para gestionar las operaciones CRUD (Create, Read, Update, Delete).
* Validación de datos: Se verifica la información ingresada por los usuarios antes de almacenarla en la base de datos.
* Gestión de reportes: Se crean controladores para manejar las solicitudes de creación, actualización y consulta de reportes.

Gestión de Base de Datos

La base de datos en SQL Server almacena la información de los reportes, los usuarios y el historial de incidencias.

* Estructura de la base de datos: Se diseñan tablas para reportes y usuarios, estableciendo relaciones entre ellas.
* Optimización de consultas: Se crean procedimientos almacenados para mejorar la eficiencia en la gestión de reportes.

Pruebas y Validación

Antes de la implementación final, se realizan pruebas para garantizar el correcto funcionamiento del sistema:

* Pruebas funcionales: Se verifica que todas las funciones (registro, consulta y actualización de reportes) operen correctamente.
* Pruebas de seguridad: Se comprueba que los accesos estén protegidos y que los datos sean almacenados de manera segura.
* Pruebas de carga: Se simulan múltiples solicitudes simultáneas para evaluar la capacidad del sistema.
* Corrección de errores: Se depuran y solucionan posibles fallos detectados en las pruebas.

Documentación y Ajustes Finales

Se elabora una documentación técnica que incluye:

* Descripción de la arquitectura del sistema.
* Instrucciones para la instalación y configuración.
* Manual de usuario para la correcta utilización de la plataforma.

Se realizan los ajustes finales con base en la retroalimentación obtenida, asegurando que la aplicación esté lista para su despliegue en la empresa. Este sistema mejora significativamente la gestión de reportes en el departamento, automatizando el proceso de creación de reportes, priorizando tareas de mayor interés para el equipo corporativo de finanzas.

3.2.5. Pruebas

Las pruebas son una etapa fundamental en el desarrollo de la aplicación, ya que permiten evaluar su correcto funcionamiento, identificar errores y garantizar que cumpla con los requerimientos establecidos. Durante este proceso, se llevan a cabo diferentes tipos de pruebas para validar tanto el desempeño del sistema como la experiencia del usuario final.

En primer lugar, se realizan pruebas unitarias, donde se verifica que cada módulo o componente de la aplicación funcione de manera individual y sin errores. Estas pruebas aseguran que las funciones, consultas y elementos visuales operen según lo esperado antes de integrarlos en el sistema completo.

Posteriormente, se ejecutan pruebas de integración, en las cuales se analiza la comunicación entre los distintos módulos de la aplicación. En esta fase se valida que el frontend y el backend se comuniquen correctamente a través de las solicitudes AJAX y que la base de datos en SQL Server almacene y recupere la información sin inconsistencias.

Las pruebas de usabilidad se centran en la experiencia del usuario, evaluando que la interfaz sea intuitiva y fácil de navegar. Se recopila retroalimentación de usuarios para identificar mejoras en el diseño, accesibilidad y funcionalidad.

Además, se aplican pruebas de rendimiento, en las cuales se examina la capacidad de la aplicación para manejar múltiples usuarios simultáneamente, así como la rapidez en la ejecución de consultas y respuestas. Esto es clave para garantizar que el sistema funcione de manera eficiente bajo diferentes niveles de carga.

Finalmente, antes de la implementación final, se llevan a cabo pruebas de aceptación, donde los usuarios finales evalúan la aplicación en un entorno real. En esta fase se confirma que la plataforma cumple con los requisitos y expectativas planteadas, permitiendo realizar ajustes finales antes de su despliegue oficial.

3.2.6. Cierre

El cierre del proyecto marca la conclusión del desarrollo e implementación de la aplicación, asegurando que se cumplen todos los requerimientos y objetivos establecidos. En esta etapa, se realiza una revisión final del sistema para verificar que todas las funcionalidades operan correctamente y que la aplicación es estable, segura y eficiente.

Se elabora un informe detallado que documenta cada fase del proyecto, donde se incluyen el análisis, diseño, desarrollo, pruebas y validaciones realizadas. Este informe también contempla las mejoras implementadas con base en la retroalimentación de los usuarios, así como las decisiones tomadas a lo largo del desarrollo.

Además, se capacita a los administradores y usuarios clave para garantizar que comprenden el uso de la aplicación y pueden operar con ella de manera efectiva. En este proceso, se entregan manuales de usuario y documentación técnica para facilitar futuras actualizaciones o mantenimiento del sistema.

Finalmente, se formaliza la entrega del proyecto, asegurando que todos los involucrados están conformes con el producto final. Se establecen mecanismos de mantenimiento para atender posibles incidencias y garantizar la sostenibilidad del sistema a largo plazo. Con esto, la aplicación queda oficialmente implementada y lista para su uso en el departamento de la empresa.

3.3. Procedimiento

En este capítulo, se documenta el procedimiento llevado a cabo para el desarrollo de la aplicación web donde se muestra el proceso que abarca desde la fase inicial del análisis de proceso de la creación de los reportes, el equipo de cómputo y la implementación de la aplicación dentro de la institución. A continuación, se estructura en varias etapas principales, desde la recopilación de los requisitos, el análisis del proceso, el desarrollo de la interfaz de usuario y la implementación de las funcionalidades principales de la aplicación web.

3.3.1. Análisis

El análisis del proyecto comienza con una recolección de datos que permite comprender la problemática existente y definir los requerimientos clave para el desarrollo de la aplicación web. En este caso, la principal problemática detectada es el tiempo que se emplea para crear ciertos reportes que el equipo corporativo requiere, ya sea para tomar acciones referentes a órdenes de compra, o para las acciones a tomar a la hora de analizar el reporte de cancelaciones de órdenes de compra y expedites.

Actualmente, los reportes suelen registrarse de manera manual, mediante hojas de cálculo de Microsoft Excel, lo cual es un proceso repetitivo, además de que requiere de mucho cálculo manual, pues la información no es constante y suele cambiar muy a menudo, lo que implica que el empleado tenga que llevar a cabo procesos diferentes en cuanto a uso de fórmulas se requiere.

Al presentarse este problema, el uso de macros en Excel se vuelve un tema complicado, puesto que las macros no funcionan de la misma forma con los reportes cada semana, debido a la información que puede ser mayor o menor, y esto puede ocasionar errores de cálculo y provocar un margen de error mayor o igual al proceso manual por parte del empleado.

