**UNIVERSIDAD TECNOLÓGICA**

**EMILIANO ZAPATA DEL ESTADO DE MORELOS**

**DIVISIÓN ACADÉMICA DE TECNOLOGÍAS DE LA INFORMACIÓN Y DISEÑO**

**SISTEMA DE CONTROL DE INVENTARIO**

**REPORTE DE ESTADÍA**

**QUE PARA OBTENER EL GRADO DE:**

**TÉCNICO SUPERIOR UNIVERSITARIO EN TECNOLOGÍAS DE LA INFORMACIÓN**

**ÁREA DESARROLLO DE SOFTWARE MULTIPLATAFORMA**

|  |  |  |
| --- | --- | --- |
| **PRESENTA:**  **JOSÉ MIGUEL DELGADO PÉREZ** | | |
| ASESOR EMPRESARIAL |  | ASESOR UNIVERSITARIO |
| **ING. ALEJANDRO BAHENA BRAVO** |  | **DR. HUGO OMAR ALEJANDRES SÁNCHEZ** |

**EMILIANO ZAPATA, MOR., SEPTIEMBRE DE 2023**

|  |
| --- |
| **CONTENIDO** |

Índice de figuras

Índice de tablas

Agradecimientos

Resumen

Summary

[CAPÍTULO 1. DESCRIPCIÓN DEL PROYECTO 2](#_Toc143160168)

[1.1 Datos generales de la empresa 2](#_Toc143160169)

[1.2 Antecedentes del proyecto 2](#_Toc143160170)

[1.3 Objetivo general 2](#_Toc143160171)

[1.4 Objetivos específicos 2](#_Toc143160172)

[1.5 Justificación 2](#_Toc143160173)

[1.6 Alcances 2](#_Toc143160174)

[1.7 Restricciones 2](#_Toc143160175)

[CAPÍTULO 2. MARCO DE REFERENCIA 2](#_Toc143160176)

[2.1 Conceptos, metodologías y herramientas 2](#_Toc143160177)

[2.1.1 Conceptos de arquitectura de software 2](#_Toc143160178)

[2.1.2 Motores de base de datos 2](#_Toc143160179)

[2.1.3 Tecnologías de backend 2](#_Toc143160180)

[2.1.4 Tecnologías de frontend 2](#_Toc143160181)

[2.1.5 Control de versiones 2](#_Toc143160182)

[2.1.6 Metodologías agiles 2](#_Toc143160183)

[2.2 Propuesta de solución 2](#_Toc143160184)

[CAPÍTULO 3. DESARROLLO 2](#_Toc143160185)

[3.1 Inicio 2](#_Toc143160186)

[3.2 Planeación 2](#_Toc143160187)

[3.2.1 Documento Formal de Requerimientos 2](#_Toc143160188)

[3.2.2 Plan de comunicación 2](#_Toc143160189)

[3.2.3 EDT 2](#_Toc143160190)

[3.2.4 Cronograma de seguimiento 2](#_Toc143160191)

[3.2.5 Diagrama de Gantt 2](#_Toc143160192)

[3.3 Ejecución 2](#_Toc143160193)

[3.3.1 Representación conceptual 2](#_Toc143160194)

[3.3.2 Diagrama de casos de uso 2](#_Toc143160195)

[3.3.3 Estructura del proyecto del lado del cliente 2](#_Toc143160196)

[3.3.4 Diseño de vistas 2](#_Toc143160197)

[3.3.5 Base de datos 2](#_Toc143160198)

[3.3.6 Servidor 2](#_Toc143160199)

[3.3.7 Control de versiones 2](#_Toc143160200)

[3.4 Control 2](#_Toc143160201)

[3.5 Cierre 2](#_Toc143160202)

[CAPÍTULO 4. CONCLUSIONES 2](#_Toc143160203)

[4.1 Cumplimiento de objetivos 2](#_Toc143160204)

[4.2 Resultados 2](#_Toc143160205)

[4.3 Contribuciones 2](#_Toc143160206)

**REFERENCIAS**

**ANEXOS**

|  |
| --- |
| **ÍNDICE DE FIGURAS** |

[Figura 1.1Logotipo de la empresa 8](#_Toc143455132)

[Figura 3.1 EDT 26](#_Toc143455133)

[Figura 3.2. Diagrama de Gantt parte 1 28](#_Toc143455134)

[Figura 3.2. Diagrama de Gantt parte 2 28](#_Toc143455135)

[Figura 3.4 Representación conceptual 29](#_Toc143455136)

[Figura 3.5 Diagrama de Casos de Uso 31](#_Toc143455137)

[Figura 3.6 Estructura de carpetas 32](#_Toc143455138)

[Figura 3.7 Creación de un módulo 33](#_Toc143455139)

[Figura 3.8. Módulo de Auth 34](#_Toc143455140)

[Figura 3.9 Creación de un componente 34](#_Toc143455141)

[Figura 3.10 Componente de Auth 35](#_Toc143455142)

[Figura 3.11 Diseño de vista login 35](#_Toc143455143)

[Figura 3.12 Listado de Inventarios 36](#_Toc143455144)

[Figura 3.13 Diagrama de la base de datos 37](#_Toc143455145)

[Figura 3.14. Tabla de Inventarios 39](#_Toc143455146)

[Figura 3.15. Tabla de lotes 40](#_Toc143455147)

[Figura 3.16. Estructura del Servidor 40](#_Toc143455148)

[Figura 3.17. Modelo de Usuarios 41](#_Toc143455149)

[Figura 3.18. Interfaz de Usuario 42](#_Toc143455150)

[Figura 3.19. Controlador de Usuarios 43](#_Toc143455151)

[Figura 3.20. Método GET de un Item 44](#_Toc143455152)

[Figura 3.21. Repositorio del Frontend 45](#_Toc143455153)

|  |
| --- |
| **ÍNDICE DE TABLAS** |

[Tabla 3.1 Involucrados y sus funciones 2](#_Toc143160228)

|  |
| --- |
| **AGRADECIMIENTOS** |

Agradezco primeramente a mis padres Yasmin Raquel y Miguel Delgado por haberme apoyado tanto emocionalmente como económicamente para poder ir desarrollándome como profesional y alcanzar las metas que tengo en mi vida, su apoyo es y ha sido fundamental para mí porque gracias a ellos puedo ir avanzando en mi vida.

También agradezco a mi abuelo José Asención por siempre darme la motivación, el valor y su amor incondicional para enfrentarme a los obstáculos que se presentan en mi vida y dar lo mejor de mí todos los días. Sus consejos y palabras me han ayudado de gran manera en mi vida, demostrándome que con esfuerzo y dedicación puedo lograr lo que sea.

A mi familia en general por su amor y cariño incondicional que me demuestran todos los días de mi vida.

|  |
| --- |
| **RESUMEN** |

Wurth México S.A de C.V cuenta con un departamento de sistemas que además de administrar los sistemas y software de la empresa, también es donde se administra todo el equipo de cómputo, como teclados, monitores, laptops, procesadores, RAM, discos de estado sólido, cámaras, PC, audífonos y cables (HDMI, VGA, Ethernet).

Este documento describe el desarrollo del Sistema de administración de inventario, en el que se incluye tanto como el desarrollo Backend como el desarrollo Frontend de dicho sistema el cual es un proceso en el que se administra y da seguimiento al inventario de computo de la empresa.

El primer capitulo de este documento describe los datos generales de la empresa Wurth México S.A de C.V, así como lo esperado y beneficios de la implementación del Sistema. En el segundo capítulo se identifica los conceptos, metodologías y herramientas utilizadas de software aplicables al desarrollo del Proyecto.

En el tercer capitulo se muestra el desarrollo del Sistema, en el que se describe el inicio, planeación, ejecución, control y cierre del proyecto. Por último, se encuentra el capitulo cuatro el cual describe los resultados obtenidos y las contribuciones alcanzadas del Sistema.

Para el desarrollo de este proyecto se implementa la arquitectura cliente servidor para lo cual se implementa C# del lado del servidor, Angular para el lado del cliente y PostgreSQL como gestor de base de datos.

|  |
| --- |
| **SUMMARY** |

Wurth México S.A de C.V has a systems department that, in addition to managing the company's systems and software, is also responsible for managing all computer equipment such as keyboards, monitors, laptops, processors, RAM, solid-state drives, cameras, PCs, headphones, and cables (HDMI, VGA, Ethernet).

This document describes the development of the Inventory Management System, which includes both Backend and Frontend development. This system is a process for managing and tracking the computer inventory of the company.

The first chapter of this document describes the general information about Wurth México S.A de C.V company, as well as the expected outcomes and benefits of implementing the System. The second chapter identifies the concepts, methodologies, and software tools used in the development of the Project.

The tird chapter the development of the System is presented, describing the initiation, planning, excution, monitoring and closure of the proyect, Finally, chapter four outlines the results obtained and the contributions achieved by the System.

For the development of this proyect, the architecture client-server is implemented. C# is used for the server-side implementation, Angular is used for the client-side, and PostgreSQL is used as the database management system.

|  |
| --- |
| DESCRIPCIÓN DEL PROYECTO |

## Datos generales de la empresa

Würth México S.A de C.V es una empresa especializada en ofrecer y comercializar productos de construcción, automóviles y herramientas. La empresa cuenta con seis divisiones tales como; cargo, auto, reventa, mantenimiento, metal y construcción, su enfoque principal radica en mantener los más altos estándares de calidad, lo que se traduce en la garantía de excelencia en sus productos.

Además, Würth se destaca por emplear las mejores herramientas y estándares disponibles, con el objetivo de lograr una mejora continua en todas sus operaciones. Para ello, la empresa tiene en cuenta de manera prioritaria las necesidades y requerimientos del mercado.



Figura 1.1Logotipo de la empresa

Würth México se encuentra ubicada en Carretera Temixco - Emiliano Zapata Lote 17, Bodega 1 Desarrollo Industrial Emiliano Zapata (D.I.E.Z.), Col. Palo Escrito C.P. 62760 Emiliano Zapata, Morelos frente al C5. Würth cuenta con un total de 700 empleados los cuales están distribuidos en las diferentes áreas: Recursos Humanos, Sistemas, Call Center, Productos, Almacén, Mantenimiento, Contabilidad y Marketing.

Este proyecto se realizará bajo la tutela del Ing. Alejandro Bahena Bravo, Ingeniero en

Tecnologías de la Información por la Universidad Politécnica del Estado de Morelos (UPEMOR), que actualmente desempeña el cargo de jefe de sistemas.

## Antecedentes del proyecto

Actualmente el departamento de Sistemas de Würth no cuenta con un sistema eficiente para la administración de su inventario, lo cual compromete la calidad en el control del inventario. Con el fin de llevar una mejor administración, la empresa requiere un sistema que registre y administre todo el inventario de cómputo y periféricos con el que se tiene dentro de la empresa.

El sistema actual presenta conflictos que impiden el seguimiento adecuado de las especificaciones de cómputo, periféricos y software, así como la gestión de altas y bajas de dichos componentes.

## Objetivo general

Gestionar el proceso de control de inventario de equipos de cómputo, periféricos y software por medio de una aplicación web para el seguimiento de las asignaciones a los usuarios internos de la empresa.

