**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 |  | ASESORA UNIVERSITARIA |
| **Ing. Alejandro Bahena Bravo** |  | **Dra. Virna Viridiana Vela Rincón** |

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

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

Índice de figuras

Índice de tablas

Agradecimientos

Resumen

Summary

[CAPÍTULO 1. DESCRIPCIÓN DEL PROYECTO 8](#_Toc141277671)

[1.1 Datos generales de la empresa 8](#_Toc141277672)

[1.2 Antecedentes del proyecto 9](#_Toc141277673)

[1.3 Objetivo general 9](#_Toc141277674)

[1.4 Objetivos específicos 9](#_Toc141277675)

[1.5 Justificación 10](#_Toc141277676)

[1.6 Alcances 10](#_Toc141277677)

[1.7 Restricciones 11](#_Toc141277678)

[CAPÍTULO 2. MARCO DE REFERENCIA 12](#_Toc141277679)

[2.1 Conceptos, metodologías y herramientas 12](#_Toc141277680)

[2.1.1 Conceptos de arquitectura de software 12](#_Toc141277681)

[2.1.2 Motores de base de datos 15](#_Toc141277682)

[2.1.3 Tecnologías de backend 16](#_Toc141277683)

[2.1.4 Tecnologías de frontend 18](#_Toc141277684)

[2.1.5 Control de versiones 19](#_Toc141277685)

[2.2 Propuesta de solución 20](#_Toc141277686)

[CAPÍTULO 3. DESARROLLO 22](#_Toc141277687)

[3.1 Inicio 22](#_Toc141277688)

[3.2 Planeación 23](#_Toc141277689)

[3.2.1 Documento Formal de Requerimientos 23](#_Toc141277690)

[3.2.2 Plan de comunicación 24](#_Toc141277691)

[3.3 Ejecución 24](#_Toc141277692)

[3.3.1 Representación conceptual 24](#_Toc141277693)

[3.3.2 Diagrama de casos de uso 26](#_Toc141277694)

[3.3.3 Estructura del proyecto 27](#_Toc141277695)

[3.3.4 Creación de componentes. 29](#_Toc141277696)

[3.3.5 Diseño de vistas 30](#_Toc141277697)

[3.3.6 Base de datos 31](#_Toc141277698)

[3.4 Control 34](#_Toc141277699)

[3.5 Cierre 35](#_Toc141277700)

[CAPÍTULO 4. CONCLUSIONES 36](#_Toc141277701)

[4.1 Cumplimiento de objetivos 36](#_Toc141277702)

[4.2 Resultados 36](#_Toc141277703)

[4.3 Contribuciones 36](#_Toc141277704)

**REFERENCIAS**

**ANEXOS**

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

[Figura 1.1Logotipo de la empresa 8](#_Toc141686837)

[Figura 3.1 Representación conceptual 24](#_Toc141686838)

[Figura 3.2 Diagrama de Casos de Uso 26](#_Toc141686839)

[Figura 3.3 Estructura de carpetas 27](#_Toc141686840)

[Figura 3.4 Creación de un módulo 28](#_Toc141686841)

[Figura 3.5. Módulo de Auth 29](#_Toc141686842)

[Figura 3.6 Creación de un componente 29](#_Toc141686843)

[Figura 3.7 Componente de auth 30](#_Toc141686844)

[Figura 3.8 Diseño de vista login 30](#_Toc141686845)

[Figura 3.9 Listado de Inventarios 31](#_Toc141686846)

[Figura 3.10 Diagrama de la base de datos 32](#_Toc141686847)

|  |
| --- |
| **ÍNDICE DE TABLAS** |

[Tabla 3.1 Involucrados y sus funciones 14](#_Toc103865184)

|  |
| --- |
| **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** |

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

|  |
| --- |
| 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. Wurth 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

sistemas computacionales 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

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

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

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

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

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

6. Pruebas en la aplicación web.

7. 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 mal 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 son: 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 la misma.

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), 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.

**C#**

Según nos dice (Sierra, 2011), C# es actualmente junto con Java, uno de los lenguajes de programación más utilizados y populares entre los desarrolladores. Además, está disponible para el desarrollo de aplicaciones en general sin olvidar que este lenguaje tiene un alcance completo sobre la web. C# proporciona al programador el nivel de abstracción requerido para poder dar solución al desarrollo de cualquier aplicación por muy compleja que esta sea. Además de esto se trata de un lenguaje moderno orientado a objetos que permite desarrollar múltiples aplicaciones para la plataforma de Microsoft, .NET, la cual se caracteriza por proporcionar utilidades y servicios para un desarrollo óptimo.

**Java**

Java según nos dice (Sznajdleder, 2012) es un lenguaje de programación de alto nivel orientado a objetos, su creación fue debido a que se necesitaba alguna tecnología que permitiese el desarrollo de programas capaces de ejecutarse en entornos distribuidos y heterogéneos, es decir, programas ejecutables en muchas plataformas. Este lenguaje se desarrolló desde cero, pero su sintaxis es similar a la de C o C++, esto facilita la escritura del código a la hora de desarrollar ya que Java fue diseñado para desarrollar múltiples construcciones de software, como aplicaciones móviles, aplicaciones web, aplicaciones orientadas a servicios y más.

**Node js**

De acuerdo con (Lucas, 2019), Node js es un entorno de ejecución de JavaScript el cual nos permite utilizar JavaScript del lado del servidor, esto, nos permite desarrollar tecnología de backend para nuestras aplicaciones. Node se ejecuta en el motor del tiempo de ejecución de JavaScript V8, este motor toma el código de JavaScript y lo convierte en lenguaje máquina más rápido.

### 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. Es un gran sistema recomendado para Windows y proyectos que no requieran acceso distribuido.