Para obtener un panorama detallado de la situación, se lleva a cabo un proceso de recopilación de información basado en entrevistas con empleados del departamento. Este proceso permite conocer sus necesidades, dificultades y expectativas con respecto a la nueva aplicación.

Se identifican los principales problemas en la gestión de reportes, como la falta de un registro controlado, la ausencia de un mecanismo de seguimiento y la imposibilidad de consultar fácilmente reportes anteriores para analizar tendencias o determinar patrones de fallas recurrentes.

Además, se documentan los procedimientos actuales para evaluar cómo pueden ser optimizados a través de la automatización y digitalización de los procesos. Tras la recopilación de información, se realiza un análisis de viabilidad técnica para determinar si es posible desarrollar una aplicación que cumpla con los requisitos establecidos.

También se examinan las herramientas y tecnologías de desarrollo más adecuadas para garantizar el correcto funcionamiento de la aplicación, priorizando soluciones que ofrezcan escalabilidad, seguridad y facilidad de uso.

Se considera la integración con bases de datos en SQL Server, asegurando que la información pueda almacenarse de manera estructurada y que los datos sean accesibles de forma rápida y confiable. El análisis de riesgos es otro aspecto crucial en esta etapa, ya que permite identificar y anticipar posibles problemas que podrían surgir durante el desarrollo e implementación de la aplicación.

Se reconocen riesgos como la resistencia al cambio por parte de los empleados, la posibilidad de fallos en la conectividad a internet que dificulten el acceso a la aplicación y la necesidad de capacitar a los usuarios para garantizar una adopción efectiva. También se contempla la posibilidad de que la aplicación no cubra inicialmente todas las necesidades operativas, lo que requerirá un proceso iterativo de mejoras y ajustes.

Para mitigar estos riesgos, se propone la realización de pruebas piloto y sesiones de capacitación previas al despliegue de la aplicación. Con base en el análisis realizado, se establecen los objetivos del proyecto, así como sus requerimientos funcionales y no funcionales.

Los objetivos incluyen la implementación de un sistema de reportes digitalizado que permita a los empleados registrar reportes de manera rápida y sencilla, asegurando que cada reporte cuente con un historial de seguimiento y que los administradores puedan gestionar estos de manera eficiente.

Entre los requerimientos funcionales destacan la creación de un formulario de reportes estructurado, la posibilidad de adjuntar documentos, la integración de filtros de búsqueda para facilitar la consulta de reportes previos y el historial de reportes creados para mantener un control en la documentación.

Los requerimientos no funcionales se enfocan en garantizar la seguridad y estabilidad del sistema. Se define la necesidad de implementar un sistema de autenticación que restrinja el acceso a usuarios autorizados, así como la optimización del rendimiento de la base de datos para manejar eficientemente grandes volúmenes de información sin afectar la velocidad de consulta.

Además, se establece que la interfaz de usuario debe ser intuitiva y accesible desde equipos de escritorio y laptops, asegurando que los empleados puedan interactuar con la aplicación sin necesidad de conocimientos técnicos avanzados.

Finalmente, con el análisis completo de viabilidad, riesgos y requerimientos, se decide adoptar una estrategia de desarrollo basada en tecnologías ampliamente utilizadas en aplicaciones web, asegurando compatibilidad con el entorno empresarial y flexibilidad para futuras mejoras.

Se opta por una arquitectura cliente-servidor en la que el backend gestionará las operaciones de la base de datos en SQL Server y el frontend ofrecerá una experiencia de usuario fluida e intuitiva mediante HTML, CSS y JavaScript con jQuery.

3.3.2. Planeación

El desarrollo del proyecto es un punto importante por visualizar, sobre todo, de monitorear, para ello es ideal utilizar un diagrama de Gantt, ya que cada actividad puede ser organizada dentro de un margen de tiempo establecido. Este recurso facilita enormemente la identificación de las dependencias entre tareas y permite prever posibles retrasos.

Cada tarea asignada se refleja en el cronograma, lo que permite que el desarrollador decida acertadamente cómo administrar el tiempo para poder completarlas con respecto al tiempo y forma, así mismo para garantizar un adecuado seguimiento del cronograma, se establece una revisión periódica del progreso, ajustando tiempos y prioridades en función del avance real de cada tarea.

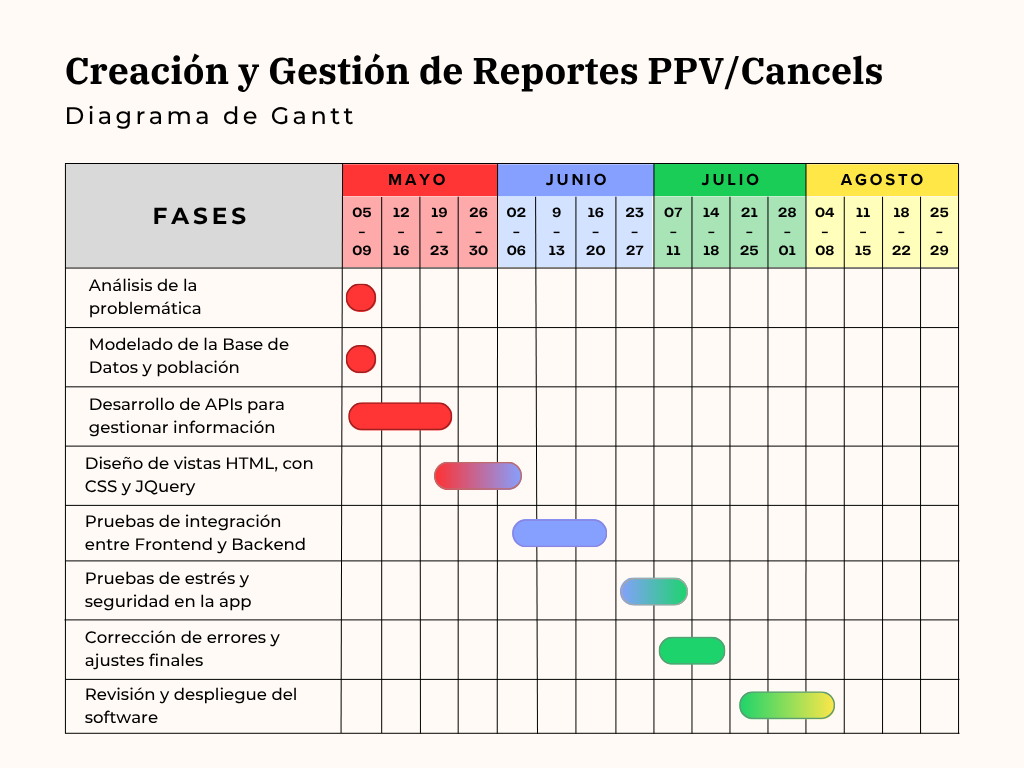


Figura 2. Cronograma del proyecto.