## Objetivos específicos

Identificar cada una de las fases que involucra el proceso de asignación de equipos de cómputo, periféricos y software a los usuarios internos.

Diseñar, analizar y modelar la base de datos a implementar para la persistencia de la información en la aplicación web.

Identificar el proceso que se tiene al registrar, asignar y dar seguimiento a cada uno de los productos que se tienen dentro del inventario.

Diseñar y desarrollar los servicios web para el acceso a la información almacenada de la base de datos.

Definir el proceso de administración y registro que se gestiona desde el lado del cliente.

Pruebas en la aplicación web.

Desplegar la aplicación web en el servidor de la empresa

## Justificación

Ya que el control que se tiene del inventario se gestiona en una hoja de Excel, administrado por un responsable de soporte, ocasiona que la información que se tiene respecto al inventario corra el riesgo de ser perdida, mal gestionada y administrada.

Con el desarrollo de la aplicación web se pretende llevar un mejor control, gestión y seguimiento del inventario de cómputo y periféricos que se tiene dentro de la empresa, así como las asignaciones de estos.

La empresa se encuentra actualmente en un proceso de auditoría con el objetivo de obtener la certificación ISO 9001:2015. Como parte integral de este proceso de aseguramiento de calidad, es necesario implementar un sistema de control de inventario.

## Alcances

* En el desarrollo del sistema se incluye los módulos de gestión y control del inventario.
* El sistema es para el uso del departamento de sistemas de la empresa Würth México.
* Se dará una capacitación a los usuarios finales para la manipulación del sistema.
* La aplicación solo llevará el control de inventario de computo

## Restricciones

* No se puede implementar servicios en la nube, por ello se hará uso del servidor de la empresa Ubuntu Server en la versión 22.04
* La versión de las tecnologías de desarrollo es: PostgreSQL 14.8, .Net Core 7 y Angular en su versión 16.

|  |
| --- |
| MARCO DE REFERENCIA |

## Conceptos, metodologías y herramientas

Para comprender el desarrollo del proyecto, primero es necesario conocer algunos conceptos y herramientas dentro del ambiente de desarrollo que ayudaran a adaptar y desarrollar el proyecto a las necesidades y requerimientos de este mismo.

### Conceptos de arquitectura de software

**Arquitectura de Software**

El concepto de arquitectura de software según nos dice (Palacios, 2021), es la manera en la que un proyecto será construido siguiendo una arquitectura que define la estructura del proyecto, todo esto basándose en las necesidades y requerimientos del proyecto o solución para tener un desarrollo con la mejor estructura que pueda ser óptima para el proyecto.

**Estilo Monolítico**

Este tipo de arquitectura se basa en que todo el proyecto existe dentro de una sola aplicación, según (Palacios, 2021), este estilo describe una aplicación de un solo nivel en la que tanto la vista del usuario y la lógica de acceso y manejo de datos existen dentro de un solo programa. Este tipo de aplicaciones son autosuficientes ya que no dependen de ningún servicio o componente externo para su correcto funcionamiento y ejecución de esta.

Ventajas del estilo monolítico:

* Fácil mantenimiento. Al tener toda la lógica e interfaces dentro de la misma aplicación, se puede dar seguimiento a los problemas vayan ocurriendo y también dar seguimiento a estos mismos.
* Fácil de desarrollar. Las aplicaciones que siguen este tipo de arquitectura son fáciles de desarrollar gracias a que se construyen dentro de una misma aplicación.
* Implementación sencilla. Una única aplicación, una única instalación

Desventajas del estilo monolítico:

* Fiabilidad. Si ocurre un error dentro de algún módulo de la aplicación, este mismo puede afectar toda la aplicación.
* Flexibilidad. Este tipo de aplicaciones están limitadas al uso de tecnologías, esto quiere decir que, si empezamos a desarrollar nuestra aplicación usando el lenguaje Java, en Java debemos de terminar de desarrollar.
* Cambios en la aplicación. Un cambio que hagamos dentro de la aplicación requiere una nueva implementación

**Estilo basado en componentes**

Según nos dice (Palacios, 2021), este tipo de estilo nos permite reutilizar piezas de código para realizar diversas tareas, en general, nos habla a cerca de separar la problemática con respecto a una amplia funcionalidad que pueda existir en un proyecto e ir dando soluciones a esas separaciones por medio de componentes independientes y a la vez la implementación y reutilización de estos.

Ventajas del estilo basado en componentes:

* Reutilización. Este tipo de estilo nos permite que los componentes o piezas de código se implementen y reutilicen en el desarrollo.
* Simplifica las pruebas. Permite hacer pruebas individuales de cada uno de los componentes antes de que sean ejecutados de manera ensamblada.
* Mantenimiento fácil. Si existe algún tipo de fallo, se puede actualizar y/o agregar componentes de manera sencilla sin que estos cambios afecten a otras partes del sistema.

Desventajas del estilo basado en componentes:

* Tiempo. Puede llegar a generar mucho tiempo respecto al desarrollo del sistema.
* Costo. Este tipo de estilos llegan a ser muy costosos debido al tiempo de desarrollo.

**Estilo cliente-servidor**

El estilo de cliente-servidor, según nos dice (Barceló, 2014), permite distribuir sistemas compuestos por: cliente, servidor y una red de interconexión. Este tipo de estilos de arquitectura, describe la relación entre un cliente y uno o varios servidores, donde el cliente puede hacer una o varias peticiones a un servidor, y este, espera respuestas y procesa la llegada de dichas peticiones para después procesarlas y devolver un resultado. Este tipo de aplicaciones trabajan mediante protocolos para el intercambio y manipulación de la información.

Ventajas del estilo cliente-servidor:

* Información centralizada. Este tipo de aplicaciones ofrecen mayor centralización ya que la información se almacena de forma centralizada en el servidor, además las actualizaciones de la información son más fáciles de llevar a cabo.
* Facilidad de integración. Facilita la integración entre diferentes sistemas y comparte la información de una manera más amigable al usuario mediante una interfaz.

Desventajas del estilo cliente-servidor

* Seguridad. Las aplicaciones que trabajan bajo este estilo de arquitectura pueden llegar a presentar un problema en cuestiones de seguridad, ya que el hecho que se comparte información entre servidores y clientes requiere que pasen por protocolos de seguridad los cuales pueden llegar a tener algún tipo de puerta abierta permitiendo ataques de terceros.
* Costos. Si es que los servidores requieren de un procesamiento de alto nivel para su correcto funcionamiento, se podría incrementar el costo con respecto a donde nuestros servidores estén montados.

### Motores de base de datos

**MySQL**

De acuerdo con (Puertas, 2014), MySQL es uno de los gestores de base de datos elegido por la mayoría de programadores. Soporta el lenguaje SQL y la conexión de varios usuarios, pero en general, se utiliza para aplicaciones de tamaño pequeño-medio. Es una herramienta gratuita la cual nos permite entre sus características avanzadas, la replicación, la partición de datos y la gestión de transacciones.

Está diseñado para ser rápido, eficiente y a su vez segura ya que proporciona mecanismos para proteger la base de datos, como la autenticación de usuarios, el control de acceso basado en roles y la encriptación de datos.

**PostgreSQL**

Según (IBM, 2023), PostgreSQL es una base de datos de código abierto que tiene una sólida reputación por su fiabilidad, flexibilidad y soporte de estándares técnicos abiertos. PostgreSQL soporta tipos de datos relacionales y no relacionales. Esto la convierte en una de las bases de datos relacionales más compatibles, estables y maduras disponibles actualmente. Una de las más grandes ventajas de PostgreSQL es su funcionalidad avanzada, ya que permite consultas complejas, indexación avanzada, integridad referencial, disparadores (triggers), procedimientos almacenados y funciones definidas por el usuario. Esto permite desarrollar aplicaciones personalizadas.

**Microsoft SQL server**

Este tipo de motor de base de datos, al igual que los demás, nos permite administrar y gestionar la información con la que nuestras aplicaciones trabajan. Este gestor fue desarrollado por la empresa Microsoft y con base en lo que nos dice (Santamaría, 2016), entre sus beneficios nos ofrece una seguridad robusta gracias a la autenticación de usuarios, autorización basada en roles y el cifrado de datos.

### Tecnologías de backend

De acuerdo con (García, 2021), un lenguaje de backend es la capa de acceso a datos en donde se desarrolla la lógica de negocio, manipulación de datos, comunicación con una base de datos y respuesta a la petición de los usuarios por medio de una interfaz gráfica (mayormente). En pocas palabras, es la parte de nuestro desarrollo que se encarga de servir y atender las peticiones de los clientes. Cabe destacar que quienes se desempeñan en esta área deben de tener conocimientos en servidores como Apache o Tomcat, protocolos de comunicación, gestores de base de datos (relacionales y no relacionales) y sistemas operativos.

**NET Core**

Según lo mencionado por (Alcolea, 2020), NET Core nos permite realizar múltiples aplicaciones para cualquier sistema operativo como Windows, Linux y Mac OS. Esta arquitectura modular permite incluir únicamente lo necesario para desarrollar nuestras aplicaciones, haciendo que no sean tan pesadas a la hora de desplegar el proyecto desarrollado. Al ser multiplataforma no depende de cosas específicas del sistema operativo, es una gran ventaja al desarrollar aplicaciones para diversos sistemas operativos.

**Spring**

Según lo mencionado por (Microsoft, 2023), Spring es una herramienta de código abierto basado en Java que facilita el desarrollo de microservicios y aplicaciones web. Una de las mayores ventajas que nos ofrece Spring es la inyección de dependencias que permite que los objetos definan sus propias dependencias, esto permite que los desarrolladores crear aplicaciones modulares que consta de componentes poco acoplados que son ideales para microservicios y aplicaciones de red distribuida. Spring también ofrece soporte integrado para tareas que las aplicaciones necesitan realizar como, enlace de datos, conversión de tipos, validaciones, manejo de excepciones, etc.

**Express**

De acuerdo con (Kinsta, 2022), Express es un framework de backend minimalista y rápido que proporciona herramientas robustas para desarrollar aplicaciones de backend escalables. Nos ofrece el sistema de enrutamiento y características simplificadas para ampliar el framework con componentes en función a los casos de uso del proyecto.

### Tecnologías de frontend

Las tecnologías de frontend son todas aquellas que se utilizan del lado del cliente, en pocas palabras, es toda la parte grafica que el usuario puede ver e interactuar con ello para asi poder hacer peticiones al servidor y que este muestre la información solicitada por el usuario de una manera más amigable y factible de entender y manipular.

**Angular**

Angular según nos dice (Gonçalves, 2021), es un framework de JavaScript de código abierto que nos permite desarrollar aplicaciones de una sola página (SPA) lo que nos permite la coherencia y reutilización del código en nuestras aplicaciones del lado del cliente. Google es el encargado del mantenimiento y constantes actualizaciones de mejoras para este framework. Con relación a lo que es TypeScript, es una tecnología opcional pero muy recomendable de utilizar al desarrollar nuestras aplicaciones de Angular, y esto, nos permite escribir código JavaScript que es más fácil de entender y mantener.