**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 , 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 asi 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. Asi 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. La 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 computo y periféricos, asi 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, asi 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 Académico |
| 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 tomó principalmente 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 llevo 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 pequeñas 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.

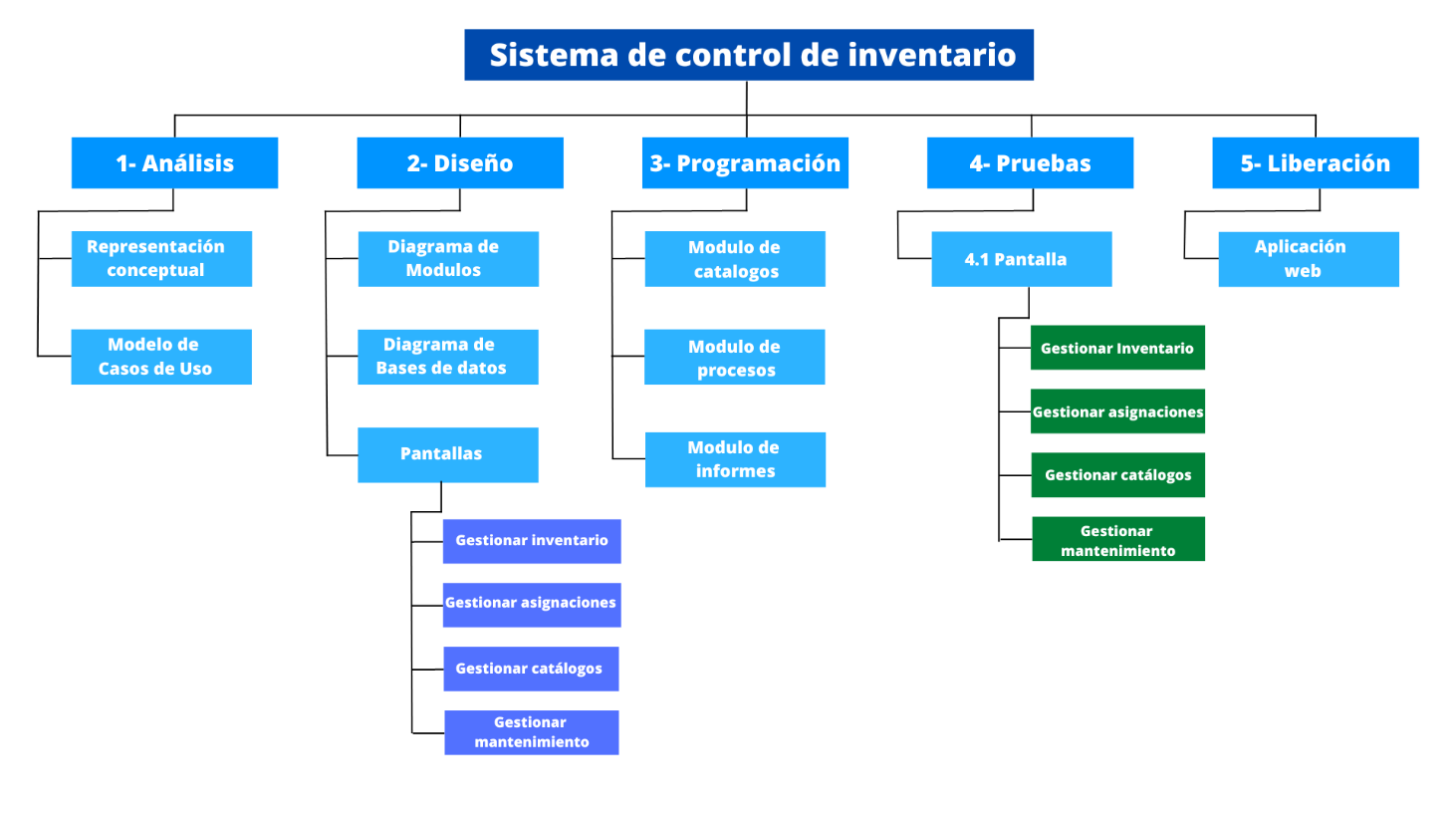
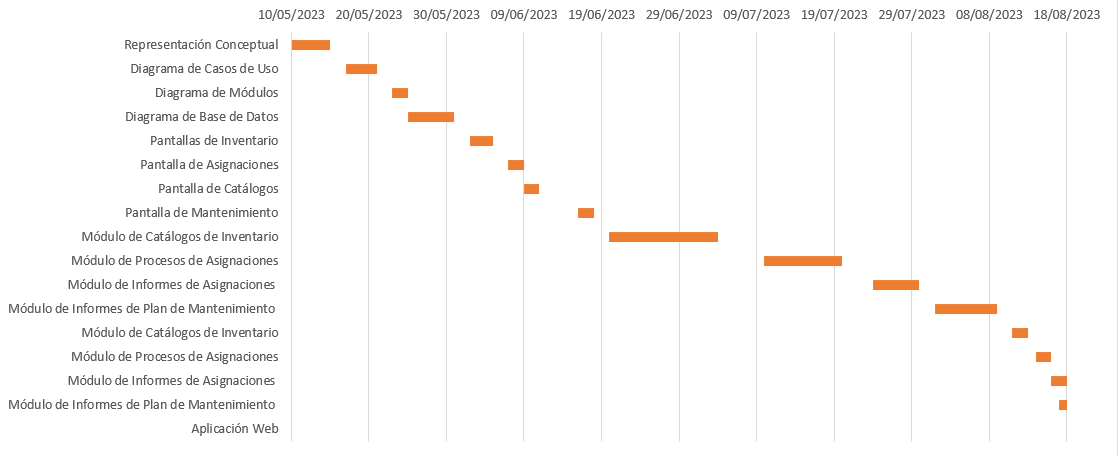


Figura 3.1 EDT

### 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 **sss** se muestra el diagrama de Gantt del proyecto, el cual plasma las actividades a realizar en las fechas definidas.