El cronograma contempla etapas fundamentales que incluyen el análisis y recolección de datos, donde se identifican los requerimientos, se definen los objetivos y se evalúa la viabilidad técnica del proyecto. Posteriormente, el diseño del sistema abarca la creación de la estructura de la base de datos, la arquitectura de la aplicación y el diseño de la interfaz de usuario, estableciendo las bases para el desarrollo posterior.

El diagrama de Gantt se estructura en ocho fases principales: Análisis de la problemática, Diseño del Modelado de la Base de Datos y población, Desarrollo de APIs para gestionar datos, Diseño de vistas HTML, con CSS y JQuery, Integración y pruebas, Corrección de errores y Revisión. Cada fase contiene tareas específicas con una duración estimada, marcando dependencias entre ellas.

Dado que se trata de un proyecto individual, la comunicación y gestión de tareas se centraliza en herramientas de productividad personal. Se utilizan aplicaciones como Trello para la organización de actividades y el control de avances, permitiendo una visualización clara de las tareas pendientes, en proceso y completadas.

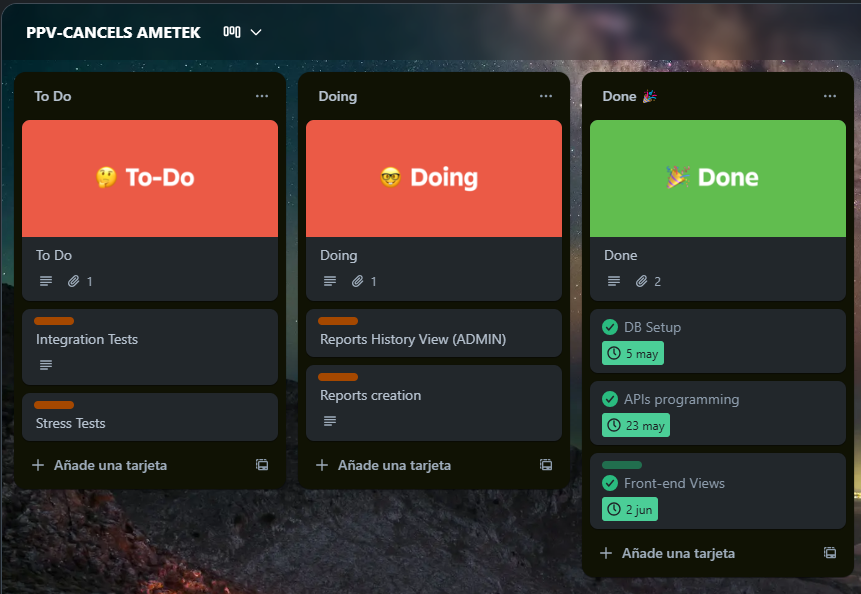


Figura 3. Gestión de tareas en Trello.

Además, se establece un sistema de control de versiones mediante Git y GitHub, lo que permite realizar un seguimiento de los cambios en el código y revertir modificaciones en caso de ser necesario. Esta estrategia asegura la estabilidad del desarrollo y facilita la implementación de mejoras a lo largo del proyecto.

El seguimiento constante del cronograma y la revisión periódica de los avances garantizan que el desarrollo de la aplicación se mantenga alineado con los objetivos establecidos y dentro del tiempo planeado.

3.3.3. Desarrollo

El desarrollo de la aplicación web se lleva a cabo siguiendo una metodología estructurada que permite construir un sistema funcional, eficiente y escalable. Esta fase involucra la implementación del backend, frontend y base de datos, asegurando la integración de cada componente y la optimización de su rendimiento.

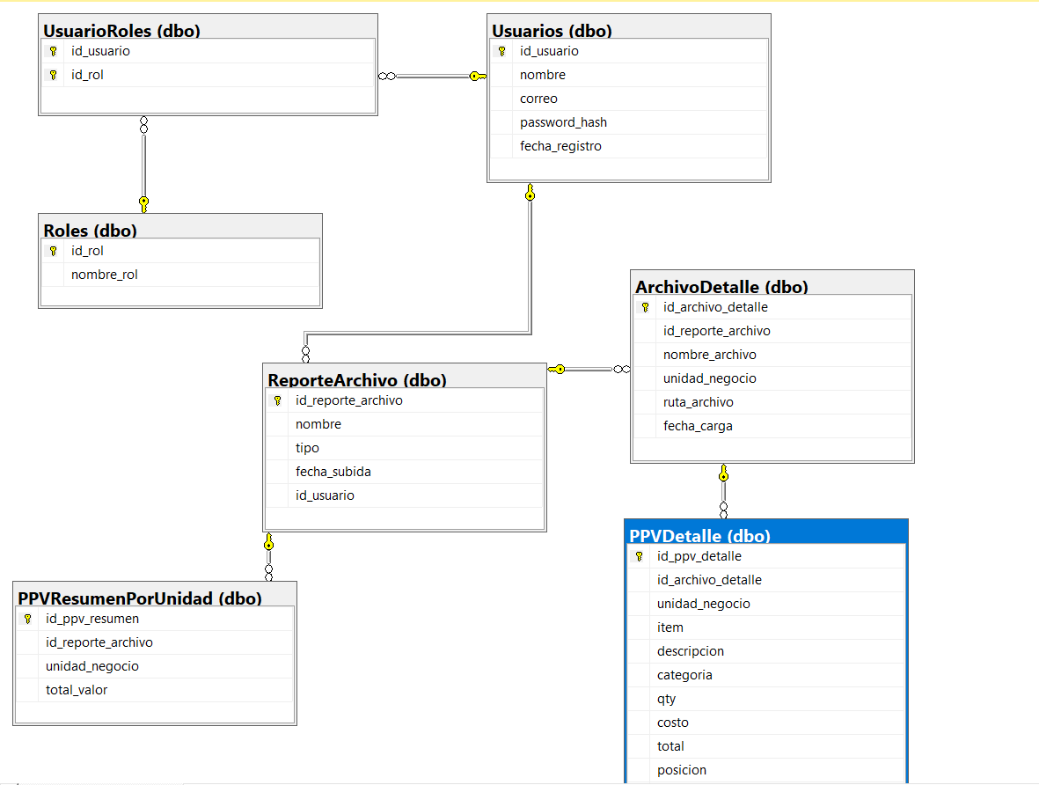


Figura 4. Diagrama ER del proyecto.