**React**

De acuerdo con (Gackenheimer, 2015), React es un framework de JavaScript, originalmente creado por ingenieros de Facebook para resolver los retos involucrados cuando se desarrollan interfaces complejas de usuario con conjuntos de datos que cambian todo el tiempo. React utiliza un paradigma orientado a componentes, cada componente es una pieza de software que podemos utilizar en todo el desarrollo de nuestra aplicación (de forma similar a las funciones), comunicándose a través de algo llamado Props, las props son las propiedades que cada componente recibe para la manipulación de estos.

**Vue**

Según nos dice (Mamani, 2019), Vue es un marco de referencia de JavaScript progresivo de código abierto para crear interfaces de usuario (UI) y aplicaciones de una sola página. Este marco de trabajo utiliza alto desacoplamiento, lo que permite a los desarrolladores crear progresivamente interfaces de usuario. Tanto Vue como Angular y React tienen modularización, lo que resalta a Vue es que gracias a su alto desacoplamiento facilita extender funcionalidades que trabajan de una manera muy eficiente a la hora de desarrollo.

### Control de versiones

Las herramientas de control de versiones nos permiten llevar el control de nuestro desarrollo de una manera más organizada lo que permite saber y conocer las diferentes versiones o cambios que fueron surgiendo en nuestro proyecto.

**Git**

Según nos dice (Morcuende, 2022), Git es la herramienta de control de versiones más popular y una de las mejores opciones para usuarios experimentados con perfil técnico, entre las ventajas de Git encontramos:

* Naturaleza distribuida que permite a los integrantes de un proyecto acceder al repositorio desde distintas redes locales.
* Rapidez y eficacia.
* Generación de ramas sencillas.
* Buen rendimiento tanto en grandes como en pequeños proyectos.
* Documentación extensa. Ya que, al ser la herramienta más utilizada por los desarrolladores, suele tener más información acerca de su funcionamiento.

**SVN**

Esta herramienta se basa en una arquitectura cliente-servidor, según nos dice, (Morcuende, 2022), SVN cuenta con algunas funcionalidades interesantes tales como versiones de enlaces simbólicos, soporte para directorios vacíos y registro de operaciones para copiar, mover, borrar y renombra.

**Mercurial**

Como nos dice (Morcuende, 2022), Mercurial es un sistema distribuido ideal para usuarios poco experimentados o perfiles poco técnicos, entre las ventajas de utilizar esta herramienta está el que suele ser muy intuitivo y sencillo de manejar con una curva de aprendizaje mucho más corta.

### Metodologías agiles

Según lo mencionado por (Amaro Calderón, 2007), para asegurar un buen desarrollo de software no basta con tener conocimientos de herramientas y modelado de software, también es importante la metodología del desarrollo la cual provee una dirección correcta con respecto a la aplicación de los demás elementos, todo esto en conjunto con el equipo de desarrollo en la cual se incluye el plan de comunicación tanto de los integrantes del desarrollo así como la comunicación con el cliente dando puntos de mejora en el desarrollo del proyecto. Algunos ejemplos de metodologías agiles son:

**Scrum**

De acuerdo con (Amaro Calderón, 2007), Scrum define un proceso empírico, interativo e incremental que enfatiza prácticas y valores de Project management por sobre las demás disciplinas de desarrollo.

Al principio del desarrollo se define el Product Backlog, el cual contiene todos los requerimientos funcionales y no funcionales que deberá incluir el sistema o proyecto que se desea construir. Así mismo existen los Sprint, los cuales se encargan de ir evolucionando la aplicación evolutivamente. Cada Sprint tiene su propio Sprint Backlog que trabaja en subconjunto con el Product Backlog basándose en los requerimientos a ser construidos en el Sprint correspondiente. La duración recomendada de cada Sprint es de un mes.

**Extreme Programming (XP).**

Según nos dice (Amaro Calderón, 2007), XP es la primera metodología ágil y que dio conciencia al movimiento actual de metodologías agiles. Esta metodología ágil se centra en la velocidad y la simplicidad esto permitiendo potenciar las relaciones interpersonales como clave en el desarrollo del proyecto o sistema. XP da la posibilidad al equipo (entre los cuales se incluyen los programadores, los testers, coach y manger) la posibilidad de adaptarse sin ningún problema a los requerimientos volátiles del cliente, es una gran ventaja ya que, durante el desarrollo de un proyecto, el cliente suele hacer cambios durante este mismo.

**Crystal Clear**

Crystal Clear según nos dice (Amaro Calderón, 2007), se basa en Xtreme Programming pero presentando un enfoque ágil, con gran énfasis en la comunicación y con más tolerancia con respecto a algunas características de XP, en pocas palabras, esta metodología se define mucho más en la manera de comunicarse y la forma muy liviana que se sigue con respecto a los entregables.

Una de las más notables características de esta metodología es la necesidad que se tiene de disponer de un usuario real para ir realizando las validaciones sobre las interfaces de usuario y participar en las definiciones de los requerimientos funcionales y no funcionales del software.

## Propuesta de solución

Con base en la investigación realizada y partiendo de un análisis respecto a las necesidades del proyecto y requerimientos específicos del mismo, se propone la siguiente solución:

Para llevar el control de versiones del proyecto, se utilizará Git para la gestión del desarrollo con el fin de organizar y dar seguimiento a los cambios que se vayan presentado durante el desarrollo del proyecto.

Respecto a la arquitectura del proyecto, seguirá la arquitectura cliente-servidor con el fin de mantener la información centralizada a la hora de guardar y manejar la información que se recibe y tiene. A su vez facilitar la integración con el fin de mostrar la información al cliente de una manera más amigable y que al recibirla, pueda manipular y hacer peticiones al servidor para obtener respuestas por parte de este.

Para el almacenamiento de la información se utilizará PostgreSQL como gestor de base de datos para mantener y manipular la información que se le mostrará al usuario final. También debido a que es una herramienta gratuita con disponibilidad multiplataforma y que gracias a su sistema de fiabilidad y robustez nos permite manejar y almacenar gran cantidad de datos e información que el usuario solicite.

Del lado del servidor, se optará por el lenguaje C# ya que es un lenguaje orientado a objetos y nos proporciona el nivel de abstracción requerido para desarrollar el proyecto, todo bajo la tecnología de .Net que nos proporcionará las utilidades y servicios para un desarrollo óptimo. Esto debido a que es un lenguaje que cuenta con una amplia librería que facilita y asegura el desarrollo del proyecto.

Para el lado del cliente se utilizará Angular, esto debido a que este framework nos ofrece muchas ventajas para desarrollar nuestra aplicación del lado del cliente garantizando en su mayoría que este no presente ningún tipo de inconformidad con respecto a como se muestra y maneja la información. Entre las ventajas de usar este framework está el que nos permite desarrollar códigos altamente optimizados que dan como resultado un rendimiento más optimo, además de que Angular cuenta con una CLI (Command Line Interface) que nos permite construir en menor tiempo y agregar componentes al instante. Angular incluye muchos beneficios, pero lo que más destacable es que utiliza TypeScript para el desarrollo de las aplicaciones del lado del cliente, el utilizar TypeScript nos da la gran ventaja de hacer un desarrollo de una manera más entendible, estricta, mantenible y optimizada en la que al ir escribiendo código no tendremos inconveniente en entender el porqué de ciertas funciones y que tipo de datos reciben estas mismas. Angular nos permitirá entregar una aplicación ligera y de calidad debido a que se siguen las mejores técnicas y estándares para la construcción y desarrollo de esta misma.

Con respecto a la metodología ágil se planea seguir la metodología Scrum, esto debido a que las reuniones diarias y las demás características de Scrum nos permiten tener un desarrollo más organizado, planificado y controlado. Las reuniones nos permiten conocer el avance del proyecto y llevar un seguimiento con respecto a los puntos de mejora en el desarrollo.

|  |
| --- |
| DESARROLLO |

## Inicio

Para el desarrollo de este proyecto se llevó a cabo una reunión para analizar cuál es una problemática principal con la que cuenta el departamento de Sistemas junto con el área de Soporte, en esta reunión estuvieron presentes el jefe del área de Sistemas, el Ing. Alejandro Bahena Bravo, y el jefe del área de Soporte, el Ing. Jaime Rojas.

En esta reunión se identificaron varios puntos a mejorar respecto a la manera en la que se gestiona y controla el inventario con el que se cuenta en el departamento de Sistemas.

Se identificó que la manera en la que se asignan y se tiene gestión del inventario de computadoras y periféricos no es la más apta y adecuada de llevar, por lo tanto, se requiere de un sistema el cual permita llevar un control más eficaz para la gestión y asignación del inventario con el que se cuenta dentro del área de Sistemas.

Una vez definido lo que se va a hacer, se llevó a cabo una segunda reunión con el objetivo de conocer el proceso que se lleva a cabo al momento de asignar equipos de cómputo y periféricos, así como el proceso para registrar los periféricos que se tienen dentro del departamento de Sistemas.

Con apoyo del Ing. Manuel Aldana, se definió el objetivo general y los objetivos específicos del proyecto. También se elaboró el DFR (Documento Formal de Requerimientos) en los cuales se especifica más a detalle lo que requiere el sistema para su correcto y optimo funcionamiento, así como las reglas de negocio y especificaciones del cliente.

Tabla 3.1 Involucrados y sus funciones

|  |  |  |
| --- | --- | --- |
| **INVOLUCRADOS EN EL PROYECTO** | | |
| **Nombre de la persona** | **Función** | |
| Dr. Hugo Omar Alejandres Sánchez | | Asesor Universitario |
| Würth México S.A. de C.V. | | Cliente |
| Ing. Alejandro Bahena Bravo | | Asesor Institucional |
| Ing. Jaime Rojas | | Jefe del área de Soporte |

## Planeación

Para la planeación del proyecto, se tomaron en cuenta los requerimientos funcionales y no funcionales del cliente, así como las reglas de negocio y el proceso que se lleva a cabo para la asignación y registro del inventario de Sistemas, de este modo, nos permite tener y conocer las bases necesarias para realizar el proyecto de una manera más eficiente.

### Documento Formal de Requerimientos

Con el propósito de conocer más a detalle las necesidades del proyecto, entre las cuales se incluyen los requerimientos funcionales y no funcionales, se llevó a cabo un análisis con base en la información ya planteada del proyecto, esto con el fin de definir el documento formal de requerimientos como se muestra en el ANEXO A. DOCUMENTO FORMAL DE REQUERIMIENTOS.

El objetivo del documento formal de requerimientos es definir y conocer más a detalle a cerca de nuestro proyecto, entre esto se incluye los requerimientos funcionales, no funcionales, datos de entrada, criterios de aceptación, los alcances del proyecto y las reglas de negocio.

### Plan de comunicación

Con el fin de ir conociendo más a detalles el proceso que se tiene y el que se desea tener en el sistema, se implementa un plan el cual nos permite mantener una comunicación con ayuda de la herramienta de Slack y reuniones las cuales son con el fin de modificar algunas partes con respecto a el proceso que llevará el proyecto.

### EDT

El diagrama de descomposición del trabajo (EDT), nos permite conocer las etapas durante el desarrollo del proyecto desde el análisis hasta la liberación de este, el EDT también desglosa las tareas de cada etapa del desarrollo, esto nos permite conocer más a detalle el proceso que se seguirá durante cada etapa del proyecto. En la Figura 3.1 se muestra el EDT del proyecto el cual se está desarrollando.