## Ejecución

### Representación conceptual

En la Figura 2.1 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 ultimo se enfoca en las funciones principales del sistema, asi como el comportamiento de este mismo.

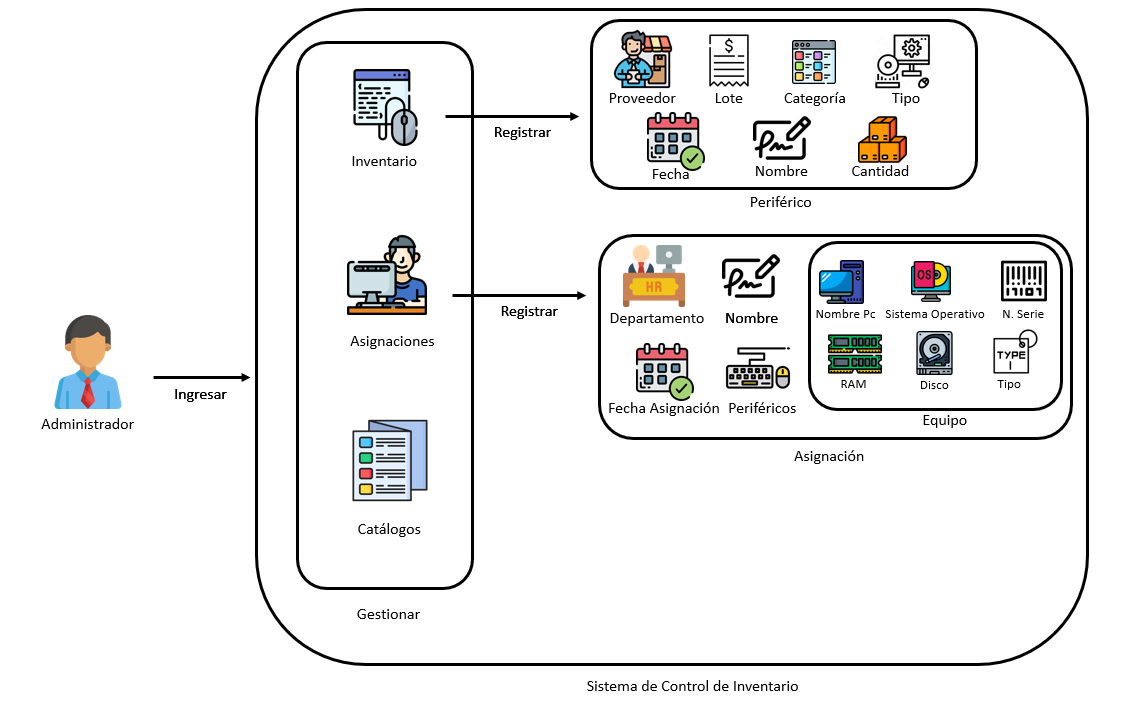


Figura 3.2 Representación conceptual

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 2.2 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.

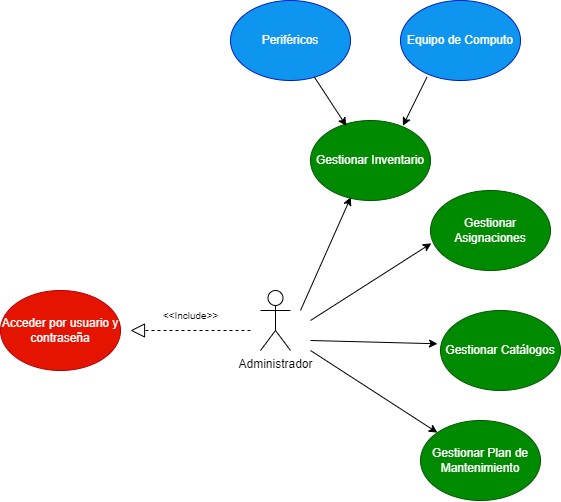


Figura 3.2 Diagrama de Casos de Uso

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 computo 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 mas organizada el desarrollo de nuestro proyecto con el fin de hacer un código más legible y organizado. En la Figura 3.3 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.

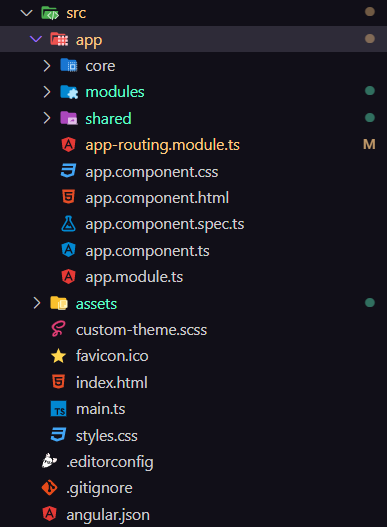


Figura 3.3 Estructura de carpetas

**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 3.4, se muestra el comando que nos ofrece la CLI (Command Line Interface) de Angular para crear módulos sin la necesidad de crearlos manualmente, para este ejemplo se crea el módulo de autenticación.