El proceso inicia con la configuración de la base de datos en SQL Server, donde se crean las tablas necesarias para almacenar la información de los usuarios, reportes, laboratorios y computadoras. Se definen relaciones entre las tablas para garantizar la integridad de los datos y se establecen claves primarias y foráneas que permiten gestionar los reportes de manera estructurada. Se implementan índices y optimizaciones en las consultas para mejorar la velocidad de acceso a la información.

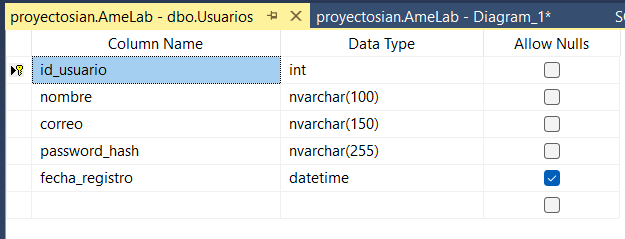


Figura 5. Usuarios en BD.

Aquí se muestra la tabla dbo.Usuarios dentro de la base de datos AmeLab, alojada en un servidor SQL Server. Esta tabla constituye uno de los elementos fundamentales del modelo de datos, ya que almacena la información de los usuarios registrados en el sistema, quienes son responsables de cargar, consultar y administrar los distintos reportes dentro de la plataforma.

La estructura de la tabla incluye campos como id\_usuario, que funciona como clave primaria autoincremental para identificar de forma única a cada usuario, así como nombre, correo y password\_hash, los cuales permiten autenticar al usuario de forma segura mediante el uso de contraseñas cifradas. También se incluye el campo fecha\_registro, que registra automáticamente el momento en que se dio de alta el usuario en la base de datos.

Esta tabla se encuentra directamente relacionada con otras entidades del sistema, como ReporteArchivo, permitiendo trazar qué usuario ha generado o cargado determinado reporte. Además, se relaciona con la tabla UsuarioRoles para definir los privilegios y permisos que tiene cada usuario según su rol (por ejemplo, administrador, analista, o usuario general).

A screenshot of a computer

AI-generated content may be incorrect.

Figura 6. Detalles de archivo en BD.

Se muestra en la figura 6 la estructura de la tabla dbo.ArchivoDetalle dentro de la base de datos AmeLab, alojada en un servidor SQL Server (instancia proyectosian.database.windows.net). En el panel izquierdo del Explorador de Objetos de SSMS (SQL Server Management Studio), se visualiza la jerarquía completa de la base de datos, en donde esta tabla forma parte del esquema dbo junto con otras entidades relacionadas al módulo de reportes.

La tabla ArchivoDetalle almacena información detallada de cada uno de los archivos individuales que forman parte de un reporte completo. Contiene campos como id\_archivo\_detalle, clave primaria que identifica cada archivo de manera única, y id\_reporte\_archivo, que actúa como clave foránea enlazando con la tabla ReporteArchivo, permitiendo establecer una relación directa entre el archivo y el reporte al que pertenece.

Además, se incluyen columnas como nombre\_archivo, unidad\_negocio y ruta\_archivo, que registran datos importantes sobre el origen y clasificación del archivo, así como su ubicación en el sistema. El campo fecha\_carga guarda la marca de tiempo del momento en que el archivo fue incorporado a la base de datos.

Esta tabla resulta fundamental para estructurar los datos provenientes de múltiples unidades de negocio, ya que cada archivo cargado puede representar la información de una unidad específica. A través de esta organización, el sistema permite no solo almacenar y vincular los archivos correctamente, sino también garantizar su trazabilidad y facilitar el procesamiento de datos durante el análisis de los reportes.

A screenshot of a computer

AI-generated content may be incorrect.

Figura 7.PPV Detalle en BD.

Se muestra la estructura de la tabla dbo.PPVDetalle dentro de la base de datos AmeLab, gestionada en un servidor SQL Server y visualizada a través de SQL Server Management Studio (SSMS). Esta tabla desempeña un papel clave dentro del modelo de datos, ya que su propósito principal es almacenar de forma estructurada cada una de las filas extraídas desde los archivos .xlsx que los usuarios cargan al sistema, específicamente relacionados con reportes de tipo PPV (Purchase Price Variance).

La tabla contiene columnas que permiten capturar información detallada como el nombre del ítem (item), su descripción, la categoría a la que pertenece, la unidad de negocio asociada, la cantidad (qty), el costo unitario y el total calculado. Cada fila también está vinculada a un archivo específico mediante el campo id\_archivo\_detalle, que actúa como clave foránea hacia la tabla ArchivoDetalle.

Además, PPVDetalle puede almacenar datos adicionales para análisis como el campo posición, que permite identificar si un valor pertenece a los 10 más altos o bajos según su impacto, y tipo\_valor, que clasifica el valor como positivo o negativo.

A screenshot of a computer

AI-generated content may be incorrect.

Figura 8. PPV por unidad en BD.

En la figura 8 se muestra la estructura de la tabla dbo.PPVResumenPorUnidad dentro de la base de datos AmeLab, alojada en un servidor SQL Server. Esta tabla almacena los resultados consolidados por unidad de negocio, derivados del análisis de los reportes PPV. Contiene campos como id\_reporte\_archivo, unidad\_negocio y total\_valor, permitiendo consultar indicadores clave sin recalcular desde los datos detallados. Su uso mejora el rendimiento y facilita la generación de reportes ejecutivos.

A screenshot of a computer

AI-generated content may be incorrect.

Figura 9. Reporte archivo en BD.