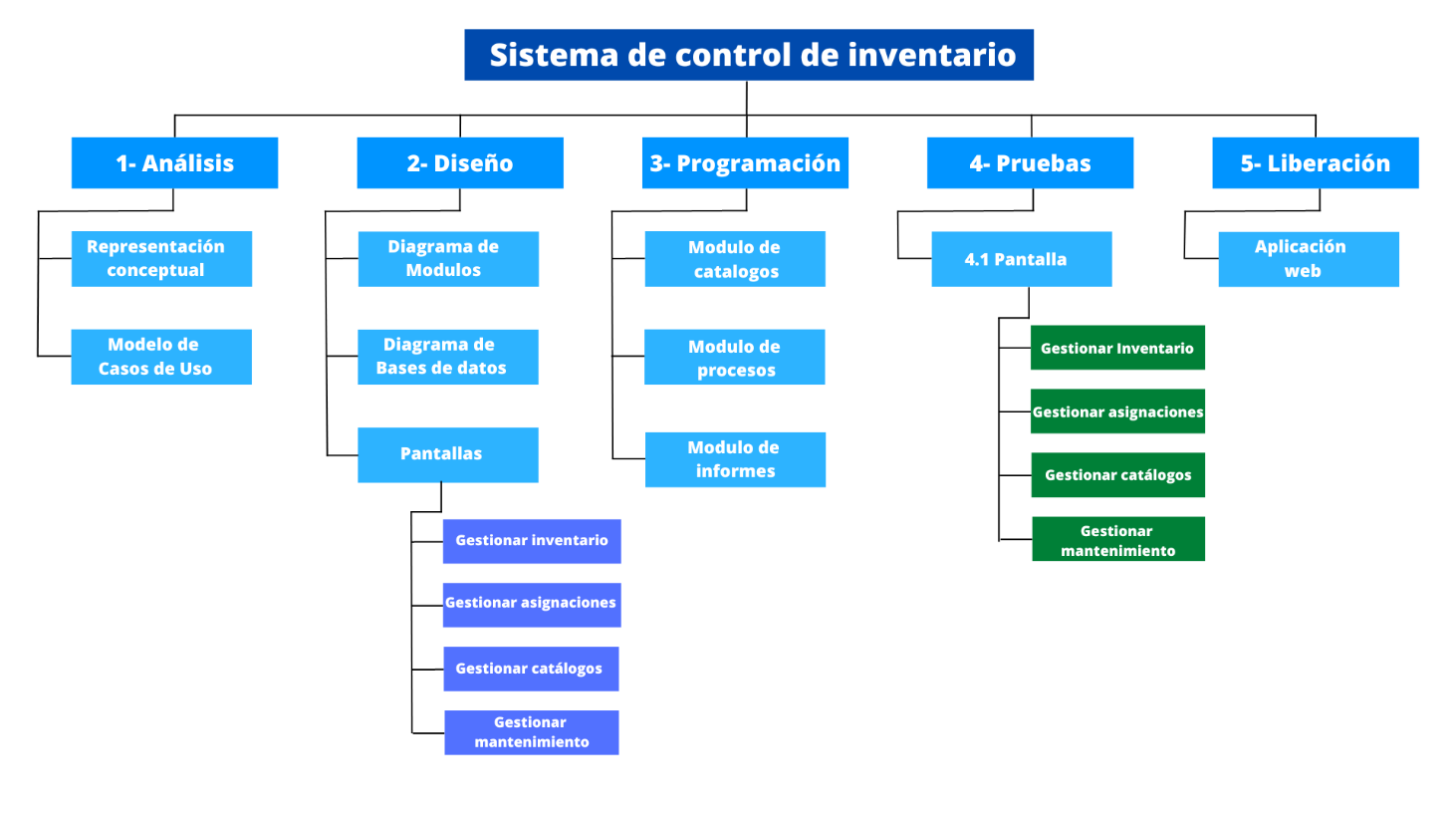


Figura 3.1 EDT

### Cronograma de seguimiento

Para llevar un seguimiento con respecto a los tiempos de desarrollo del proyecto, se define un cronograma de seguimiento el cual propone las actividades a realizar con sus fechas correspondientes. Como se observa en la Figura 3.2, el cronograma plasma avance de acuerdo a las fechas establecidas, se puede tener un mejor control con respecto a la organización de desarrollo y de esta manera administrar y organizar nuestros tiempos durante el desarrollo.



Figura 3.2 Cronograma de seguimiento parte 1



Figura 3.2. Cronograma de seguimiento parte 2

### Diagrama de Gantt

Se desarrolla el diagrama de Gantt con el fin de llevar un seguimiento con respecto al plan y/o proceso de desarrollo del proyecto. Este diagrama nos proporciona las actividades con sus fechas en las cuales se planean ejecutar estas mismas. En la Figura 3.3,se muestra el diagrama de Gantt del proyecto, el cual plasma las actividades a realizar en las fechas definidas.

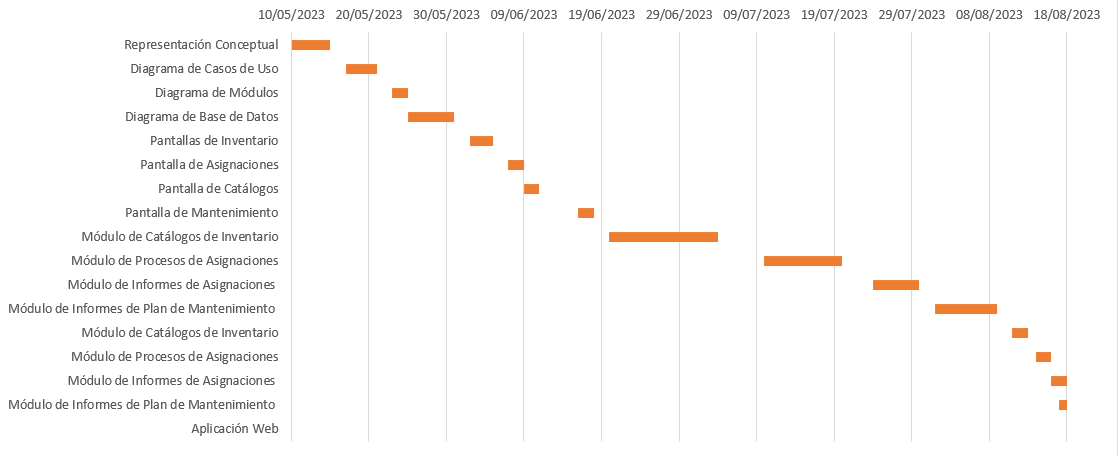


Figura 3.3. Diagrama de Gantt parte 1

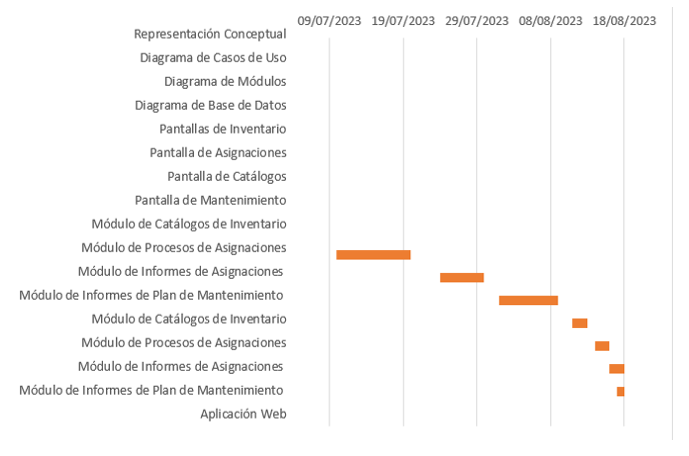


Figura 3.3. Diagrama de Gantt parte 2

## Ejecución

### Representación conceptual

En la Figura 3.4 se muestra el rol de administrador el cual se encarga de gestionar toda nuestra aplicación, también se muestra toda la parte del proceso que se lleva a cabo para asignar periféricos y/o equipos de cómputo a los usuarios internos, asi como, la manera en la que se registran lotes de inventario del área de Sistemas y que son administrados por estos mismos. Cabe destacar que la representación conceptual va en conjunto con el diagrama de casos de uso ya que este último se enfoca en las funciones principales del sistema, asi como el comportamiento de este mismo.

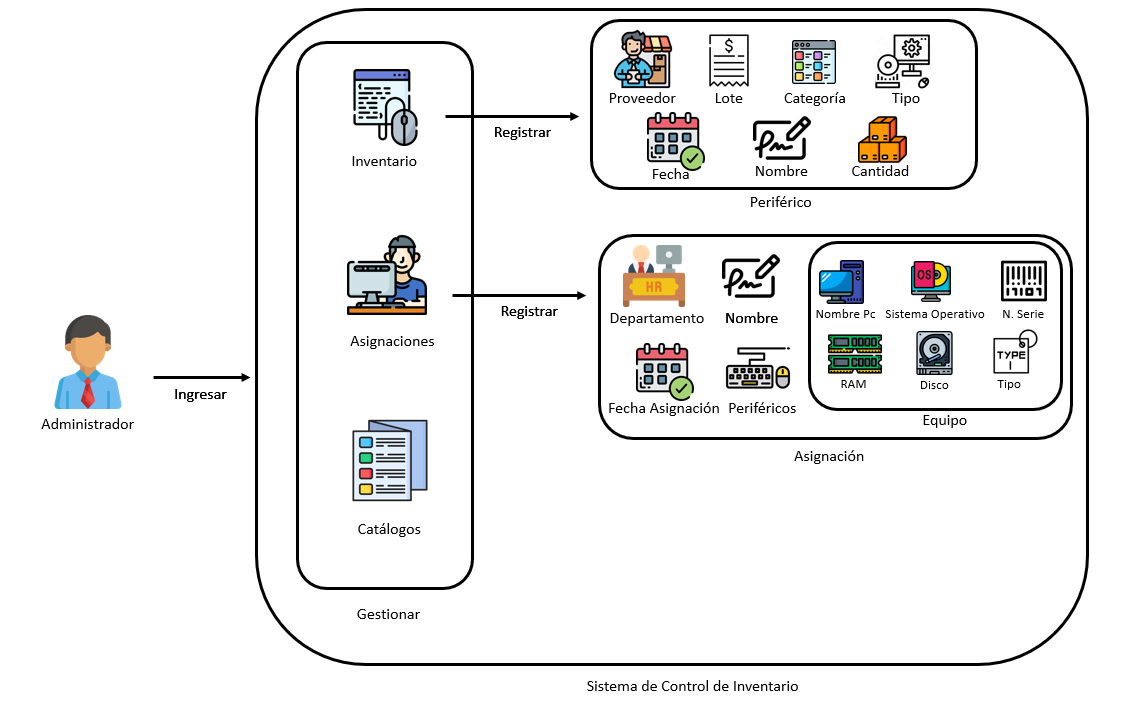


Figura 3.4 Representación conceptual

La representación conceptual nos permite conocer de una manera más grafica el proceso de nuestro sistema, así como la manera en la que está estructurado y definido nuestro proyecto.

Con el objetivo de comprender el funcionamiento del sistema con base en la representación conceptual, se describen las fases las cuales definen los procesos del sistema.

**Proceso de registro de inventario.**

Una vez que llega un nuevo lote de inventario al área de Sistemas, el encargado asignado debe de registrar ese lote y los periféricos que este tiene dentro de un formulario para después visualizar la información en una lista la cual nos permite consultar todo el inventario que hay dentro del departamento de Sistemas.