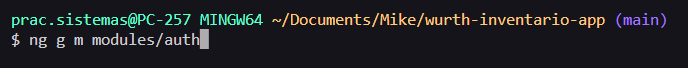


Figura 3.4 Creación de un módulo

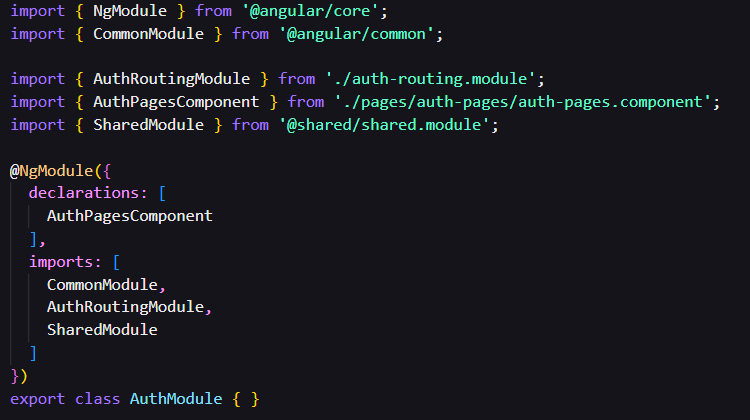


Figura 3.5. Módulo de Auth

Dentro de las importaciones se encuentra el módulo de Shared, el cual, es un modulo 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 3.5, se muestra un ejemplo de la creación de un componente, que en este caso, se genera el componente de autenticación.

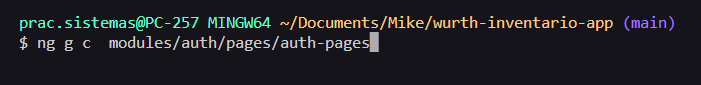


Figura 3.6 Creación de un componente

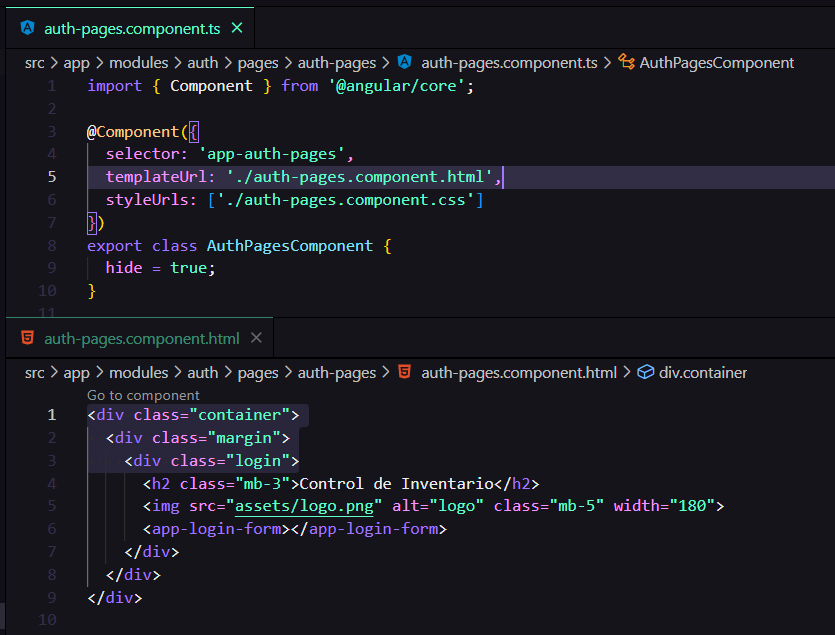


Figura 3.7 Componente de auth

### 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.

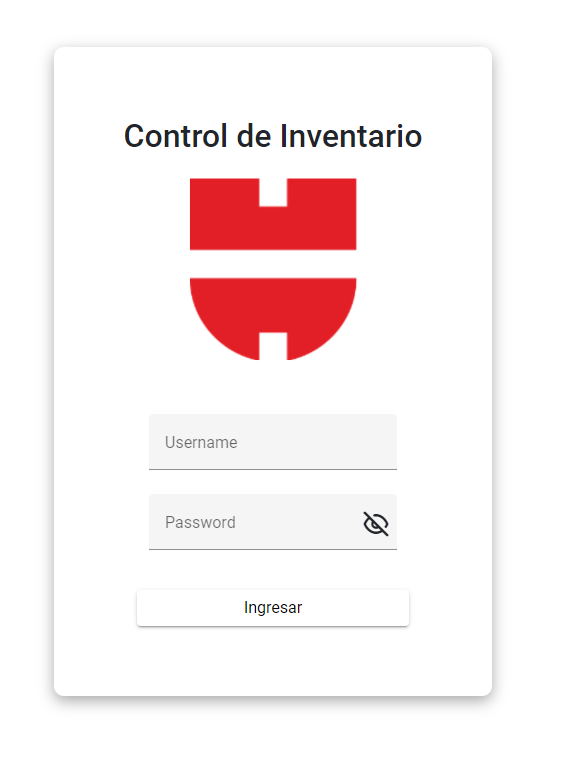
****

Figura 3.8 Diseño de vista login

**Vista de asignaciones**

En la Figura 3.8, 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 es con el fin de mantener un control sobre quienes tienen alguna asignación y asi poder dar seguimiento a estos mismos.