Se muestra la estructura de la tabla dbo.ReporteArchivo dentro de la base de datos AmeLab, alojada en un servidor SQL Server y administrada a través de SQL Server Management Studio (SSMS). Esta tabla forma parte fundamental del modelo relacional diseñado para gestionar el registro y control de reportes generados por los usuarios dentro del sistema.

ReporteArchivo contiene campos como id\_reporte\_archivo (clave primaria), nombre, tipo, fecha\_subida y id\_usuario. Estos campos permiten identificar el nombre del reporte, clasificarlo según su tipo (por ejemplo, PPV o Cancels), registrar la fecha en que fue cargado y asociarlo con el usuario responsable mediante una clave foránea.

Desde esta tabla se establecen relaciones con otras entidades del sistema, como ArchivoDetalle, que almacena los archivos individuales relacionados a un reporte, y PPVResumenPorUnidad, que consolida los resultados analíticos por unidad de negocio. Esta estructura facilita la trazabilidad completa desde la carga inicial hasta el análisis final.

A screenshot of a computer

AI-generated content may be incorrect.

Figura 10. Roles en BD.

Se muestra en la figura 10 la estructura de la tabla dbo.Roles dentro de la base de datos AmeLab, visualizada mediante SQL Server Management Studio (SSMS). Esta tabla forma parte del esquema de autenticación y control de acceso del sistema, permitiendo definir los distintos tipos de roles disponibles para los usuarios.

Roles contiene dos columnas: id\_rol, que funciona como clave primaria autoincremental, y nombre\_rol, que almacena el nombre descriptivo del rol (por ejemplo, "admin", "usuario" o "analista"). Estos datos permiten clasificar y gestionar los permisos según el perfil asignado.

La tabla se vincula con UsuarioRoles para establecer relaciones muchos a muchos entre usuarios y roles. Gracias a esta estructura, el sistema puede asignar múltiples roles a un mismo usuario, garantizando una gestión de permisos flexible y escalable.

A screenshot of a computer

AI-generated content may be incorrect.

Figura 11. Roles de usuario en BD.

Se muestra en la figura 10 la estructura de la tabla dbo.Roles dentro de la base de datos AmeLab, visualizada mediante SQL Server Management Studio (SSMS). Esta tabla forma parte del esquema de autenticación y control de acceso del sistema, permitiendo definir los distintos tipos de roles disponibles para los usuarios registrados.

La tabla Roles contiene dos columnas principales: id\_rol, que funciona como clave primaria autoincremental, y nombre\_rol, que almacena el nombre descriptivo del rol asignado (por ejemplo, "admin", "usuario" o "analista"). Estos campos permiten clasificar y gestionar los permisos de cada usuario según el perfil que se le asigne dentro del sistema.

Además, esta tabla se vincula con UsuarioRoles, que actúa como tabla intermedia para establecer una relación muchos a muchos entre los usuarios y los roles. Gracias a esta estructura, el sistema permite asignar múltiples roles a un mismo usuario, lo que facilita una gestión de accesos más flexible, escalable y adaptable a distintos escenarios de uso.

A screenshot of a computer program

AI-generated content may be incorrect.

Figura 12. Servidor con Node.Js.

Posteriormente, se desarrolla el backend utilizando Node.js con Express, el cual se encarga de gestionar las solicitudes de los usuarios, procesar los datos y comunicarse con la base de datos. Se crean las rutas y controladores que permiten registrar reportes, consultar el historial de incidencias, gestionar usuarios y actualizar el estado de cada reporte.

A computer screen shot of a program

AI-generated content may be incorrect.

Figura 13. Código HTML del Login.

En el código mostrado en la figura 13 se construye la interfaz de la vista Login, utilizando HTML para estructurar la información. Se define un contenedor principal donde se coloca un título dentro de un contenedor (container) y los campos correspondientes a un formulario en el que se agregan el correo del usuario y su contraseña, terminando con un botón de entrar al final del formulario.

A screenshot of a computer

AI-generated content may be incorrect.

Figura 14. Seguimiento de reportes.

En esta vista del sistema web de Ametek, se presenta el módulo dedicado al seguimiento de reportes. Este módulo tiene como objetivo central brindar una visión clara del estado actual de los reportes cargados por los usuarios a través de la plataforma. El diseño está centrado en facilitar la supervisión operativa de los archivos subidos y su procesamiento interno.

En la parte superior se muestra un título claro que indica el nombre del módulo, seguido de cuatro tarjetas de resumen. Estas tarjetas indican la cantidad total de reportes en distintos estados: "Reportes Procesados", "En Proceso", "Con Errores" y "Total Cargados". Cada tarjeta está diferenciada con un borde de color distinto (verde, amarillo, rojo y azul, respectivamente) para una identificación visual rápida.

Debajo del resumen, se encuentra una sección titulada "Últimos reportes cargados", que muestra una tabla. Esta tabla está estructurada para presentar información clave de cada reporte: su número correlativo, el nombre del archivo, la unidad de negocio a la que pertenece, el nombre del usuario que lo cargó, la fecha exacta de subida y el estado actual del reporte. Aunque en este momento la tabla aparece vacía, está preparada para mostrar los datos en tiempo real conforme los reportes sean procesados.

Este diseño está pensado para ser tanto visualmente limpio como funcional. Permite a los administradores o analistas verificar rápidamente el flujo de archivos, identificar cuellos de botella, y acceder a un historial reciente sin tener que navegar por múltiples vistas. En general, esta pantalla es una pieza clave para el monitoreo operativo del sistema de reportes en la empresa.

A screenshot of a computer

AI-generated content may be incorrect.

Figura 15. Historial Reportes.

La vista corresponde al módulo de Historial de Reportes, accesible desde el menú lateral izquierdo. Este módulo está diseñado para que el usuario pueda consultar todos los reportes previamente cargados en el sistema, sin importar su estado o fecha. Es una herramienta útil para auditorías, revisiones de actividades pasadas y seguimiento general.

En la parte superior central de la sección principal, se encuentran los filtros interactivos que permiten buscar reportes específicos según tres criterios: fecha de carga, usuario que lo subió y estado del reporte. Estos filtros están distribuidos en forma horizontal y usan íconos representativos para que el usuario pueda identificar rápidamente su funcionalidad.

Debajo de los filtros, está la tabla que debería mostrar los resultados según los criterios seleccionados. Actualmente, aparece un mensaje de que “No hay reportes registrados”, lo que indica que aún no se han cargado datos o que no hay coincidencias con los filtros aplicados. Esta área se actualizará dinámicamente conforme se suban reportes al sistema.