**Proceso de asignación de periféricos y/o equipos de computo**

Para el proceso de asignación a usuarios de la empresa, el encargado debe de tener un registro previo de los usuarios antes de hacer una asignación, para esto, puede visualizar la información a través de una lista en la cual selecciona el usuario al cual desea asignar un periférico o equipo de cómputo. Este proceso nos ayuda a conocer como están distribuidos los periféricos y equipos de cómputo dentro de la empresa Würth México y de esta manera tener el control y poder dar seguimiento a estos.

**Proceso para gestionar catálogos**

El encargado de la aplicación podrá llevar la gestión de los catálogos los cuales el sistema requiere al momento de hacer el registro del inventario, estos catálogos contienen información como el tipo de inventario (hardware, software), marcas, tipo de equipo (laptop, pc), departamentos a los que pertenece cada usuario, tipo de periférico (monitor, procesador, disco, teclado, mouse, cables), se podrá modificar o agregar información a estos catálogos con el fin de también llevar la gestión de las marcas que el proveedor proporciona a la hora de traer los lotes que contienen los periféricos y/o equipos de computo

### Diagrama de casos de uso

Para comprender lo que el sistema hace, se realizó un diagrama de casos de uso como se muestra en la Figura 3.5 en el cual se reflejan las funciones principales del proyecto, asi como lo que es obligatorio u opcional de realizar en cada uno de los casos de uso.

La ventaja de este diagrama es proporcionar al desarrollador una vista más estructurada con respecto a los requisitos funcionales en el contexto de procesos empresariales y de un sistema, asi mismo, nos permite manejar la complejidad de sistemas robustos.

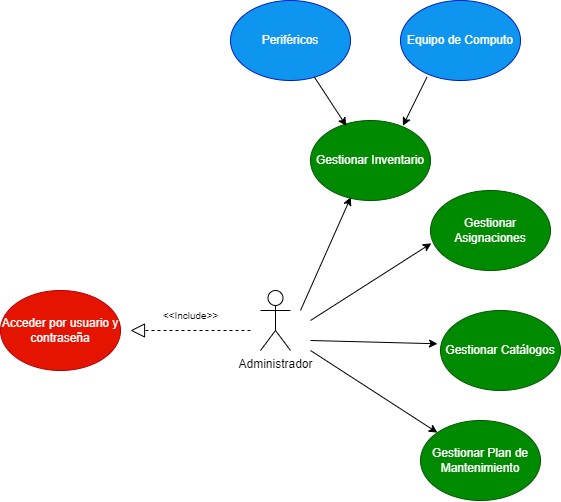


Figura 3.5 Diagrama de Casos de Uso

Los casos de uso con gestionados por un solo usuario que es el administrador, se llevará la gestión del inventario que a su vez incluye los periféricos y el equipo de cómputo del departamento de Sistemas, así mismo se lleva la gestión de asignaciones con respecto al inventario con el que se cuenta, también se lleva la gestión de los catálogos que se manejan a la hora de registrar y gestionar el inventario, por último, se gestiona el plan de mantenimiento con respecto al equipo de cómputo.

### Estructura del proyecto del lado del cliente

Dentro del proyecto se sigue una estructura, la cual nos permite llevar de una manera más organizada el desarrollo de nuestro proyecto con el fin de hacer un código más legible y organizado. En la Figura 3.6 describe de manera gráfica la estructura de carpetas del lado del cliente, cabe mencionar que para consumir los servicios de nuestra aplicación se utiliza el Framework Angular, esto debido a que el desarrollo es

más estricto y ordenado a comparación de otros Frameworks de desarrollo.

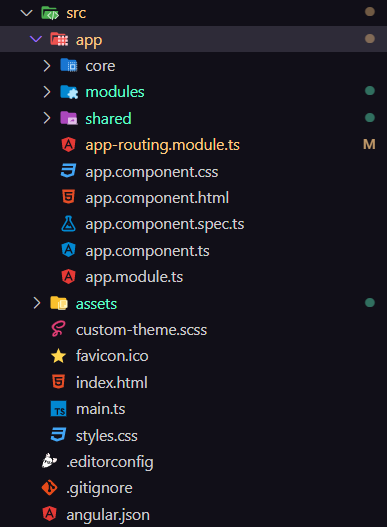


Figura 3.6 Estructura de carpetas

**App**

Dentro de esta carpeta se encuentran los módulos, componentes e imágenes que el proyecto utiliza y que son de vital importancia para que este funcione, a continuación, se describe más a detalle cada carpeta dentro de ‘App’:

* Modules. Dentro de esta carpeta se definen los componentes de cada sección de la aplicación, así como sus respectivos módulos para que nuestra aplicación se comunique a través de estos. Asi mismo cada módulo tiene su módulo de routing el cual nos permite definir las rutas y los componentes que se cargan una vez nosotros accedamos a estas rutas.
* Core. En esta carpeta se definen elementos como las interfaces las cuales nos permiten definir un tipo de dato con el cual deseamos que se trabaje dentro de una función o de una definición. Estas interfaces son reutilizables lo cual nos permite utilizarlas durante todo nuestro desarrollo.
* Shared. Dentro de shared se definen los componentes compartidos como lo es la barra de navegación, formularios, listas y pipes (dentro de estos se realizan funciones para ordenar listas).

**Creación de un módulo.**

Un módulo, cuando trabajamos con Angular, se refiere a un apartado el cual nos permite organizar y/o agrupar los componentes, directivas, pipes y servicios de nuestra aplicación. Un módulo maneja la agrupación lógica de nuestra aplicación permitiendo manejar módulos en vez de componentes. En la Figura 3.7, se muestra el comando que nos ofrece la CLI (Command Line Interface) de Angular para crear módulos sin la necesidad de crearlos manualmente,

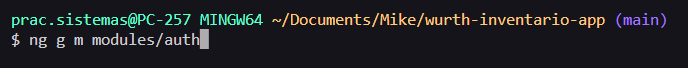


Figura 3.7 Creación de un módulo

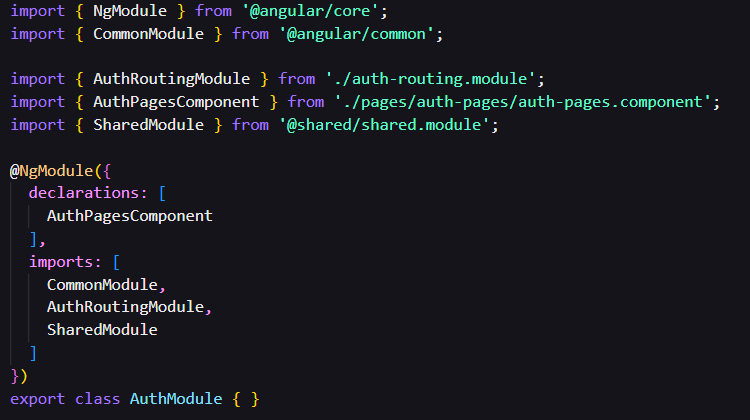


Figura 3.8. Módulo de Auth

Dentro de las importaciones se encuentra el módulo de Shared, el cual, es un módulo el cual contiene componentes tales como el formulario del inicio de sesión, componentes para dar estilos.

**Creación de componentes.**

Un componente archivo el cual está compuesto por nuestro template, que es una vista HTML que se visualiza en la interfaz del usuario. Dentro de un componente se pueden definir métodos, estilos y estructura HTML dentro de este mismo, es una gran ventaja ya que al, por ejemplo, trabajar con estilos, estos se aplicarán solamente sobre el componente en el que estamos trabajando sin afectar a los demás. En la Figura 3.9 ,se muestra un ejemplo de la creación de un componente.

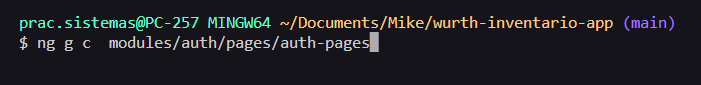


Figura 3.9 Creación de un componente

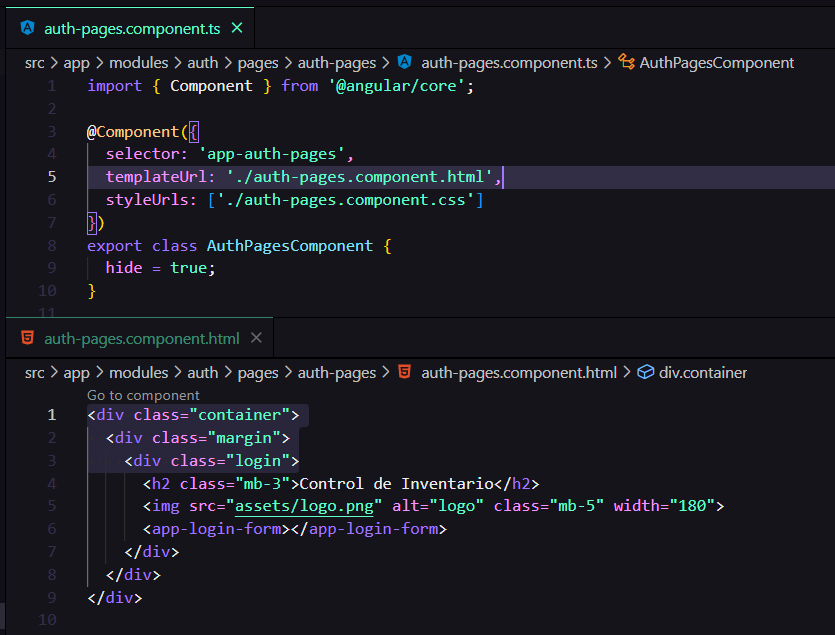


Figura 3.10 Componente de Auth

### Diseño de vistas

**Vista de inicio de sesión**

Dentro del lado del cliente, se diseñó la vista y el formulario de inicio de sesión el cual permite al usuario autenticarse y poder acceder al sistema.

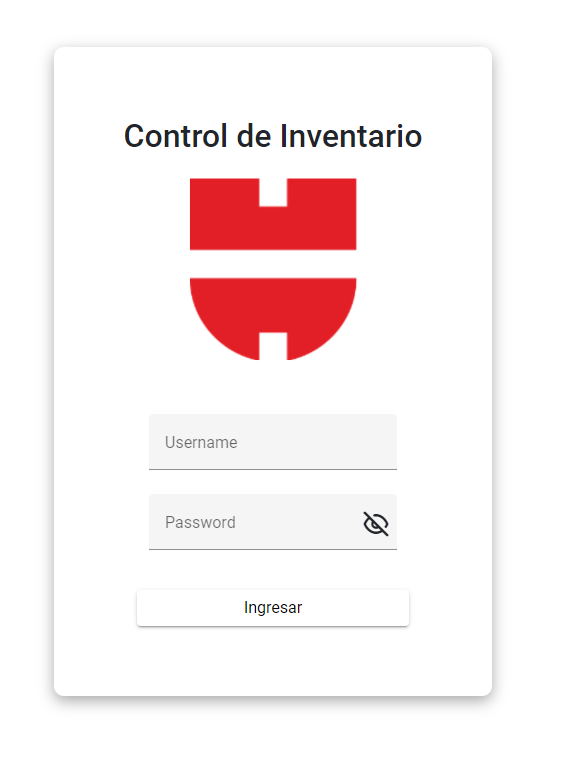
****

Figura 3.11 Diseño de vista login

**Vista de asignaciones**

En la Figura 3.12, se muestra el listado de los usuarios los cuales tiene asignado un equipo de cómputo y/o periféricos como monitores, teclados, nobreaks y mouse. Esta vista se realiza con el fin de mantener un control sobre quienes tienen alguna asignación y así poder dar seguimiento a estos mismos.

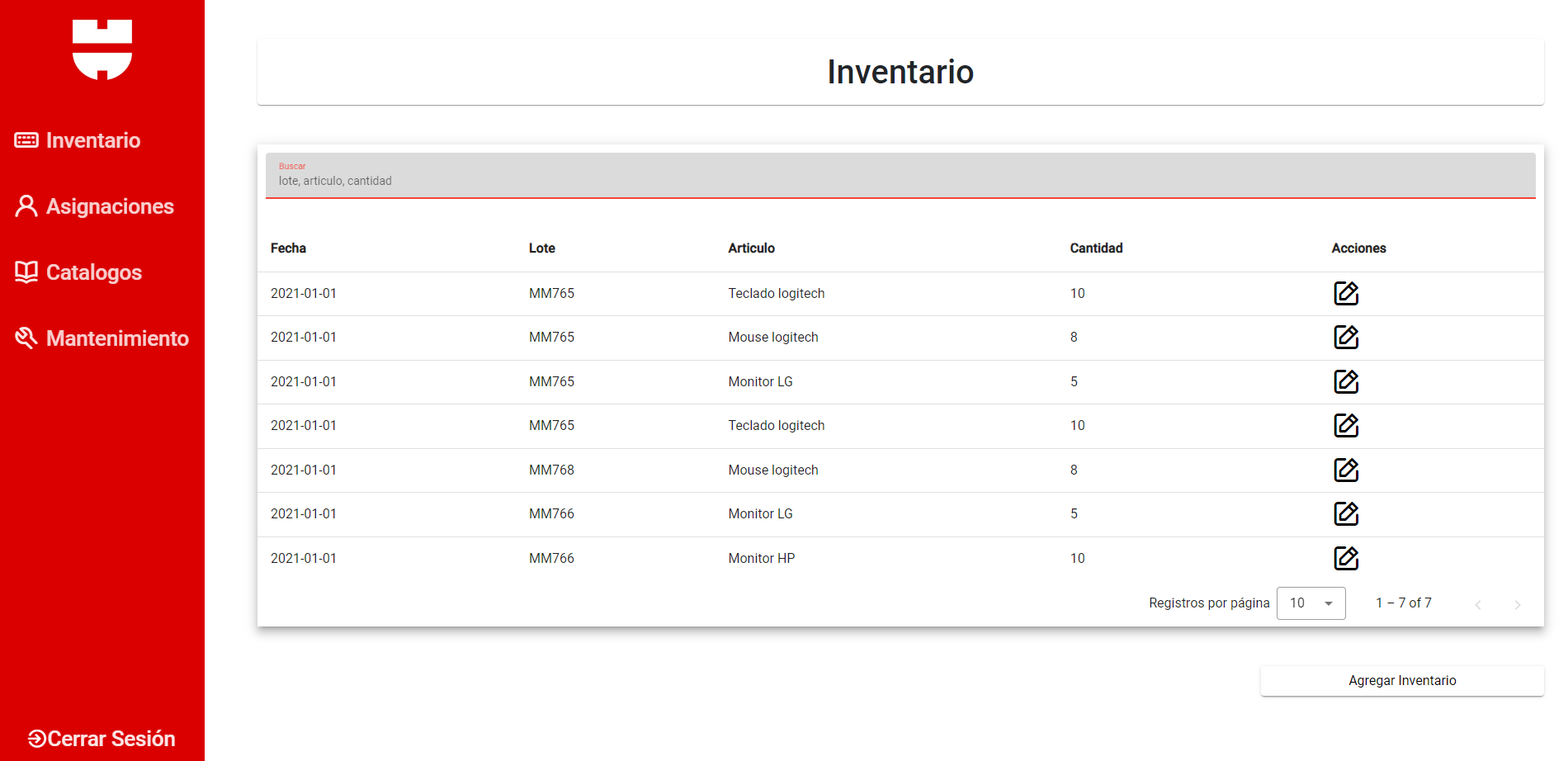


Figura 3.12 Listado de Inventarios

Esta vista cuenta con filtros y paginación, esto con el fin visualizar los registros de una manera más cómoda y rápida.

### Base de datos

Para persistir la información del proyecto, se define un modelo de base de datos el cual nos permite manipular y consultar la información con la que trabaja el proyecto, asi mismo como se muestra en la Figura 3.13, se muestra el diagrama de la base de datos de una manera más visual con el fin de conocer las relaciones y datos con los que se trabaja.

Cabe resaltar que para la definición de la base de datos se tomó en cuenta el proceso que se tiene y que se desea llevar para el registro del inventario de Sistemas y la asignación de estos mismos, así también como el plan de mantenimiento de los equipos de cómputo con el fin de llevar un control no solo en la asignación del equipo de cómputo y de los periféricos, sino también en el seguimiento con respecto al mantenimiento.

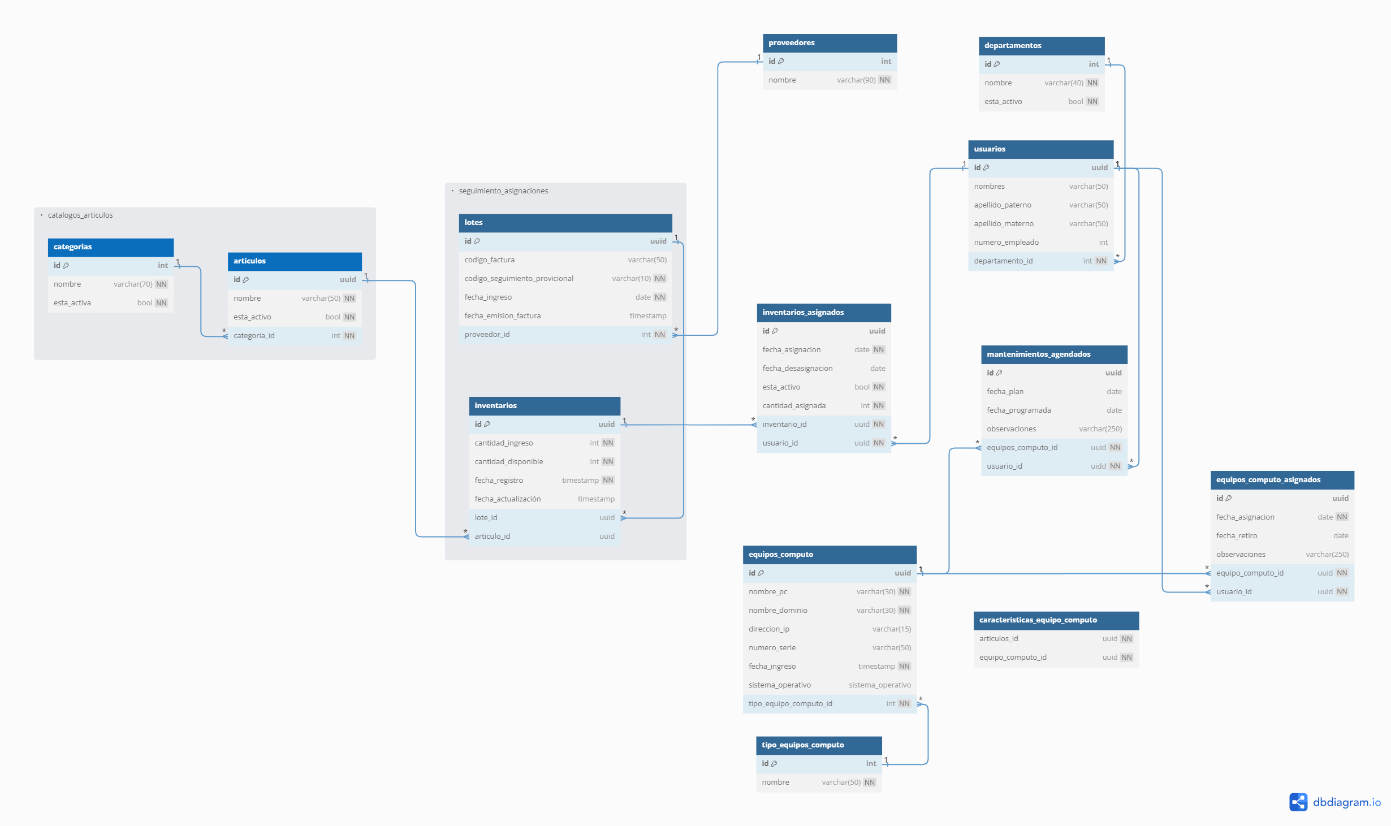


Figura 3.13 Diagrama de la base de datos

En la tabla de artículos se definen el nombre y la categoría de cada artículo, esta última se relaciona con la tabla de categorías la cual nos permite definir si el artículo que se registra pertenece a la categoría de Hardware o Software.

Al momento de hacer el registro de inventario, se necesitan de 4 tablas que son las principales en este proceso:

* Tabla de proveedores. En esta tabla se registran los proveedores con los cuales la empresa Würth México realiza la compra del inventario de Sistemas (periféricos, equipos de cómputo, servidores), para esto se registran los nombres de los proveedores.
* Tabla de lotes. Para hacer el registro del inventario que llega al departamento de Sistemas, es necesario registrar el lote en el que ese inventario llega, esto con el fin de conocer las fechas en las que se terminan los artículos de ese lote. El lote no es más que el número de factura que emite el proveedor al entregar los artículos los cuales entrega al encargado del departamento de Sistemas.
* Tabla de inventarios. En esta tabla se definen más a detalle la información con respecto a los artículos y el número de lote de estos mismos. Esta tabla nos permite registrar y llevar el control de cada artículo con respecto a la cantidad que ingresan, la fecha en la que ingresan, la cantidad actual disponible y la fecha en la que se acaban.
* Tabla de departamentos. Esta tabla nos permite manejar la información con respecto a los departamentos a los que pertenece cada usuario.
* Tabla de usuarios. La tabla de usuarios nos permite llevar el control de la información de los usuarios internos de la empresa, asi mismo, esta tabla nos ayuda a la hora de realizar las asignaciones.
* Tabla de inventarios asignados. Esta tabla está relacionada con la tabla de ‘usuarios’ y la de ‘inventarios’, y nos permite realizar las asignaciones correspondientes para cada usuario, dentro de esta tabla se lleva la información de los periféricos que cada usuario tiene (monitores, mouse, teclados, nobreaks). También nos permite manejar la información del día en el que se asignan y la fecha en la que se desasigna ese periférico.
* Tabla de equipos de cómputo. Esta tabla, así como la del inventario también nos permite llevar el registro de los equipos de cómputo dentro de la empresa, los cuales, son asignados a los usuarios. Se relaciona con la tabla de tipos de equipos de cómputo, esto permite definir el tipo de cada equipo, si es computadora de escritorio o una laptop.
* Tabla de características de equipos de cómputo. Esta tabla es más como una tabla informativa, la cual se usa al momento de hacer una asignación a un usuario, ya que, basándose en el tipo de usuario se asigna un equipo de cómputo necesario para este y que el equipo de cómputo sea el adecuado cumpliendo con las necesidades del usuario al que se asigna.
* Tabla de equipos de cómputo asignados. Lo que nos permite realizar esta tabla es realizar las asignaciones correspondientes a cada usuario dentro de la empresa, así mismo, tener el control del equipo de cómputo que están asignados pudiendo ver la fecha en la se le asigna al usuario.
* Tabla de mantenimientos agendados. Con respecto al mantenimiento de los equipos de cómputo, se cuenta con esta tabla que nos permite dar seguimiento a las computadoras con las que cada usuario cuenta con relación al plan de mantenimiento con el fin de contar con equipos de cómputo actualizados y en correcto funcionamiento. Se define la fecha plan, que es la fecha en la que el encargado de mantenimiento del departamento de Sistemas crea que es pertinente realizar el mantenimiento, y la fecha programada.

**Creación de una tabla**

Para la creación de tablas y manejo de la base de datos, se usa el gestor de base de datos PostgreSQL en su versión 14.8 para facilitar el manejo de nuestra base de datos y la información la cual mantiene. En la Figura 3.14, se muestra la creación de una tabla que en este caso es la de Inventario.

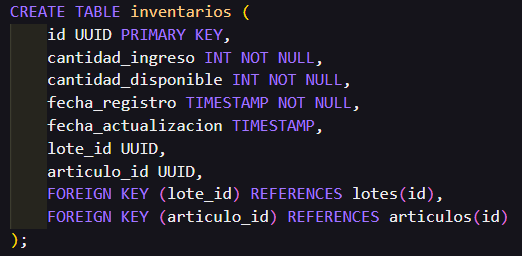


Figura 3.14. Tabla de Inventarios

Como lo muestra la Figura 3.14, la tabla de inventario contiene los campos que son necesarios para persistir la información respecto al inventario, así mismo, también contienen llaves foráneas las cuales relacionan la tabla con otras dos más que en este caso con las de lotes y artículos.

En la Figura 3.15, se muestran los campos de la tabla de lotes que a su vez van relacionados con la tabla de inventarios. Esta relación, permite persistir la información de una manera más ordenada,

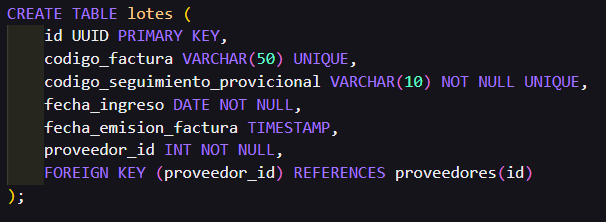


Figura 3.15. Tabla de lotes

### Servidor

Del lado del servidor se propone llevar una estructura, así como la del lado del cliente, para esto se propone la siguiente solución como se muestra en la Figura 3.16

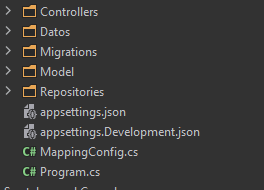


Figura 3.16. Estructura del Servidor

**Model**

Dentro de la carpeta de Model, se definen las clases que contienen los mismos atributos que las tablas con las que se relacionan, de esta manera se pueden mapear las clases basándose en el modelo de la base de datos ya definido. En la Figura 3.17, se muestra el ejemplo de una clase que está siendo definida con el nombre de la tabla, los atributos de esta, y la relación que tiene con la tabla de departamento.



Figura 3.17. Modelo de Usuarios

De esta manera es como se mapea y relaciona la clase basándose en el modelo de la base de datos. Esto con la finalidad de manejar la información de una manera correcta en la que sea sencillo hacer peticiones a la base de datos y que esta no cause ningún conflicto.

**Repositories**

En esta carpeta se crean las interfaces en las cuales se definen los métodos que usa nuestro controlador. Como se muestra en la Figura 3.18, cada método recibe un tipo de dato, que, en este caso, recibe un modelo de clase el cual permite manejar objetos de la misma clase que se define en el método.



Figura 3.18. Interfaz de Usuario

El objetivo de una interfaz, es crear una especie de contrato que con base en donde se defina, este caso el controlador, el mismo debe de implementar los métodos que la interfaz define.

**Controllers**

Dentro de la carpeta de controllers, se definen los controladores los cuales a través de métodos HTTP se pueden acceder a los métodos definidos dentro de este mismo. Como se muestra en la Figura 3.19, se define el controlador y la interfaz por medio de la inyección de dependencias, al hacer esto, seguimos los métodos ya definidos en la interfaz, lo cual es importante porque obliga al controlador a seguir una estructura que se basa en lo que ya está definido en la interfaz.

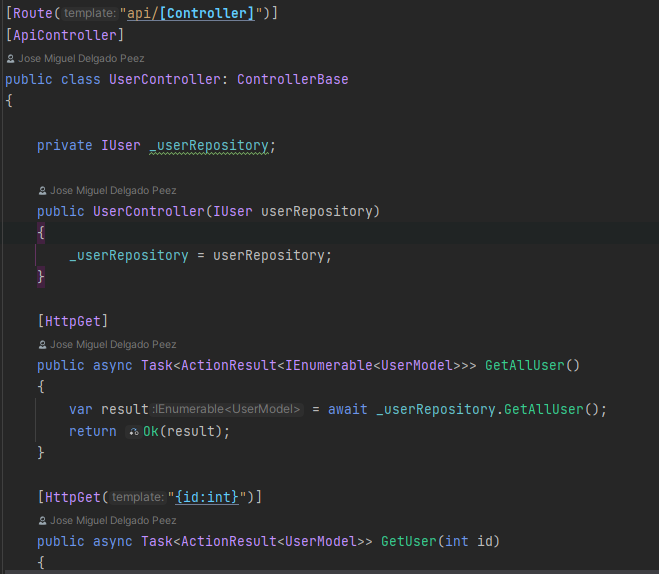


Figura 3.19. Controlador de Usuarios

En cada método se define la acción de este mismo, en el caso del método GetAllUser implementan más métodos que ya vienen definidos en la interfaz. Pero nos solo se pueden implementar métodos que ya vienen por defecto, sino que también se puede realizar nuestras propias consultas, es decir, con base en la necesidad de cada caso se pueden realizar querys los cuales nos dan los datos que requiramos solamente o realizar consultas complejas dependiendo de cada caso.

Como se muestra en la Figura 3.20, se muestra otra manera de realizar las peticiones a la base de datos, en este caso, se pasa un id como parámetro de búsqueda el cual se usa al momento de realizar el query. Es otra manera en la cual se puede hacer consultas a la base de datos.

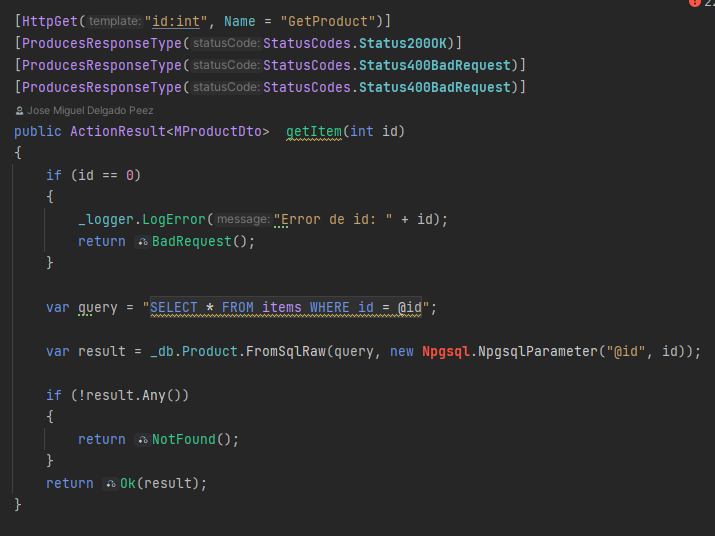


Figura 3.20. Método GET de un Item

### Control de versiones

Con el fin de llevar un mejor control con respecto a las versiones del sistema, como se muestra en la Figura 3.21 se usa Github para almacenar nuestros repositorios y Gitbash que es una terminal de Git para administrar de una mejor manera los avances y versiones del proyecto, esto permite administrar las versiones que surjan durante el desarrollo del proyecto.

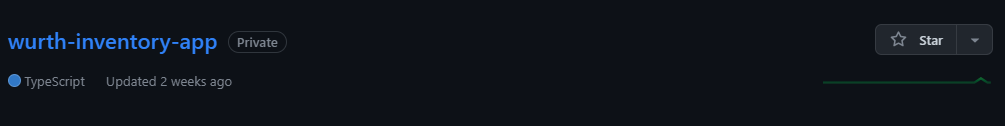


Figura 3.21. Repositorio del Frontend

### Pruebas

Al finalizar el proyecto se tuvo que realizar pruebas para verificar el correcto funcionamiento del sistema, esto el fin de entregar un sistema funcional y cumpliendo con el compromiso de un sistema de calidad al validar los requerimientos funcionales y no funcionales solicitados por el cliente

### Despliegue

Para el despliegue de la aplicación web se uso

## Control

Para llevar un control durante el desarrollo del proyecto, se llevaron a cabo reuniones con el jefe de Soporte el Ing. Jaime Rojas y con el Ing. Manuel Aldana, para buscar puntos de mejora en el proceso de las asignaciones, registro y consulta de los datos dentro del sistema, así mismo, con el fin de validar los datos de entrada con respecto al registro del inventario.

También se realizó el seguimiento de las actividades con el fin de llevar un control en las fechas, esto permite tener un desarrollo más controlado para la finalización del proyecto.

Así mismo se validaban los avances del proyecto periódicamente con el fin de mantener los criterios de aceptación como el cliente los solicitó.

## Cierre

En el momento de finalización del proyecto, se mostró el sistema al cliente con el fin de que analizara el funcionamiento de este, así mismo, se analizaron las funciones y procesos del sistema con respecto al registro del inventario, asignaciones y plan de mantenimiento. También se solicitaron los servicios necesarios para que el sistema pueda funcionar y se puedan consumir del lado del cliente para la manipulación de la información.

Se hizo entrega del proyecto al Jefe de Sistemas el Ing. Alejandro Bahena Bravo, y que, a su vez, se solicitó el proceso de liberación como se muestra en la carta de liberación en el ANEXO B. CARTA DE LIBERACIÓN, donde se menciona que el proyecto fue entregado y el periodo de estadías fue finalizado satisfactoriamente.

|  |
| --- |
| CONCLUSIONES |

## Cumplimiento de objetivos

Se logró un resultado favorable en el desarrollo y fin del proyecto, todo esto gracias a la adecuada identificación del problema que en beneficio a esto se definieron las bases, metodologías y tecnologías a implementar en el proyecto. El identificar cada una de las partes del proceso de asignación de equipos de cómputo, así como el registro de estos favoreció al desempeño del sistema.

El definir la arquitectura de software a seguir permitió que se llevara un correcto desarrollo, todo esto gracias al análisis desde el comienzo del proyecto y también a profundizar más en los requerimientos del cliente.

El desarrollo del servidor se logró completar y definir correctamente gracias a la correcta definición de cada uno de los procesos que involucra el proyecto y que con base en los procesos manuales del cliente se definieron, facilitó la implementación de los módulos en cada uno de los casos.

Para la persistencia de los datos que usa nuestra aplicación, se modeló y diseñó una base de datos la cual nos permite administrar la información de una manera centralizada para evitar la pérdida de información.

Se realizaron pruebas de cada uno de los casos de uso del sistema, esto ayudó a que hubiera puntos de mejora y a entregar un proyecto en el que se cumplieran los requerimientos funcionales y no funcionales solicitados por el cliente.

Al finalizar el proyecto se montó en un servidor , para su correcto funcionamiento

## Resultados

Gracias a la fase de programación y pruebas, se realizaron pruebas en las cuales se pudieron definir los puntos de mejora con respecto a los procesos de asignación de inventario y equipo, registro de asignación de inventario y equipo, y plan de mantenimiento de los equipos de cómputo. Esto permitió obtener resultados positivos y eficientes garantizando la satisfacción del cliente

Gracias al proyecto, el departamento de Sistemas logró dar seguimiento a su inventario de cómputo y periféricos, así como obtener un mejor proceso con respecto a las asignaciones y plan de mantenimiento

## Contribuciones

Dentro del departamento de Sistemas de la empresa Würth México S.A de C.V, se implementó el proyecto con el fin de llevar un mejor seguimiento con respecto al inventario de cómputo y periféricos sin tener la necesidad de usar hojas de cálculo lo cual propicia a la perdida de información fácilmente. Esto permitió al departamento tener un mejor control y gestión de los procesos de asignación y mantenimiento de los equipos de cómputo.

Al desarrollar el proyecto, se adquirieron nuevos conocimientos respecto a los estilos de arquitectura de software, frameworks de desarrollo, lenguajes de backend, herramientas de control de versiones y a realizar documentación de un proyecto de software.

|  |
| --- |
| **REFERENCIAS** |

Alcolea, C. D. (09 de 11 de 2020). *Qué es .NET Core.* Recuperado el 20 de 08 de 2023, de https://openwebinars.net/blog/que-es-net-core/

Amaro Calderón, S. D. (2007). *Metodologías Ágiles.* Trujillo, Perú: Universidad Nacional de Trujillo. Recuperado el 24 de Junio de 2023

Barceló, A. V. (2014). *Desarrollo global de software.* RA-MA Editorial. Recuperado el 26 de 06 de 2023

Gackenheimer, C. (2015). *Introduction to React.* Apress. Recuperado el 27 de junio de 2023

García, I. J. (30 de marzo de 2021). *servnet*. Recuperado el 26 de junio de 2023, de https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programacion-de-una-aplicacion-web

Gonçalves, M. J. (13 de octubre de 2021). *hiberus.* Recuperado el 27 de junio de 2023, de https://www.hiberus.com/crecemos-contigo/que-es-angular-y-para-que-sirve/

IBM. (2023). *¿Qué es PostgreSQL?* Recuperado el 26 de junio de 2023, de https://www.ibm.com/mx-es/topics/postgresql

Kinsta. (19 de 12 de 2022). *¿Qué es Express.js? Todo lo que Debes Saber.* Recuperado el 20 de 08 de 2023, de https://kinsta.com/es/base-de-conocimiento/que-es-express/

Mamani, M. L. (26 de noviembre de 2019). Recuperado el 27 de julio de 2023, de https://www.encora.com/es/blog/qu%C3%A9-es-vue.js-y-c%C3%B3mo-lo-usamos

Microsoft. (20 de 08 de 2023). *¿Qué es Java Spring Boot?* Obtenido de ¿Qué es Java Spring Boot?: https://azure.microsoft.com/es-mx/resources/cloud-computing-dictionary/what-is-java-spring-boot

Morcuende, S. (21 de octubre de 2022). *BETWEEN.* Recuperado el 27 de junio de 2023, de https://impulsate.between.tech/5-sistemas-control-versiones

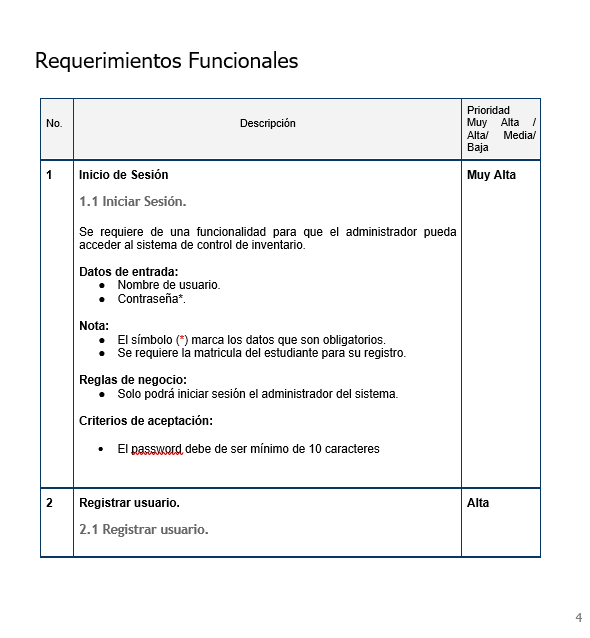
Palacios, I. N. (12 de abril de 2021). *medium.* Recuperado el 26 de junio de 2023, de https://kaldt-slange.medium.com/estilos-de-arquitectura-de-software-parte-i-6c5b60eb27b8

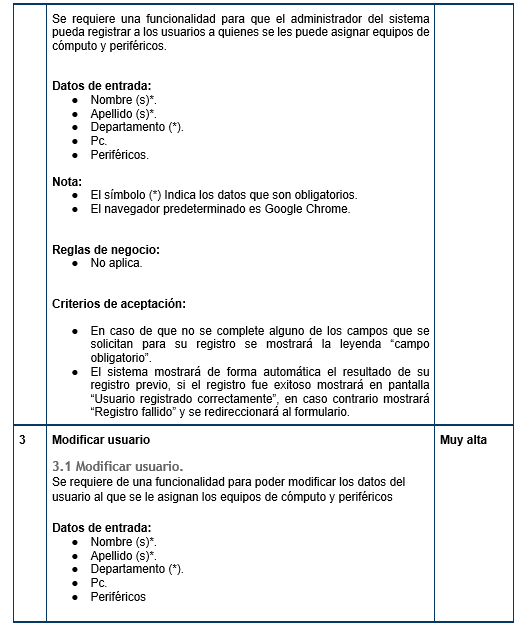
Puertas, J. P. (2014). *Creación de un portal con PHP y MySQL (4a. ed.).* RA-MA Editorial.

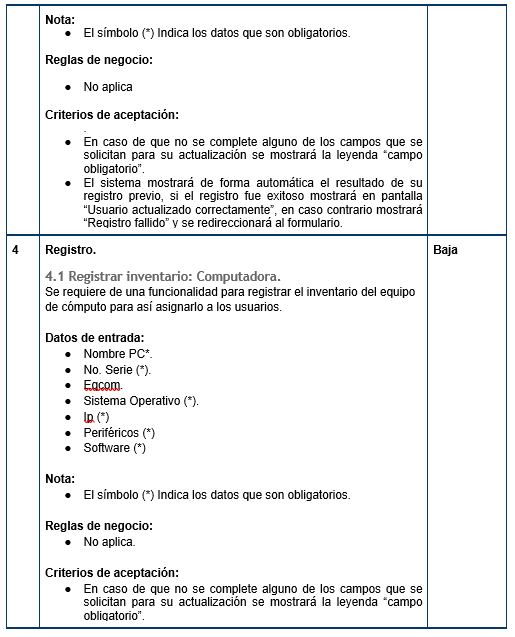
Santamaría, J. (2016). *Microsoft SQL Server. SQL SER vs MY SQL.* Recuperado el 26 de junio de 2023, de https://iessanvicente.com/colaboraciones/sqlserver.pdf

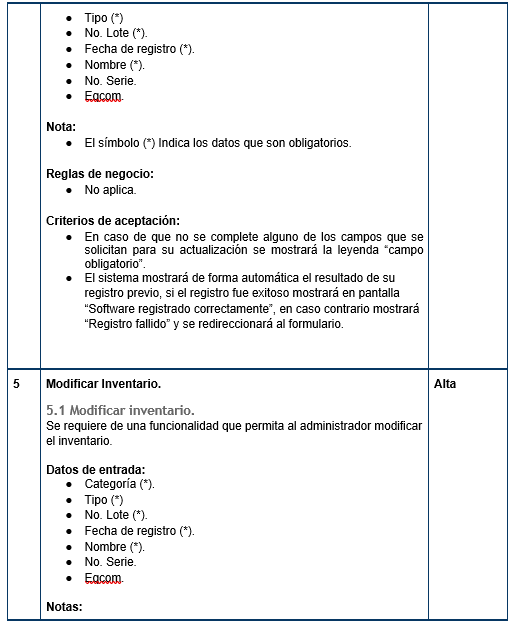
|  |
| --- |
| **ANEXOS** |

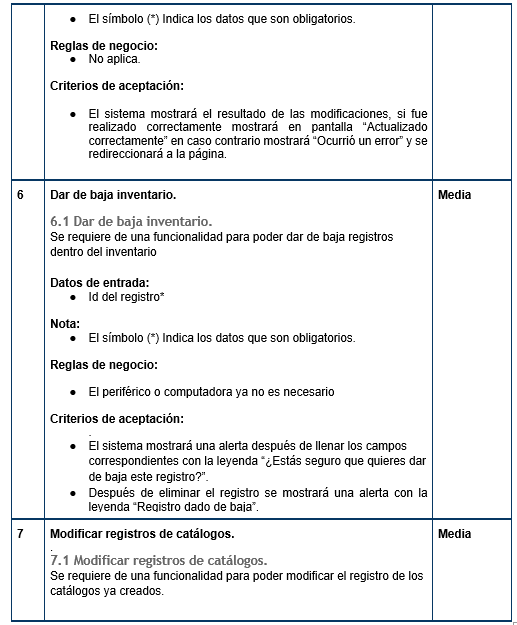
**ANEXO A. DOCUMENTO FORMAL DE REQUERIMIENTOS**

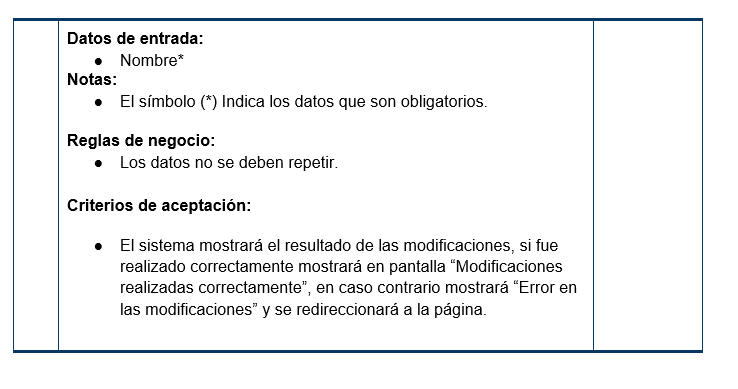
****





****

****

****

**ANEXO B. NOMBRE DEL ANEXO**