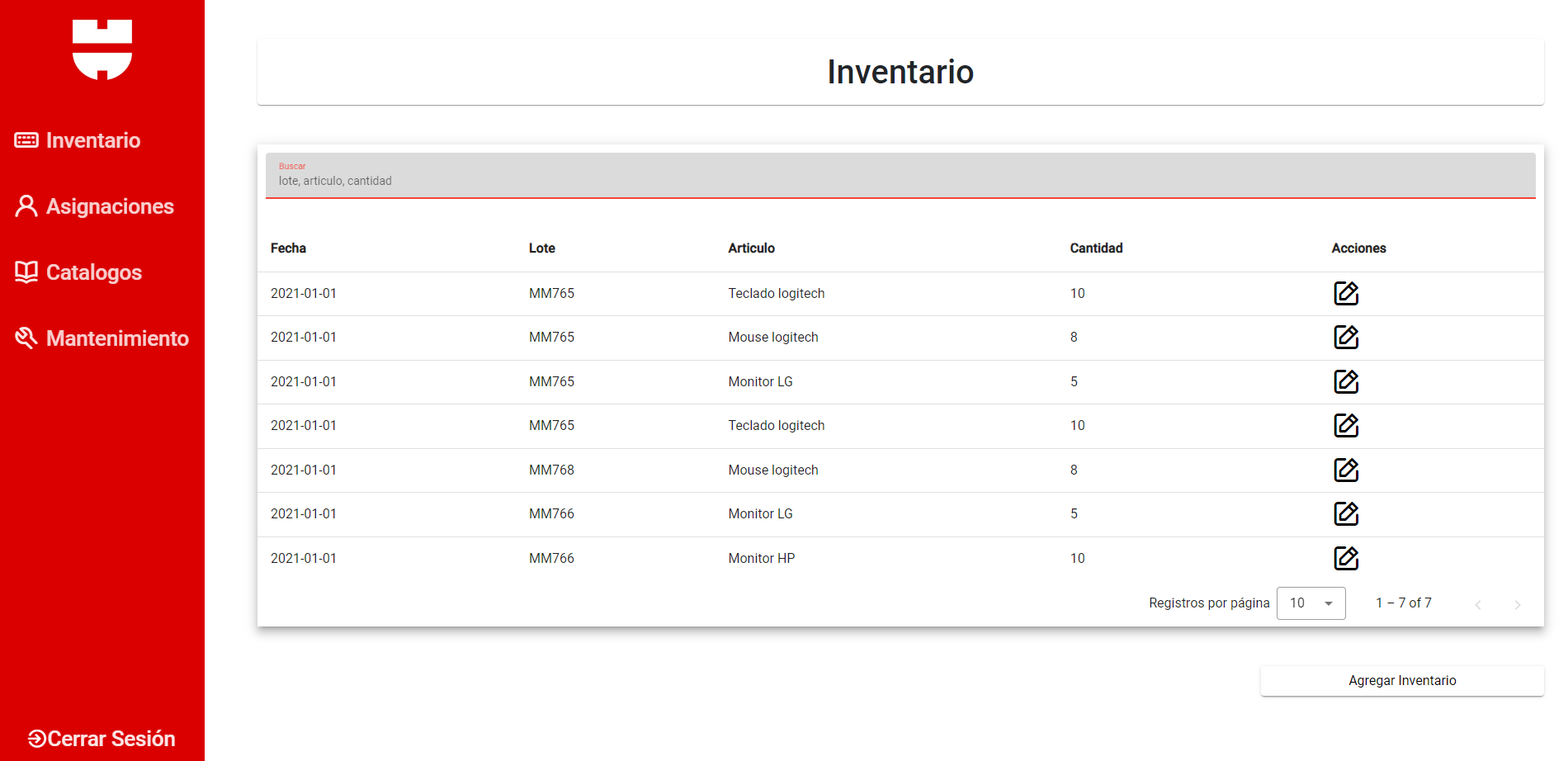


Figura 3.9 Listado de Inventarios

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, 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, asi también como el plan de mantenimiento de los equipos de computo con el fin de llevar un control no solo en la asignación del equipo de computo y de los periféricos, sino también en el seguimiento con respecto al mantenimiento.