A screenshot of a computer

AI-generated content may be incorrect.

Figura 16. Prueba POST de Login con Postman.

En la imagen se observa una prueba realizada en Postman para verificar la respuesta de la API en la ruta http://localhost:3000/login. Se utiliza el método POST para dar de alta a los usuarios de la base de datos y evaluar la correcta comunicación entre el backend y el sistema de almacenamiento.

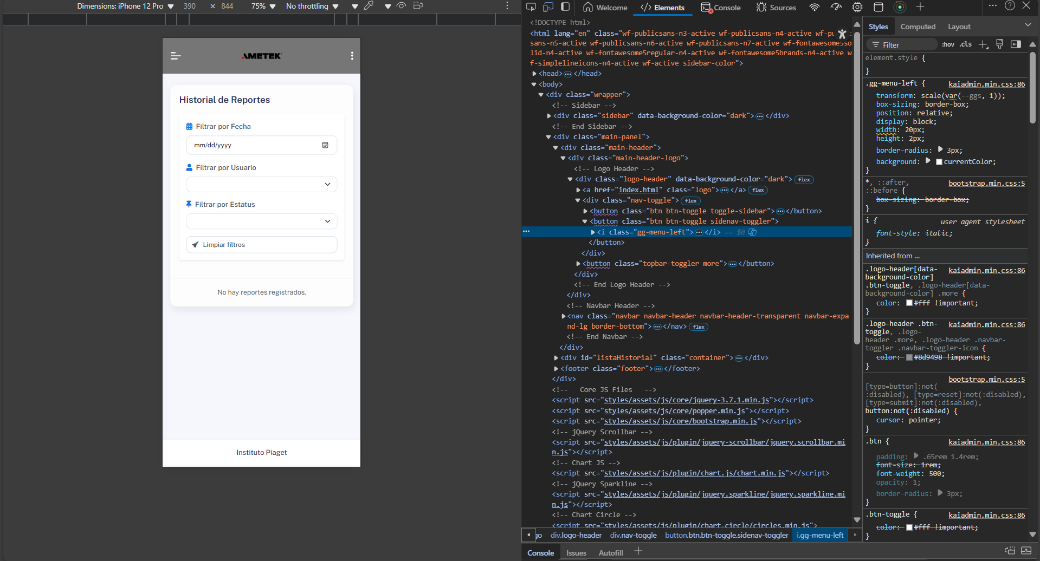


Figura 17. Integración con Bootstrap.

La integración entre el backend, frontend y la base de datos se prueba mediante el envío y recepción de datos en distintos escenarios. Se simulan registros de reportes por parte de los docentes y modificaciones de estado por parte de los administradores, verificando que los cambios se reflejen en la interfaz sin errores.

A screenshot of a computer

AI-generated content may be incorrect.

Figura 18. Prueba GET de usuarios con Postman.

Finalmente, se documentan todas las funciones y estructuras del código, facilitando futuras modificaciones y escalabilidad del sistema. Con la finalización del desarrollo, la aplicación está lista para su fase de pruebas finales e implementación en la institución.

A screenshot of a computer program

AI-generated content may be incorrect.

Figura 19. Código de la vista de usuarios.

En la figura 19 se aprecia el contenido de un archivo HTML, donde se define un formulario para la asignación de laboratorios y niveles educativos a los usuarios. Se incluyen etiquetas <label> y <input> para capturar información como el correo del usuario y la selección de laboratorio.

Además, se integran elementos <select> con múltiples opciones dentro de etiquetas <option>, permitiendo a los usuarios elegir entre diferentes niveles educativos. La estructura sugiere que esta interfaz facilita la administración de usuarios dentro del sistema, asegurando que cada usuario esté vinculado a un laboratorio y un nivel específico.

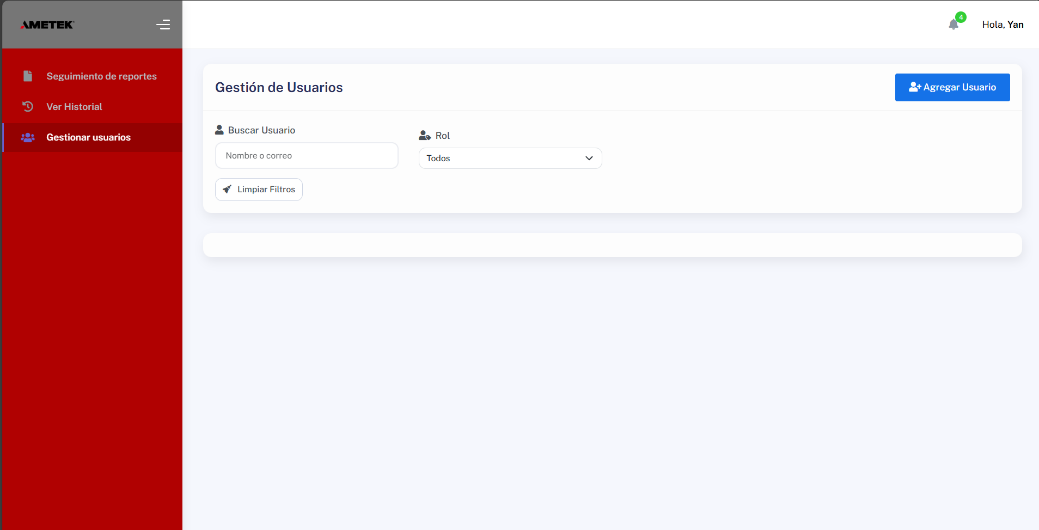


Figura 20. Vista HTML de usuarios.

En la sección principal de la pantalla, en la figura 20, se encuentra un módulo de filtros que permite refinar la búsqueda de usuarios mediante diferentes criterios, como nombre de usuario y rol. Cada filtro cuenta con un campo desplegable que permite seleccionar valores específicos, lo que facilita la localización y organización de los usuarios dentro del sistema.

Debajo de los filtros, se despliega una lista de usuarios registrados en el sistema, mostrando información relevante como el correo electrónico y el rol que desempeñan dentro de la plataforma.

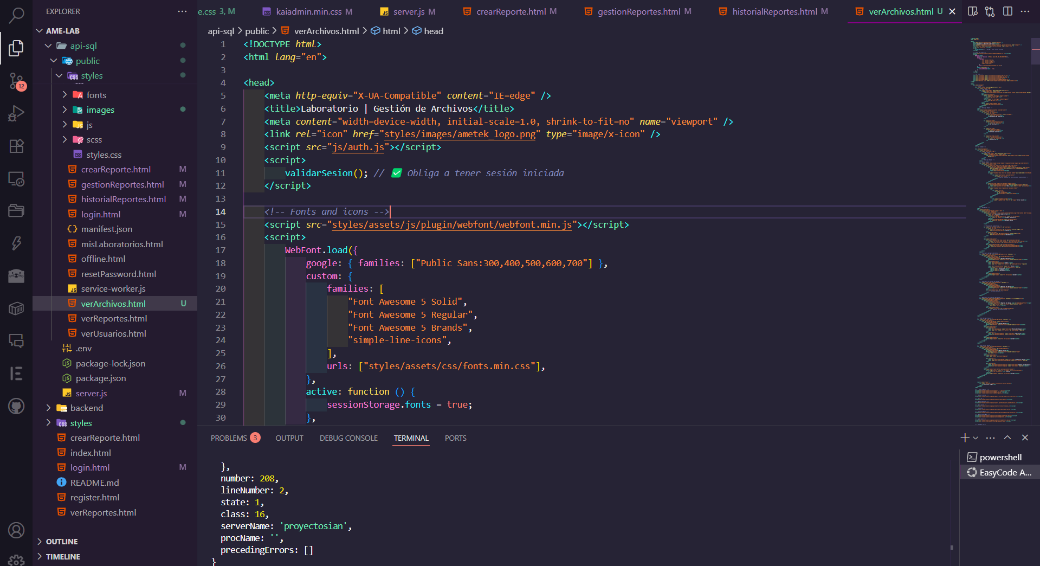


Figura 21. Código HTML de la vista de Archivos.

Se observa un formulario en HTML en la figura 21, esta contiene un campo de entrada para el nombre del laboratorio, asegurando que sea un campo obligatorio mediante el atributo required. En la parte inferior del código, se implementa un modal en Bootstrap, que se activa cuando se desea ver los detalles de un equipo dentro de un laboratorio.

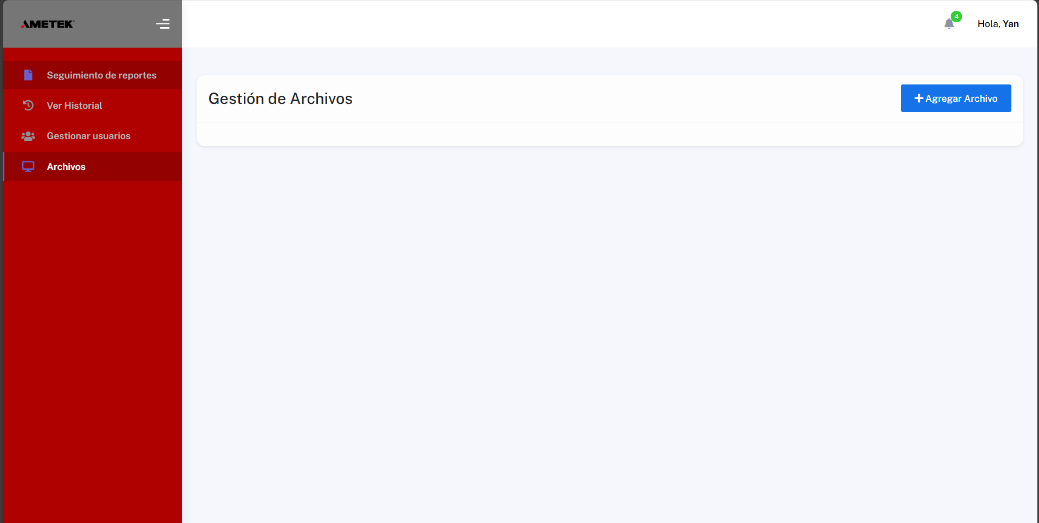


Figura 22. Vista de archivos.

Esta imagen muestra la interfaz del módulo de Gestión de Archivos dentro del sistema web de reportes de Ametek. Es una sección pensada para permitir al usuario agregar, consultar o administrar archivos que forman parte de los reportes o que serán procesados por el sistema. La sección central está encabezada con el título “Gestión de Archivos”, lo que da claridad inmediata sobre la funcionalidad de esta vista. A la derecha, se encuentra un botón de acción con el texto “+ Agregar Archivo”, cuyo propósito es el de permitir la subida de archivos Excel por los usuarios.

**3.3.4. Implementación**

Para garantizar el correcto funcionamiento del sistema dentro del instituto, es necesario seguir un proceso de implementación que abarque la infraestructura tecnológica, la instalación del software y el mantenimiento continuo. Se documenta y se muestran pruebas gráficas de cómo se lleva a cabo la implementación de la aplicación.

**Infraestructura necesaria**

Para llevar a cabo la implementación de la aplicación web, se establece una infraestructura tecnológica que asegure su estabilidad, rendimiento y accesibilidad dentro del entorno institucional. Se contempla la ejecución del backend en un servidor compatible con Node.js y Express, mientras que la gestión de datos se realiza a través de una base alojada en SQL Server, lo que permite una integración segura y eficiente con las operaciones de la aplicación. Se considera el uso de un servidor interno con dirección IP estática, aunque también se contempla la opción de migrar a un servidor en la nube, como Azure, en caso de que el cliente decida optar por una solución más robusta y escalable.

Los dispositivos desde los cuales se accederá a la aplicación, como computadoras de escritorio o laptops dentro de los laboratorios, deben contar con conectividad a la red interna del instituto. Esta condición garantiza el acceso fluido a la aplicación y permite su instalación. Una vez instalada en el navegador, la aplicación puede ejecutarse incluso en ausencia de conexión a Internet, gracias al aprovechamiento de la caché y almacenamiento local proporcionado por el navegador, asegurando así la continuidad del servicio.

Como parte de esta infraestructura, se incluye el archivo manifest.json, el cual cumple un rol fundamental en la configuració. Este archivo define elementos visuales como el nombre de la aplicación, la URL de inicio, los colores temáticos y los íconos que se utilizarán para su instalación en los dispositivos.

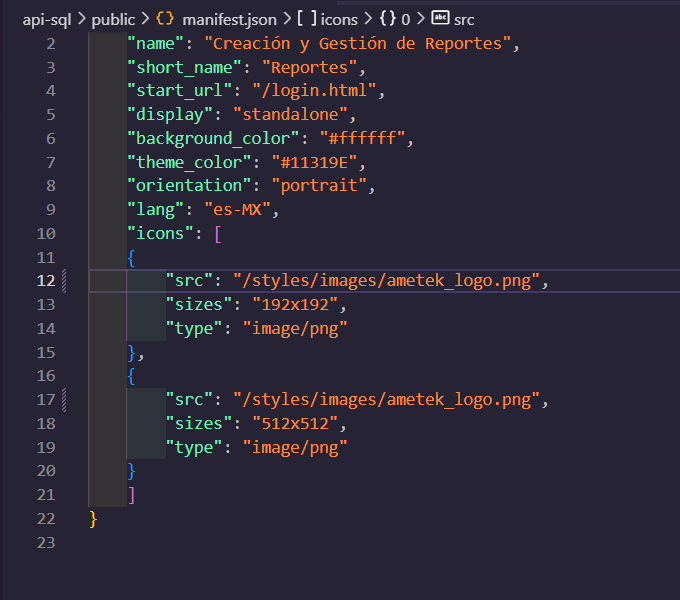


Figura 27. Manifest.json

Asimismo, se incorpora un Service Worker, un script en JavaScript que opera en segundo plano y permite que la aplicación web funcione sin conexión a internet. Para lograrlo, se crea un archivo llamado service-worker.js en la raíz del proyecto.

Durante el evento install, el Service Worker guarda en caché los recursos principales de la aplicación, como HTML, CSS, JavaScript e imágenes. Posteriormente, mediante el evento fetch, se interceptan las peticiones de red y, si los archivos solicitados ya están en la caché, se sirven desde ahí. Esto garantiza que la aplicación continúe funcionando, aunque no haya conexión a internet.



Figura 28. Service Worker

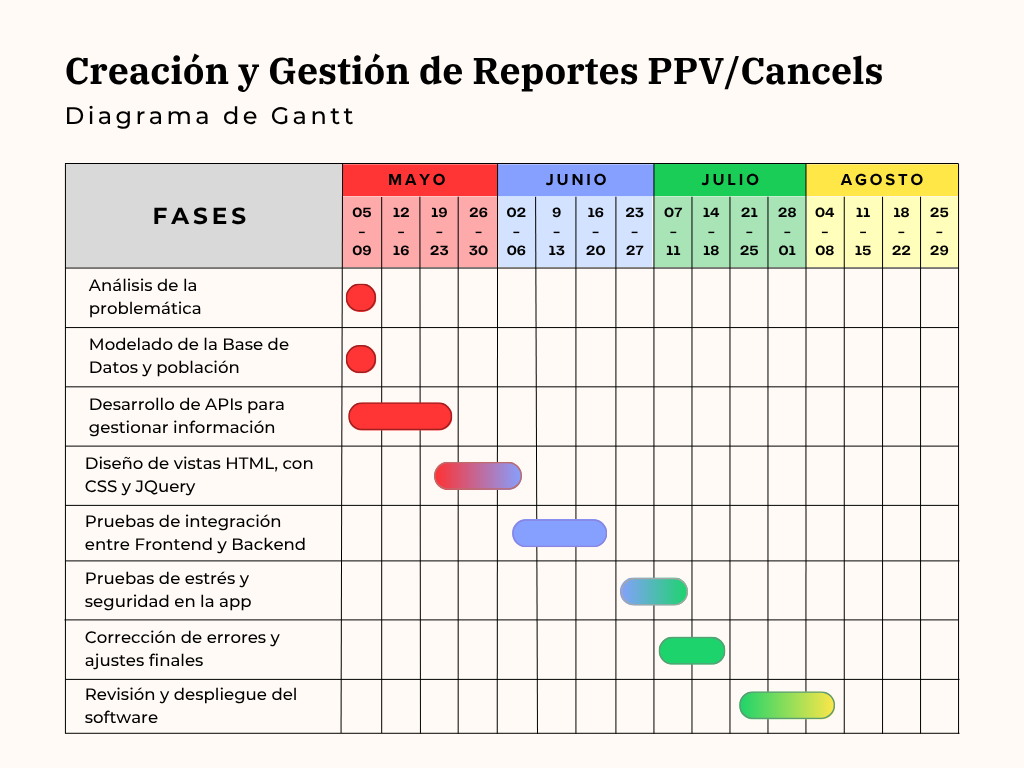
**Uso de la Aplicación Web**

La aplicación está diseñada para ser utilizada tanto en computadoras de escritorio como en dispositivos móviles. Su instalación es sencilla: los usuarios solo deben acceder al sistema mediante navegadores como Google Chrome o Microsoft Edge y seleccionar la opción "Añadir a la pantalla de inicio".

Gracias a la capacidad de funcionamiento sin conexión (offline), los usuarios pueden generar reportes aun cuando no tengan acceso a internet. Una vez restablecida la conexión, los datos se sincronizan automáticamente con el servidor, asegurando que la información no se pierda.

3.4. Control del programa de trabajo

En este punto se describe el Diagrama de Gantt del proyecto, así como su descripción de cada fase mostrada en el diagrama con el objetivo de comprender el manejo y desarrollo del proyecto como se puede apreciar en la figura



Dentro del mes de enero….

[Screenshot]

En el mes de febrero…

[Screenshot]

Durante el mes de marzo…

[Screenshot]

Concluyendo con el mes de abril…

[Screenshot]

3.5. Presupuesto

En esta sección se hace hincapié en los presupuestos para la elaboración del proyecto, entre los cuales se encuentran los recursos materiales, recursos humanos, servicios y presupuesto total, los cuales son de suma importancia para este proyecto, mismos presentados y explicados a continuación.

3.5.1. Recursos materiales

En los recursos materiales entran todos aquellos consumibles que fueron requeridos durante este proyecto, tal como los son los softwares, así como el hardware que se utiliza a lo largo de la implementación para así tener en cuenta las herramientas utilizadas y su presupuesto.

Herramientas de software

Para llevar a cabo este proyecto se hace uso de aplicaciones