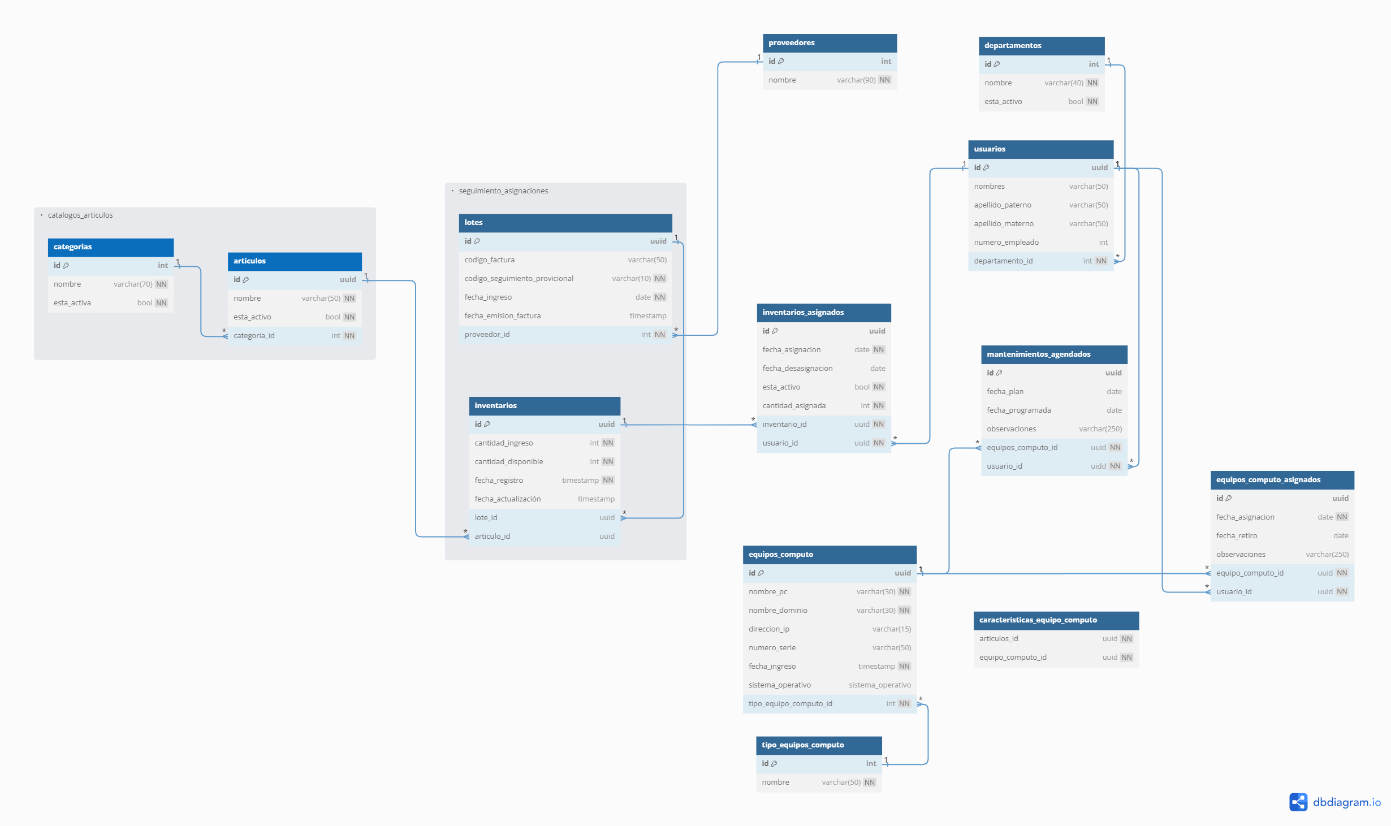


Figura 3.10 Diagrama de la base de datos

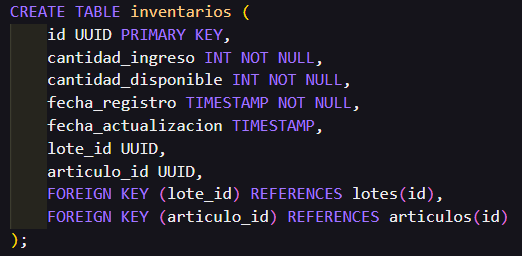
En la tabla de artículos se definen el nombre y la categoría de cada artículo, esta última se relaciona con la tabla de categorías la cual nos permite definir si el articulo que se registra pertenece a la categoría de Hardware o Software.

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

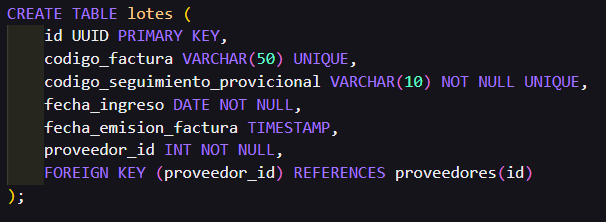
* Tabla de proveedores. En esta tabla se registran los proveedores con los cuales la empresa Würth México realiza la compra del inventario de Sistemas (periféricos, equipos de cómputo, servidores), para esto se registran los nombres de los proveedores.
* Tabla de lotes. Para hacer el registro del inventario 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 numero de lote de estos mismos. Esta tabla nos permite registrar y llevar el control de cada articulo 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, asi como la del inventario también nos permite llevar el registro de los equipos de computo 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 computo asignados. Lo que nos permite realizar esta tabla es realizar las asignaciones correspondientes a cada usuario dentro de la empresa, asi mismo, tener el control del equipo de computo que están asignados pudiendo ver la fecha en la se le asigna al usuario además de las observaciones.
* 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 en relación al plan de mantenimiento con el fin de contar con equipos de computo 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 es la que el usuario confirma para que se realice el mantenimiento.

**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 , se muestra la creación de una tabla que en este caso es la de Inventario, en la cual se persiste la información respecto al inventario con el que se cuenta dentro del departamento de Sistemas.



Como lo muestra la figura , 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 mas que en este caso con las de lotes y artículos. A continuación, se muestra el script de la tabla de lotes.



En la figura , 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, el principal objetivo de las bases de datos relacionales es mantener una organización estructurada de tal manera que la información se encuentre ordenada

### Servidor

Del lado del servidor se propone llevar una estructura, así como la estructura del lado del cliente, para esto se propone la siguiente solución.

**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 , se muestra el ejemplo de una clase que esta siendo definida con el nombre de la tabla, los atributos de esta, y la relación que tiene con la tabla de departamento.



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 , 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.

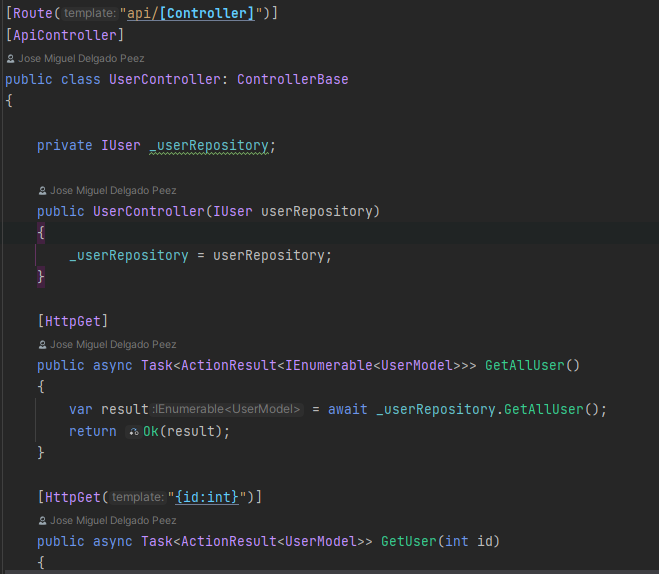


El objetivo de una interfaz, es crear una especie de contrato que con base en donde se defina, este caso el controlador, el mismo debe de implementar los métodos que la interfaz define.

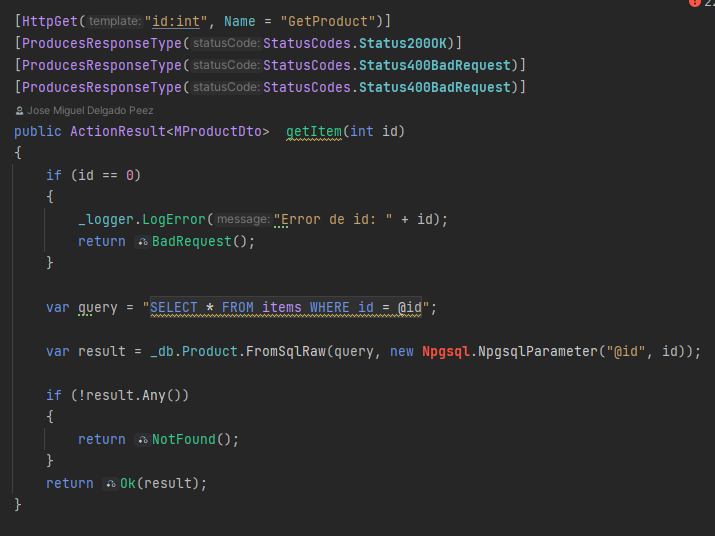
**Controllers**

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 , 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.



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

## Control

Para llevar un control durante el desarrollo del proyecto, se llevaron a cabo pequeñas reuniones con el jefe de Soporte el Ing. Jaime Rojas y con el Ing. Manuel Aldana, esto para buscar puntos de mejora en el proceso de las asignaciones, registro y consulta de los datos dentro del sistema, asi 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, asi 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 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.

## Resultados

Se deben describir los entregables finales, probados y en operación, así como los beneficios obtenidos por la empresa como resultado del proyecto.

## Contribuciones

Se deben describir las aportaciones a la empresa. Generalmente, no medibles. Por ejemplo: se utilizó una metodología nueva para el desarrollo. ¿Qué se utiliza ahora en la empresa que no se usaba antes?

De igual forma, se debe describir como benefició el proyecto a la formación profesional

del alumno. NO ESCRIBIR EN PRIMERA PERSONA. Al igual que el resto del documento, se debe escribir en tercera persona.

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

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

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

Sznajdleder, P. (2012). *JAVA A FONDO - Estudio del lenguaje y desarrollo de aplicaciones - 2ª Edición.* Alfaomega. Recuperado el 27 de junio de 2023

Sierra, J. C. (2011). *Microsoft C#. Curso de Programación. 2ª edición.* Grupo Editorial RA-MA. Recuperado el 26 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. (s.f.). *IBM.* Recuperado el 26 de junio de 2023, de https://www.ibm.com/mx-es/topics/postgresql

Lucas, J. (04 de septiembre de 2019). *OpenWebinars.* Recuperado el 27 de junio de 2023, de https://openwebinars.net/blog/que-es-nodejs/

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

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

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. NOMBRE DEL ANEXO**

Los anexos, además de tener una letra, deberán indicar un nombre para poder identificarlos. No son títulos, por lo que no aparecerán en el índice.

**ANEXO B. NOMBRE DEL ANEXO**

**Estilos del formato de reporte**

Este archivo incluye 5 estilos definidos para títulos y texto del contenido, que puedes encontrar en la banda **Inicio** de Word, y son:

* **Título 1**: para los títulos de capítulo.
* **Título 2**: para los títulos de segundo nivel. Por ejemplo: 1.2
* **Título 3**: para los títulos de tercer nivel. Por ejemplo: 1.2.1
* **Descripción**: para los títulos de figuras y tablas.
* **Normal**: para el texto del contenido.



Todo el texto del documento debe tener aplicado al menos uno de estos estilos, según corresponda.

Se deben utilizar los estilos para que, al actualizar los índices de contenido, se adopten adecuadamente los textos colocados como títulos y se indiquen las páginas donde se encuentran.

Los títulos del índice de contenido deben ser los únicos elementos numerados.

**Portada**

Toda la portada deberá estar escrita en mayúsculas. Se debe elegir la palabra que defina correctamente a los asesores empresarial y universitario (asesor o asesora, universitario o universitaria); es decir, el texto colocado en rojo deberá sustituirse según corresponda.



Se debe colocar correctamente el nombre del programa educativo, considerando dos escenarios:

* Los nombres de programas que no tienen área (Ingenierías) se colocan en tamaño 16.
* Los nombres de programas que tienen área (TSU) deben seguir la regla anterior y, además, colocar el área en tamaño 14.

Es necesario que la portada tenga centrados tanto el nombre del proyecto como el grado. Esto implica que no solo debe estar centrado horizontalmente, sino también verticalmente en el espacio que corresponde a cada texto (área blanca para el nombre del proyecto y área azul para grado).

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

El lugar y fecha para el reporte se encuentra en la parte inferior de la portada. Se debe verificar que el mes que se indica sea un mes posterior a la finalización de la estadía.



**Formato del documento**

Se deberá respetar el formato del documento actual. Si por algún motivo se llegase a modificar, es necesario revisar los siguiente:

* Los márgenes del documento serán de 3cm. a la izquierda, 2.5 cm. a la derecha, superior e inferior.
* No se deberá incluir ninguna leyenda (ni líneas, ni títulos de trabajos, ni nombre del alumno) de encabezado.
* El pie de página solo deberá contener el número de la página fuente arial 10, alineado a la derecha.

**Cantidad de páginas**

* Las páginas del documento empiezan a contar después de la portada, que se considera la página 0, pero el número de página se hace visible hasta el Capítulo 1.
* La cantidad de páginas que deberá contener el reporte de estadía será de 40 como mínimo, las cuáles **se empiezan a contar a partir de la** **primera página del Capítulo 1** y **hasta la última página del capítulo 4**. No se consideran portada, índices, agradecimientos, resumen, referencias o anexos.
* Por ejemplo: un reporte que **inicia en la página 8** deberá **termina**r, al menos, **en la página 47**.

**Saltos de línea**

Se deberán considerar las siguientes reglas para los saltos de línea o renglones vacíos entre párrafos, títulos, imágenes y tablas:

* Se coloca un salto de línea entre título y párrafo; y entre párrafo y párrafo.
* Las imágenes y las tablas se consideran parte del texto o un párrafo. Si una imagen se encuentra con una tabla, o viceversa, se coloca un salto de línea entre ellas.
* Se colocarán dos saltos de línea entre párrafo y título siguiente.

Cuando un título queda solo al final de una página, existen dos opciones:

* Si el espacio restante es un salto de línea, se puede insertar un salto de página antes del título para moverlo a la siguiente página.
* Si el espacio restante es más de un salto de línea, se debe complementar la redacción del texto previo para evitar que el título quede solo al final de la página.

****

**Redacción y ortografía**

El documento debe estar escrito en tercera persona y evitar el uso del verbo ser seguido de un adjetivo. Por ejemplo:

* **No** **escribir**: Fue necesario, fue requerido, fue desarrollado, etc.
* **Utilizar**: Se necesitó, se requirió, se desarrolló.

Se debe evitar el uso de la palabra nosotros. Por ejemplo:

* **No escribir**: Tuvimos, realizamos, obtendremos, etc.
* **Utilizar**: Se tuvo, se realizó, se obtendrá, etc.

Es necesario revisar acentuación del documento (F7 con Word) y prestar especial atención en verbos escritos en pasado y futuro.

Se debe revisar que el texto este redactado procurando utilizar enunciados cortos. Generalmente se hace un uso excesivo de comas en lugares erróneos. Para verificar la necesidad de un punto, solicitar:

***Lee los párrafos en voz alta. Si durante la lectura te falta el aire para terminar, es posible evidencia de que requieres un punto o punto y coma.***

**Numeración y viñetas**

Se prefieren las viñetas sobre la numeración, salvo cuando sea necesario indicar un orden. **NO** se deben colocar saltos de línea entre numeración o viñetas. **Esto se debe observar a detalle, ya que se abusa de ello** con el fin de obtener hojas adicionales.

Se debe colocar un salto de línea antes de iniciar el listado con las viñetas.

**Referencias**

Las referencias se deben incluir en el texto y, en la medida de lo posible, seguir el formato APA utilizado por Microsoft Word. Para ello, se deben utilizar las herramientas para la gestión de referencias que proporciona el editor.

Se deberá evitar el plagio. Evitar el texto copiado y pegado. Si no hubiese otra opción, referenciarlo correctamente. Es preferible parafrasear o escribir con palabras propias.

**Referencia parafraseada:** comunica el mensaje del autor original con palabras propias y conservando el sentido en cuanto a organización, continuidad y coherencia. Generalmente, se inicia con una frase introductoria (así como, por tanto, de acuerdo con, según, etc.)

*De acuerdo con (Salazar, 2018), un programa computacional aplicado (PCA) se encarga de resolver un problema trivial en el campo...*

**Referencia textual:** es una cita fiel del texto original de un autor, palabra por palabra. El texto debe ponerse entre comillas, acompañados de la referencia.

*Para comprender los ejes corporales del cuerpo humano, "se han concebido planos imaginarios de referencia que atraviesan el cuerpo de forma que unos son perpendiculares a otros" (Palastanga, Field & Soames, 2000).*

**Las referencias deben tomarse de sitios confiables** y debe priorizarse la referencia de elementos obtenidos de libros, enciclopedias, artículos científicos, artículos de revistas académicas, entre otros.

Los documentos o sitios Web deben ser confiables, de entidades reconocidas como expertos o de renombre en el área.

Si se trata de una **entrada de blog**, debe tratarse de una persona reconocida en el campo (experto, no necesariamente popular). Por ejemplo: Richard Stallman, Bruno Munari, entre otros.

**Imágenes y tablas**

Si se añade una imagen es porque:

* Ayuda a la compresión del texto y debe describirse.
* El lector puede distinguir el texto y formas que incluye la imagen.
* Es necesaria y no se repite en algún otro lugar del documento.

El **tamaño de las imágenes** en el reporte quedará **a consideración del asesor universitario**, mismo que validará que sean necesarias y no se abuse de ellas para ocupar espacio.

Es válido colocar dos imágenes seguidas siempre que sea necesario para mantener el correcto formato del documento. **Se debe verificar que no se abuse de esto.**

Las imágenes y tablas NO deben colocarse en cuadros de texto o con diseño flotante (que pueda arrastrarse con el mouse sobre el documento). Las imágenes deben colocarse en línea con el texto.

Las imágenes deben estar referenciadas en el texto. Si se añade una imagen es porque ayuda a la compresión del texto y debe describirse. Por ejemplo, para hacer referencia a la figura con título Figura 2.1 en el texto descriptivo, se deberá hacer como:

“… en la Figura 2.1 se puede observar el funcionamiento paso a paso del algoritmo de búsqueda Bubble Sort para un arreglo de datos numéricos aleatorios.”

El rótulo y número, en el título de las imágenes, debe colocarse en texto normal (sin cursivas o negritas). Por ejemplo: Figura 2.1 Servidor de base de datos. El título de las tablas se coloca en la parte superior del encabezado, también en texto normal (sin cursivas o negritas).

Los títulos de imágenes o tablas se deben colocar sin punto.

Si una imagen no fue de creación propia, es necesario agregar la referencia al título de las imágenes. Por ejemplo:

Gráfico, Gráfico de líneas

Descripción generada automáticamente

No debe existir algún salto de línea entre la imagen y su título de imagen. Deben cortarse los espacios en blanco alrededor de la imagen.



Cuando un título queda solo al final de una página, existen dos opciones:

* Si el espacio restante es un salto de línea, se puede insertar un salto de página antes del título para moverlo a la siguiente página.
* Si el espacio restante es más de un salto de línea, se debe complementar la redacción del texto previo para evitar que el título quede solo al final de la página. Por ejemplo:

