

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE COACALCO

Sistema de información web con C# y
SQL Server para el desarrollo de
módulos ERP para la empresa
Mercurio Comunicación S.A. de C.V.

Titulación

Para obtener el título de Ingeniero en Sistemas
Computacionales

Especialidad
Ingeniería de Software

Presenta:
Luis Angel Valencia Arcega

Calificación: _____

No seré brillante, hasta no ser consciente de mi propia ignorancia, por ello le dedico este trabajo a mi yo del pasado, al que se apasionó por aprender algo nuevo cada día y tener la disciplina de seguirlo. Les dedico un pedazo de mi memoria, a todas aquellas personas que han confiado en mí, nombrarlas, sería olvidar a algunas.

RESUMEN

La labor en la empresa Mercurio se concentra en el desarrollo de módulos de su ERP que son fundamentales: el Customer Relationship Management (CRM) y el módulo de Gestión de Inventario. Mercurio, como entidad intermediaria, juega un papel esencial en la comunicación de campañas entre laboratorios farmacéuticos internacionales y pacientes.

El módulo CRM está destinado a la gestión integral de los clientes de la empresa, implicando un registro minucioso de las interacciones con los laboratorios y pacientes, así como la facilitación de la coordinación de campañas farmacéuticas.

Por su parte, el módulo de Gestión de Inventario adquiere una relevancia primordial, ya que se encuentra intrínsecamente relacionado con el CRM. Su objetivo principal es mantener un control meticuloso de los productos farmacéuticos que son remitidos por los laboratorios a la empresa, garantizando su disponibilidad para su posterior distribución a los pacientes.

El proyecto se enfoca en el desarrollo de módulos críticos para la empresa. Su finalidad es mejorar la eficiencia y la precisión en la comunicación de campañas entre los laboratorios farmacéuticos y los pacientes, garantizando un control riguroso de los productos farmacéuticos y una gestión efectiva de las relaciones con los clientes gracias a su inventario.

ÍNDICE

Generalidades Del Proyecto.....	9
1.1 Introducción	10
1.2 Descripción De La Empresa.....	12
1.3 Problema A Resolver	14
1.4 Objetivos	15
1.5 Justificación	16
Marco Teórico.....	17
2.1 ERP.....	18
2.2 CRM.....	18
2.3 Estado Del Arte	19
2.4 Metodologías De Desarrollo De Software.....	24
2.5 Herramientas De Software.....	26
<i>2.5.1 SQL Server Management Studio (SSMS)</i>	<i>26</i>
<i>2.5.2 Visual Studio</i>	<i>26</i>
<i>2.5.3 Playwright E2E</i>	<i>27</i>
2.6 Diferencias Entre Front-End Y Backend	27
2.7 Lenguajes De Front-End	28
<i>2.7.1 JavaScript.....</i>	<i>28</i>
<i>2.7.2 CSS.....</i>	<i>29</i>
<i>2.7.3 HTML</i>	<i>29</i>
<i>2.7.4 ASP.NET.....</i>	<i>30</i>
2.8 Lenguajes De Backend Y Bases De Datos.....	30
<i>2.8.1 C#.....</i>	<i>30</i>

2.8.2 <i>SQL</i>	31
2.9 Patrones De Diseño	33
2.10 MVC	34
2.11 Arquitecturas De Software	35
2.12 Diseño De La Arquitectura	35
3.1 Análisis	39
3.1.1 <i>Atributos De Calidad</i>	39
3.2 Diseño	41
3.2.1 <i>Diagramas UML</i>	42
3.3 Codificación	44
3.4 Pruebas	50
Resultados	51
4.1 Seguimiento De Versiones	52
4.2 Análisis, Diseño Y Modelado De La Estructura	53
4.2.1 <i>Product Backlog</i>	53
4.2.2 <i>Sprint Backlog</i>	53
4.2.3 <i>Burndown Chart</i>	56
4.2.4 <i>Definition Of Done</i>	57
4.3 Codificación Del Módulo De Inventario	57
4.3.1 <i>Product Backlog</i>	57
4.3.2 <i>Sprint Backlog</i>	58
4.3.3 <i>Burndown Chart</i>	62
4.3.4 <i>Definition Of Done</i>	63
4.4 Codificación Del Módulo De CRM	63
4.4.1 <i>Product Backlog</i>	63

4.4.2 <i>Sprint Backlog</i>	64
4.4.3 <i>Burndown Chart</i>	68
4.4.4 <i>Definition Of Done</i>	68
4.5 Pruebas, Implementación Y Capacitación	69
4.5.1 <i>Product Backlog</i>	69
4.5.2 <i>Sprint Backlog</i>	69
4.5.3 <i>Burndown Chart</i>	72
4.5.4 <i>Definition Of Done</i>	72
4.6 Interfaces De Usuario	73
4.7 Comparación De Procesos	89
Conclusiones	91
Competencias Desarrolladas	93
Fuentes De Información	96
Anexos	99
Contrato De Confidencialidad	99

ÍNDICE DE FIGURAS

Figura 1. Organigrama de la empresa.....	12
Figura 2. Esquema del patrón MVC.....	34
Figura 3. Esquema de arquitectura del sistema.	36
Figura 4. Diagrama de flujo inicio de sesión.....	41
Figura 5. Diagrama de secuencia inicio de sesión.....	42
Figura 6. Diagrama de caso de uso general.	43
Figura 7. Diagrama de caso de uso inicio de sesión.....	43
Figura 8. Diagrama de secuencia de operaciones al inventario.....	44
Figura 9. Pseudocódigo del controlador.	48
Figura 10. Pseudocódigo de clase conexión DAL.....	49
Figura 11. Burndown chart Días por sprint.	56
Figura 12. Burndown chart Días por sprint.	63
Figura 13. Burndown chart Días por sprint.	68
Figura 14. Burndown chart Días por sprint.	72
Figura 15. Interfaz de inicio de sesión.....	73
Figura 16. Interfaz para enviar código de recuperación.	74
Figura 17. Alerta de código enviado al usuario.....	74
Figura 18. Bandeja de correo con el código de recuperación.....	75
Figura 19. Alerta de código valido y campo de nueva contraseña.	75
Figura 20. Alerta contraseña restablecida exitosamente.	76
Figura 21. Interfaz de pantalla principal, gestor de inventario.....	76
Figura 22. Interfaz de pantalla principal, gestor de CRM.	77
Figura 23. Interfaz de dashboard inventario.....	78
Figura 24. Interfaz de productos.....	79
Figura 25. Interfaz de detalle de producto.....	79
Figura 26. Interfaz de alta de producto.....	80
Figura 27. Interfaz de edición de productos.	81
Figura 28. Interfaz de dashboard inventario - hover en almacén.	81
Figura 29. Interfaz de existencias por almacén.	82
Figura 30. Interfaz de dashboard principal gestor de CRM.	82

Figura 31. Interfaz de alta de paciente información básica.	83
Figura 32. Interfaz de alta de paciente información de clínica.....	84
Figura 33. Interfaz de alta de paciente información clínica de campañas.....	85
Figura 34. Interfaz de alta de paciente información médica.....	85
Figura 35. Interfaz de alta de paciente información de estudio.....	86
Figura 36. Interfaz de médico especialista.	87
Figura 37. Interfaz de campañas.....	88
Figura 38. Comparación de días en operaciones generales pre sistema y post sistema.	89
Figura 39. Comparación de horas en operaciones generales pre sistema y post sistema.	90

ÍNDICE DE TABLAS

Tabla 1. Estado del arte	20
Tabla 2. Atributos de calidad.....	40
Tabla 3. Seguimiento de versiones del sistema	52
Tabla 4. Spring backlog 1 del Análisis, diseño y modelado de la estructura	54
Tabla 5. Spring backlog 2 del Análisis, diseño y modelado de la estructura	54
Tabla 6. Spring backlog 3 del Análisis, diseño y modelado de la estructura	55
Tabla 7. Spring backlog 4 del Análisis, diseño y modelado de la estructura	55
Tabla 8. Spring backlog 1 de Codificación del módulo de inventario	58
Tabla 9. Spring backlog 2 de Codificación del módulo de inventario	59
Tabla 10. Spring backlog 3 de Codificación del módulo de inventario	60
Tabla 11. Spring backlog 4 de Codificación del módulo de inventario	62
Tabla 12. Spring backlog 1 de Codificación del módulo de CRM.....	64
Tabla 13. Spring backlog 2 de Codificación del módulo de CRM.....	65
Tabla 14. Spring backlog 3 de Codificación del módulo de CRM.....	66
Tabla 15. Spring backlog 4 de Codificación del módulo de CRM.....	67
Tabla 16. Spring backlog 1 de Pruebas, implementación y capacitación.....	69
Tabla 17. Spring backlog 2 de Pruebas, implementación y capacitación.....	70
Tabla 18. Spring backlog 3 de Pruebas, implementación y capacitación.....	71
Tabla 19. Spring backlog 4 de Pruebas, implementación y capacitación.....	71
Tabla 20. Competencias desarrolladas	93

Capítulo I

Generalidades Del Proyecto

1.1 Introducción

El presente reporte de proyecto se centra en un enfoque específico del desarrollo y administración de módulos del sistema de planificación de recursos empresariales (ERP) como el Customer Relationship Management (CRM), que se dedica a la gestión de relaciones con clientes. Y el módulo que prioriza la administración del inventario de la empresa Mercurio Comunicación S.A. de C.V. El cual guarda una estrecha relación e importancia con la gestión de sus clientes.

La relevancia de este trabajo reside en la creciente importancia de los sistemas de información en el mundo empresarial actual. Los sistemas ERP desempeñan un papel fundamental en la optimización de procesos y la toma de decisiones estratégicas. Específicamente, el CRM es esencial para la gestión efectiva de relaciones con los clientes y su relación con el inventario interno y el intercambio de artículos, lo que influye directamente en la satisfacción y el éxito empresarial. Este trabajo tiene un doble propósito: aporta valor académico al explorar tecnologías de vanguardia como .NET, el lenguaje C# y SQL Server en un contexto empresarial y al mismo tiempo, ofrece una solución concreta a los desafíos que enfrenta la empresa en la gestión de clientes y su inventario, lo que tiene un valor profesional significativo.

Este trabajo se desarrolla siguiendo una metodología que involucra la revisión exhaustiva del marco teórico relacionado con los ERP, los CRM y la gestión de inventario. Además, se incluye un análisis detallado de las características específicas de la empresa, sus necesidades y objetivos. La metodología se enfoca en el diseño, desarrollo e implementación del módulo de gestión de inventario en la empresa, utilizando tecnologías como .NET y SQL Server. Se llevará a cabo un seguimiento constante del proceso, y se recopilarán datos y resultados para su posterior análisis.

Es importante destacar que este trabajo tiene ciertas limitaciones, como la disponibilidad de recursos, el acceso a datos específicos de la empresa debido al documento de confidencialidad firmado al momento de la asignación de proyecto.

La estructura de este trabajo se divide en los siguientes capítulos:

1. Generalidades del Proyecto: En este capítulo se presentarán los antecedentes de la empresa, sus necesidades, objetivos y la relevancia de la implementación de un CRM en su contexto.
2. Marco Teórico: Aquí se analizarán conceptos clave relacionados con un ERP y las tecnologías, proporcionando una base teórica sólida para comprender el trabajo.
3. Desarrollo: Este capítulo abordará la planificación, diseño, desarrollo e implementación de módulos como Gestión de inventario y CRM en la empresa, detallando la tecnología utilizada y los pasos clave del proceso.
4. Resultados y Conclusiones: Se presentarán los resultados obtenidos a lo largo del proyecto, incluyendo la evaluación de la implementación, los beneficios logrados y las lecciones aprendidas.

1.2 Descripción De La Empresa

Nombre, razón social

Mercurio Comunicación S.A. de C.V.

Ubicación

Av. Universidad número 1379, Col. Axotla Código Postal 01030, CDMX

Organigrama

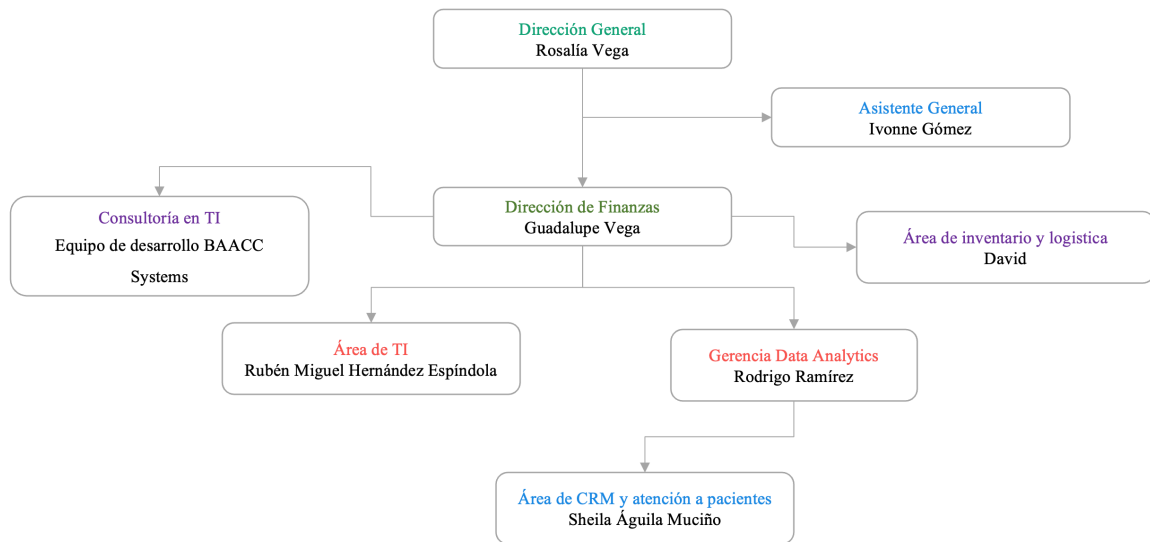


Figura 1. Organigrama de la empresa. Fuente: Mercurio (2023)

Sector productivo de la empresa

Servicios de información

Misión, Visión, Filosofía y/o Valores

(Mercurio, 2023)

Misión: Ser una empresa de comunicación altamente especializada en el manejo de la información médica.

Visión: Generar proyectos informativos hechos a la medida de las necesidades promocionales y publicitarias de nuestros clientes.

Valores (Mercurio, 2023):

- Constancia. Fuerza de voluntad y esfuerzo continuo, aplicado hasta llegar a la meta propuesta y cumplir el propósito. No rendirse y contagiar esta actitud al resto del equipo.
- Puntualidad. Cumplir los compromisos adquiridos en forma obligada. Saber gestionar y coordinar el tiempo como prioridad para atender a los clientes, los colegas y los proveedores.
- Precisión. Saber lo que se hace y el resultado que se obtendrá de ello. Conocer el punto al que se quiere llegar. Definir concisamente lo que se está diciendo.

Breve historia de la empresa

En mayo de 1988, la entidad comenzó sus operaciones como una editorial médica especializada en la creación de proyectos personalizados para respaldar campañas publicitarias de medicamentos recientemente lanzados. A lo largo de 35 años de dedicación y esfuerzo continuo, la organización ha ampliado su gama de servicios. Además de la producción editorial impresa y digital, ahora ofrece servicios publicitarios y de formación médica, operación de centros de atención telefónica, creación de artículos promocionales, cobertura de eventos y servicios de edición de video (Mercurio, 2023).

1.3 Problema A Resolver

En la empresa Mercurio, que funge como un intermediario esencial entre laboratorios farmacéuticos y pacientes, se presenta una problemática crítica relacionada con la gestión de inventario y la comunicación de campañas. La empresa opera en un entorno altamente regulado y competitivo, donde la precisión y eficiencia son fundamentales para su éxito en el mercado farmacéutico.

La necesidad imperante en este contexto es la optimización de la gestión de inventario y la comunicación efectiva entre laboratorios y pacientes. La propuesta es la implementación de un sistema de información web para la correcta gestión del inventario y los clientes para lograr una sincronía de calidad.

El sistema de gestión de inventario deberá mantener un registro preciso y en tiempo real de los productos farmacéuticos proporcionados por los laboratorios (CRM), así como su disponibilidad para su inclusión en campañas. Esto permitirá una distribución más precisa de los productos a los pacientes, reduciendo el riesgo de agotamiento de existencias o la acumulación innecesaria de productos farmacéuticos en el almacén de la empresa.

La no implementación de esta propuesta conlleva una serie de consecuencias negativas tanto para la empresa como para los laboratorios y los pacientes:

- Ineficiencia operativa: La falta de un sistema de gestión de inventario podría resultar en la pérdida de productos farmacéuticos debido a su caducidad o en la indisponibilidad de productos críticos en campañas.
- Dificultades en la comunicación: La ausencia de mejoras en el módulo CRM limitaría la capacidad de la empresa para mantener relaciones sólidas con los laboratorios y brindar un servicio personalizado a los pacientes.
- Riesgo de incumplimiento regulatorio: En el sector farmacéutico, el incumplimiento de normativas y regulaciones puede tener graves consecuencias legales y financieras.

1.4 Objetivos

Objetivo general

Desarrollar un sistema de información web utilizando C# y SQL Server que permita la implementación eficiente de módulos como la Gestión de Inventario y CRM en un ERP destinado a la empresa Mercurio Comunicación S.A. de C.V., con el propósito de optimizar la gestión de inventario, fortalecer las relaciones con clientes y mejorar la operatividad y competitividad de la empresa en el sector farmacéutico.

Objetivos específicos

1. Desarrollar un módulo de Gestión de Inventario en el sistema de información web con C# y SQL Server, que permita la identificación, registro y seguimiento de productos farmacéuticos recibidos de laboratorios, asegurando su disponibilidad en tiempo real y evitando pérdidas en el inventario de la empresa.
2. Implementar un módulo de CRM en el sistema de información web, orientado a la gestión eficiente de las relaciones con los clientes de la empresa o laboratorios farmacéuticos, mediante un seguimiento detallado de interacciones y la personalización de servicios.
3. Integrar los módulos de Gestión de Inventario y CRM dentro del ERP de la empresa, garantizando una interacción fluida y sincronizada entre ambas áreas para optimizar la planificación y ejecución de campañas farmacéuticas.

1.5 Justificación

La presente investigación aborda la imperante necesidad de desarrollar un sistema de información web basado en tecnologías C# y SQL Server con el propósito de implementar módulos de Gestión de Inventario y CRM en el ERP de la empresa. Esta justificación se fundamenta en diversos criterios de alta relevancia. La conveniencia de este proyecto se sustenta en la creciente complejidad inherente a la gestión de inventario y la administración de relaciones con clientes en el sector farmacéutico. La operación actual de la empresa enfrenta carencias operativas y falta de sincronización en sus procesos, lo que conlleva ineficiencias y errores.

La automatización y la coherencia de estos procesos se erigen como imperativos en un entorno empresarial que demanda precisión y cumplimiento normativo. La relevancia social de esta investigación radica en el papel crítico desempeñado por la empresa en la facilitación de la comunicación y distribución de productos farmacéuticos que directamente impactan en la salud y el bienestar de los pacientes. La optimización de la gestión de inventario y la mejora de las relaciones con los clientes, en particular laboratorios farmacéuticos, conlleva un efecto positivo en la disponibilidad y entrega puntual de productos de significativa importancia para la salud pública.

La metodología aplicada en este proyecto para el desarrollo de los módulos y su integración en el ERP posee utilidad metodológica en términos de su replicabilidad. La utilización de tecnologías como C# y SQL Server en este contexto puede servir como modelo y guía para empresas que enfrentan desafíos similares en la gestión de inventario y la mejora de las relaciones con los clientes. Este enfoque metodológico se presta a una potencial adopción en distintos sectores industriales.

Capítulo II

Marco Teórico

2.1 ERP

De acuerdo con Díaz (2005), Enterprise Resource Planning (Planificación de Recursos Empresariales) es un sistema integral, diseñado para coordinar y optimizar las operaciones y los procesos de negocio en una organización. Este sistema proporciona una plataforma unificada para gestionar diversas funciones, integrando datos y procesos en una única base de datos centralizada.

La característica fundamental de un ERP radica en su capacidad para facilitar la colaboración interdepartamental, permitiendo a diferentes secciones de la empresa compartir información en tiempo real. Esto conduce a una mayor eficiencia operativa, reducción de costos y mejora en la toma de decisiones a través del acceso a datos precisos y actualizados.

Algunos de los módulos que pueden integrarlo son:

- Recursos humanos
- Contabilidad
- Control de almacén o inventario
- Clientes (CRM)
- CX/UX y Gestión de proyectos
- Atención al cliente

2.2 CRM

Desde el punto de vista de (Fairhurst, 2001) se trata de una estrategia empresarial orientada a la gestión y optimización de las interacciones con los clientes a lo largo de todo el ciclo de vida de la relación comercial. Implica la utilización de tecnologías, estrategias y prácticas para analizar y administrar las interacciones entre una empresa y sus clientes actuales y potenciales. Su objetivo principal radica en mejorar las relaciones con los clientes, fomentar la retención y lealtad y en última instancia, incrementar la rentabilidad del negocio, así como la expansión a más clientes por medio de conexiones.

Las plataformas de software CRM desempeñan una función crucial en la implementación de estrategias de CRM. Estas herramientas posibilitan a las empresas recopilar, organizar y

analizar datos relacionados con los clientes, tales como historiales de compras, preferencias, interacciones anteriores y otra información pertinente. Al centralizar estos datos, las empresas pueden obtener una visión más completa de sus clientes y consecuentemente, personalizar sus interacciones y servicios de manera más efectiva.

Entre las funciones comunes de un sistema CRM se encuentran la gestión de contactos, el seguimiento de ventas, el servicio al cliente, la automatización de marketing y la generación de informes analíticos. Estas funciones operan en conjunto para mejorar la eficiencia operativa y ofrecer una experiencia más personalizada y satisfactoria al cliente.

La incorporación de un Sistema de Gestión de Relaciones con el Cliente en un proyecto de desarrollo de software conlleva una serie de beneficios significativos que inciden de manera positiva en la administración empresarial y en la interacción con los clientes. A continuación, se resaltan algunas de las razones fundamentales que subrayan la relevancia de un CRM en este entorno (Fairhurst, 2001):

- Centraliza la información relativa a los clientes, englobando historiales de interacciones, preferencias y transacciones. Este enfoque facilita la gestión más efectiva de las relaciones con los clientes.
- Tiene la capacidad de automatizar tareas repetitivas dentro del proceso de ventas, como seguimientos, recordatorios y actualizaciones de estado. Esto posibilita que el equipo de ventas dedique su tiempo a actividades más estratégicas y de mayor valor.
- Conforme la empresa experimenta un crecimiento, un CRM escalable puede ajustarse para gestionar un aumento en el volumen de datos y usuarios. De esta manera, proporciona una base robusta para el desarrollo y la expansión de la empresa.

2.3 Estado Del Arte

Se realiza para englobar las ideas principales de proyectos similares del porque realizar un sistema web con tecnologías .NET y base de datos en lenguaje SQL analizando diferentes motores y lenguajes de programación alternativos para conocer el problema, solución y técnicas aplicadas.

Tabla 1. Estado del arte. Fuente: Elaboración propia

Referencia	Problema	Métodos utilizados	Resultados
(Granja, 2017)	Manejo y gestión de la información	Aplicación móvil "SYS-MET" que ayuda a la transmisión de información de reportes diarios	Mejorar la transmisión de los reportes diarios de obra (RDO) desde los principales campos
(Guaman, 2021)	El proceso de gestión de inventarios se realiza de forma manual	Sistema web con Laravel, framework de PHP bajo la metodología SCRUM y el patrón MVC	La eficiencia de los procesos analizados mejora en un promedio de 78.63%
(Cruz, 2013)	Contar con una herramienta que ayude a organizar a los clientes y brindarles un mejor servicio	Sistema web con PHP y como motor de base de datos SQL Server 2005	Generar la oportunidad de crear lazos de fidelidad con los clientes
(Príncipe, 2021)	Los datos son imprecisos, ya que no se cuenta con la información histórica o necesaria del stock en inventario	Sistema web con PHP y MySQL y marco de trabajo SCRUM	Un 57% de reducción de tiempo en el proceso de gestión de inventarios
(Casañ, 2020)	Gestionar la información con los servicios que automatizan el trabajo relacionado con los aspectos operativos	Software con .NET Framework y base de datos en SQL Server que se usa en un entorno empresarial	La eficiencia de los procesos mejora la relación con los clientes, cumpliendo con sus expectativas

Tabla 1 (continuación). Estado del arte. Fuente: Elaboración propia

Referencia	Problema	Métodos utilizados	Resultados
(Pecharromán, 2015)	Mala optimización de los procesos empresariales	Sistema con C# y Microsoft SQL Server 2012 con SCRUM y MVC	Se optimiza el acceso a toda la información de forma precisa y asegurando la integridad de datos
(López, 2020)	Procesos de gestión de inventario y trato con los clientes deficientes	Sistema web con ASP.NET MVC y SQL Server	Con los módulos se cubre a cabalidad los problemas de la parte administrativa
(Mejía, 2012)	Los procedimientos son manuales ocasionando un descontrol de material y proporcionando información poco confiable	Sitio web con ASP .NET y SQL Server	Se muestra información 100% confiable a la organización que audita el laboratorio
(Escudero, 2017)	Automatización del tiempo en procesos de evaluación a clientes	Sitio web con SQL Server y ASP .NET Core MVC	El sistema permite almacenar diferentes tipos de datos y comunicarse con otros sistemas
(Patarroyo, 2020)	Las pequeñas y medianas empresas no cuentan con herramientas que ayuden a realizar una adecuada gestión financiera	Software web con ASP .NET Core (con el lenguaje de programación C#) y MySQL como motor de base de datos	Obtener indicadores en tiempo real, lo que ayuda a mantener un control continuo, sobre las tareas y los tiempos de respuesta en el área de finanzas

Tabla 1 (continuación). Estado del arte. Fuente: Elaboración propia

Referencia	Problema	Métodos utilizados	Resultados
(Chang, 2017)	Originalmente los procesos de gestión de inventario se llevan manera antigua, lo cual lleva a errores y descuadres al momento de redactar los reportes mensuales	Aplicación para el control de inventario con .NET Framework y SQL Server	Se realiza el registro de cada categoría, marca y productos, dando un beneficio en el control de los productos a inventariar
(Gomez, 2020)	Actividades manuales, por ello que hacen ineficientes las herramientas básicas de reporte o historial de ventas, clientes y movimientos de inventario	Sistema ERP para control del inventario y gestión en módulos de logística	Duración promedio de sus productos en 0.52 y disminuye en gran medida el sobre stock de estos
(Idrobo, 2016)	No tener un mayor control de los empleados y brindar un mejor y más rápido servicio a sus clientes	Arquitectura MVC (Modelo Vista Controlador) con motor RAZOR, SQL Server y Code Behind C#	Los resultados fueron mejores que los planteados. Se permite mejorar la logística de la empresa
(Zhou, 2020)	La mala optimización y poca escalabilidad de las bases de datos Access	Aplicación web sobre el framework Laravel y base de datos MySQL	Un ahorro de tiempo y costes considerable porque se automatizan muchas tareas que antiguamente se realizaban a mano y se unifica toda la información

Tabla 1 (continuación). Estado del arte. Fuente: Elaboración propia

Referencia	Problema	Métodos utilizados	Resultados
(Sandoval, 2021)	Seis años sin tener soporte y por tal razón muchas de las nuevas tecnologías y avances de software no tienen compatibilidad	Reingeniería con C#, .NET Framework y SQL Server con el patrón MVC	Se logra reestructurar el módulo de Compras del Sistema KOHINOR DBS y obtener un buen prototipo
(Manayalle, 2021)	La generación de facturas manuales o a papel, ocasionan un costo excesivo. No solo para la empresa que la emite, sino también para la que la recibe	Sistema web con arquitectura de capas con ASP.NET y base de datos en SQL Server	Permite ahorrar costos de impresión, y almacenamiento, así mismo permite reducir el tiempo en las operaciones de consulta, ofreciendo flexibilidad en operaciones
(Hernández, 2017)	Permitir unificar los datos de los distintos departamentos, facilitando la fluidez de la información del inventario de productos que la empresa oferta a sus clientes	Desarrollar un módulo de ERP (inventario) que permita agilizar los procesos con tecnologías .NET y SQL Server	Un alto porcentaje de agrado ante la implementación sistemas informáticos que brinden información veraz de los productos de inventario que se ofertan a los clientes
(Legua, 2015)	Pérdida de dinero y tiempo para los procesos	Sistema de control en C# y la Base de Datos estará en SQL Server 2012	Se reduce el tiempo que le tomaba al cliente en regresar reiteradas veces a la tienda, a recoger su equipo (antes 3 días, ahora 30 min. aproximadamente)

Tabla 1 (continuación). Estado del arte. Fuente: Elaboración propia

Referencia	Problema	Métodos utilizados	Resultados
(Garma, 2018)	No poder asegurar un nivel óptimo de calidad y por lo tanto que se vea reflejado en una baja demanda de productos	Sistema Web con C# como lenguaje de programación con el Framework ASP.NET MVC y motor de base de datos SQL Server	El porcentaje de productos defectuosos se redujo en un 32,2% y el porcentaje de productos entregados a tiempo aumenta en un 35,87%
(Arciniegas, 2019)	Se requiere la Implementación de los módulos de ERP con el fin de mejorar sus procesos administrativos y unificar sus bases de datos	Sistema de información con C# y SQL Server aplicando la metodología Rational Unified Process (RUP)	Manipulación de módulos en tiempo real brindando resultados intrínsecos a cada sección

2.4 Metodologías De Desarrollo De Software

Son marcos de trabajo que estructuran, planifican y controlan el proceso de desarrollo de sistemas de software. Su objetivo es organizar el trabajo de los equipos de desarrollo para que estos puedan diseñar, implementar y mantener aplicaciones de software de manera eficiente y eficaz. Las metodologías de desarrollo de software se basan en un conjunto de técnicas y métodos que se adaptan a las necesidades específicas de cada proyecto. Algunos de los factores que se deben considerar a la hora de elegir una metodología son (Santander, 2020):

- El tamaño y la complejidad del proyecto
- El presupuesto disponible
- El plazo de entrega
- Las características del equipo de desarrollo
- Los requisitos del cliente

El desarrollo de software es un proceso complejo que puede verse afectado por una serie de factores, como los cambios en los requisitos, los errores técnicos y los problemas de comunicación. Las metodologías de desarrollo de software ayudan a mitigar estos riesgos y garantizar que los proyectos se completen con éxito.

Las metodologías de desarrollo de software tradicionales más utilizadas son (Santander, 2020):

- Cascada: Donde las etapas se organizan de arriba abajo (de ahí el nombre), antes de cada etapa se debe revisar el producto para ver si está listo para pasar a la siguiente fase. Los requisitos y especificaciones iniciales no están predispuestos para cambiarse, por lo que no se pueden ver los resultados hasta que el proyecto ya esté bastante avanzado.
- Espiral: Se divide en cuatro etapas: planificación, análisis de riesgo, desarrollo de prototipo y evaluación del cliente. El nombre de esta metodología se da por su funcionamiento, ya que se van procesando las etapas en forma de espiral. Cuanto más cerca del centro se está, más avanzado está el proyecto.
- Incremental: En esta metodología de desarrollo de software se va construyendo el producto final de manera progresiva. En cada etapa incremental se agrega una nueva funcionalidad, lo que permite ver resultados de una forma más rápida. El software se puede empezar a utilizar incluso antes de que se complete totalmente y, en general, es más flexible que las demás metodologías.

Las metodologías de desarrollo de software ágiles más utilizadas son:

- Scrum: Es una metodología ágil que divide los requisitos y tareas y se itera sobre bloques de tiempos cortos y fijos, para conseguir un resultado completo en cada iteración. Las etapas son: planificación de la iteración (planning sprint), ejecución (sprint), reunión diaria (daily meeting) y demostración de resultados (sprint review). Cada iteración por estas etapas se denomina también sprint.
- Programación Extrema (XP): Es una metodología de desarrollo de software basada en las relaciones interpersonales. Su principal objetivo es crear un buen ambiente de trabajo en equipo y que haya una retroalimentación constante del cliente. El trabajo

se basa en 12 conceptos: diseño sencillo, pruebas, refactorización y codificación con estándares, propiedad colectiva del código, programación en parejas, integración continua, entregas semanales e integridad con el cliente, cliente in situ, entregas frecuentes y planificación.

2.5 Herramientas De Software

2.5.1 *SQL Server Management Studio (SSMS)*

De acuerdo con (Perez, 2011), es un entorno de desarrollo integrado (IDE) que proporciona una interfaz gráfica de usuario para administrar y desarrollar bases de datos SQL Server. SSMS es una herramienta esencial para los administradores de bases de datos y desarrolladores de SQL Server, ya que proporciona una amplia gama de herramientas y características para realizar una variedad de tareas.

Específicamente, SSMS se puede utilizar para:

- Crear, administrar y modificar bases de datos, tablas, vistas, procedimientos almacenados, funciones y otros objetos de base de datos.
- Realizar consultas y análisis de datos.
- Realizar copias de seguridad y restauración de bases de datos.
- Administrar el rendimiento y la seguridad de las bases de datos.

SSMS es una herramienta poderosa y versátil que puede ayudar a los administradores de bases de datos y desarrolladores de SQL Server a ser más eficientes y productivos.

2.5.2 *Visual Studio*

Como afirma (Microsoft, 2023) es un entorno de desarrollo integrado (IDE) que proporciona una plataforma completa para el desarrollo de software. Incluye un editor de código, compiladores, depuradores, herramientas de control de versiones y otras funciones que ayudan a los desarrolladores a escribir, depurar, probar y compilar código, así como a implementar aplicaciones. Visual Studio es compatible con una amplia gama de lenguajes de programación y plataformas, lo que lo convierte en una herramienta versátil para el desarrollo de software moderno.

Algunos conceptos de relevancia son (Microsoft, 2023):

1. Entorno de desarrollo integrado (IDE): Es un software que proporciona un conjunto de herramientas para el desarrollo de software. Suelen incluir un editor de código, compiladores, depuradores, herramientas de control de versiones y otras funciones que ayudan a los desarrolladores a crear proyectos
2. Compiladores: Los compiladores son programas que traducen el código fuente escrito por los desarrolladores a un lenguaje que las computadoras puedan entender
3. Depuradores: Los depuradores son programas que ayudan a los desarrolladores a encontrar y corregir errores en su código o bien para dar seguimiento a las líneas de código paso por paso permitiendo ver el estado de las variables o componentes en cada etapa de la ejecución
4. Control de versiones: El control de versiones es un sistema que permite a los desarrolladores realizar un seguimiento de los cambios en su código y restaurar versiones anteriores del código si es necesario

2.5.3 Playwright E2E

Según (Playwright, 2023) es una herramienta de software de código abierto desarrollada por Microsoft para pruebas de tipo *end-to-end* y pruebas *cross-browser* (compatibilidad entre diferentes navegadores web) en ambientes de desarrollo con Node.JS, .NET, Python y Java. Es una herramienta muy rápida y optimizada que permite lanzar un navegador dedicado para simular comportamientos de un sitio web, es compatible con Chrome (cuenta con una extensión en la Chrome Web Store), Edge, Firefox y Safari,

2.6 Diferencias Entre Front-End Y Backend

El desarrollo web se divide en dos grandes áreas llamadas backend y front-end. Por el lado del backend se encuentra toda la lógica de negocio y conexión a base de datos para realizar operaciones CRUD con la información de dicha base, información la cual será enviada al front-end por medio de una herramienta llamada API.

Las API (Application Programming Interface) son creadas por el desarrollador backend y fungen como puente de comunicación entre estas dos áreas. Esta segmentación permite, por una parte, tener la seguridad de tratar los datos del lado del servidor (backend), cuidar la robustez de la base y su información, mientras que la vista (front-end) consume la información en formatos de intercambio de datos como JSON o XML (por mencionar dos de los más populares) y servirlos en las interfaces de usuario que ve el cliente al cargar la página, sitio o aplicación web.

2.7 Lenguajes De Front-End

2.7.1 *JavaScript*

Según el (TC39, s.f.) es un lenguaje de programación multiparadigma, orientado a objetos y de alto nivel, que se utiliza para desarrollar aplicaciones web dinámicas. Su flexibilidad y universalidad lo convierten en una herramienta esencial para los desarrolladores, permitiéndoles crear sitios web interactivos, aplicaciones móviles y aplicaciones de servidor.

La especificación ECMAScript es desarrollada por el Comité Técnico 39 de Ecma International. La última versión de la especificación ECMAScript para el lenguaje es la versión 2023, que fue publicada en junio de 2023.

Este lenguaje se ha convertido en la columna vertebral de la programación web moderna, permitiendo a los desarrolladores crear experiencias de usuario ricas, optimizadas y dinámicas. Una de las razones de su popularidad es la amplia variedad de frameworks y bibliotecas disponibles, que facilitan el desarrollo y aumentan la productividad del programador.

Entre los frameworks y librerías más populares de JavaScript se encuentran: React, Angular, Vue.js, Next.js, entre otros.

Además de estas herramientas, JavaScript también se utiliza en el desarrollo de aplicaciones móviles multiplataforma (iOS, Android, etc.) a través de tecnologías como React Native, aplicaciones de escritorio multiplataforma como Tauri y en el desarrollo de

aplicaciones de servidor mediante el entorno de ejecución Node.js. En el desarrollo web, JavaScript se utiliza para crear aplicaciones de chat en tiempo real, sitios web de compras entre muchas otras aplicaciones interactivas (Schmedtmann).

2.7.2 CSS

Cascading Style Sheets (Hojas de estilo en cascada) es un lenguaje de marcado utilizado en el desarrollo web para definir y controlar la presentación visual de los documentos HTML. Permite a los diseñadores y desarrolladores web especificar atributos precisos, como colores, fuentes, transiciones, efectos y espaciados para los elementos gráficos en un navegador. Esta separación de la estructura del documento de su estilo visual facilita la creación de sitios web con una apariencia coherente y estética.

Además, CSS ofrece la capacidad de aplicar estilos a elementos individuales o grupos de elementos, permitiendo así un control minucioso sobre el diseño y la disposición de los elementos en una página web. Además, es un lenguaje que vive bajo la especificación de la web por lo que asegura la consistencia y compatibilidad en la presentación visual de los sitios web en diferentes plataformas y dispositivos (Bos, 2023).

2.7.3 HTML

Según (Claro, 2010) HyperText Markup Language es un lenguaje de marcado (no de programación) que los desarrolladores web utilizan para estructurar y describir el contenido y la semántica de una página web. HTML consta de elementos que describen diferentes tipos de contenido para el esqueleto de la aplicación, sitio o página tales como párrafos, enlaces, encabezados, imágenes, secciones, vídeos, modales entre otros. Actualmente cuenta con más de 100 elementos semánticos. Los navegadores web entienden y renderizan el código HTML como páginas gracias a la especificación de la web.

La importancia de HTML para poder manejar frameworks como ASP es que HTML es el lenguaje de marcado que se utiliza para crear el front-end de una aplicación web y la semántica de las mismas etiquetas se puede utilizar al desarrollar en herramientas que

requieren de estructura de componentes por ejemplo la librería de JavaScript React o las etiquetas de ASPX al usar C#.

2.7.4 ASP.NET

De acuerdo con (Microsoft, 2023) Active Server Pages es un framework de desarrollo web del lado del servidor que se utiliza para crear aplicaciones web dinámicas que permiten crear componentes a partir de etiquetas semánticas que al final se compila en un ensamblado que contiene el código máquina que se ejecuta en el servidor web. El ensamblado contiene código que genera el código HTML y se envía para poder ser interpretado por el navegador web.

Trabaja del lado del servidor y se utiliza para crear aplicaciones web dinámicas que permiten crear componentes a partir de etiquetas semánticas, que al final se compila en un ensamblado que contiene el código máquina y se ejecuta en el servidor web. El ensamblado contiene código que genera el HTML y se envía para poder ser interpretado por el navegador web y así mismo al cliente.

Trabajar con ASP permite por un lado utilizar muchas de las virtudes del lenguaje C# para tareas específicas y complejas tales como la creación de listas, métodos tipados que retornen un valor con base a la respuesta de un procedimiento almacenado, conexión a base de datos, métodos de renderizado condicional y manejo de sesiones.

Los archivos que son compilados a código que pueda entender el navegador (HTML) en un proyecto de ASP.NET son de extensión .aspx y los que se transpilan (proceso de adaptación del código JavaScript para que todos los motores puedan ejecutar el código sin importar la versión) a código JavaScript tienen la extensión .aspx.cs.

2.8 Lenguajes De Backend Y Bases De Datos

2.8.1 C#

Es un lenguaje de programación moderno, orientado a objetos y con seguridad de tipos. Está diseñado para ser un lenguaje de alto rendimiento y seguridad, y es parte de la plataforma .NET.

C# es un lenguaje de programación orientado a componentes, lo que significa que se basa en la idea de que las aplicaciones se pueden construir a partir de componentes reutilizables. C# proporciona construcciones de lenguaje para admitir directamente estos conceptos, lo que lo hace un lenguaje natural para crear y usar componentes de software. Admite nuevas cargas de trabajo y prácticas de diseño de software emergentes. Sin embargo, en el fondo, C# sigue siendo un lenguaje orientado a objetos. Esto significa que las aplicaciones están construidas con base a entidades que tienen un tipo y un comportamiento, algunas características del lenguaje (Microsoft, 2023):

- Orientado a objetos: Un paradigma de programación que trata los datos y el código como entidades separadas. Los objetos se componen de propiedades, que almacenan datos, y métodos, que ejecutan código.
- Seguridad de tipos: Un sistema que garantiza que los datos se utilicen de forma correcta. En C#, los tipos de datos se comprueban en tiempo de compilación, lo que ayuda a prevenir errores y fallas.
- Componente: Una unidad de código reutilizable que puede ser utilizada en diferentes aplicaciones.

2.8.2 SQL

Según (Lozada, 2021) Structured Query Language (lenguaje de consulta estructurado) es un lenguaje utilizado para administrar los datos y la información de una base de datos relacional.

La diferencia con los lenguajes de programación de propósito general tales como JavaScript, Go, Java, es un lenguaje de propósito específico; significa que cumple una función concreta en un proyecto de software. Permite crear la estructura de la base de datos, administrar sus permisos, datos, roles, usuarios y esquemas.

Los 3 lenguajes de SQL

1. DML (Data Manipulation Language): Este lenguaje tiene las instrucciones que manipulan los datos, es decir se realizan las operaciones CRUD de la información dentro de la base de datos, operaciones tales como altas, ediciones, consultas y

eliminaciones. La siguiente línea de código es un ejemplo de una instrucción de SQL que actualiza un campo de una tabla dada una condición:

```
UPDATE [esquema].[tabla] SET NOMBRE = @NOMBRE WHERE ID = @ID
```

2. DDL (Data Definition Language): Este lenguaje tiene las instrucciones que definen el esqueleto y arquitectura de la base de datos, esquemas, tablas entre otros. La siguiente línea de código es un ejemplo de una instrucción de SQL que agrega una columna de tipo entero a una tabla:

```
ALTER TABLE [esquema].[tabla] ADD EDAD INT
```

3. DCL (Data Control Language): Este lenguaje permite asignar o revocar permisos, así como creación de usuarios. La siguiente línea de código es un ejemplo de una instrucción de SQL que revoca los permisos de algún rol para insertar a una tabla:

```
REVOKE INSERT ON [esquema].[table] TO someRole
```

Algunos de los motores de bases de datos relacionales más populares son:

- Microsoft SQL Server
- PostgreSQL
- SQLite

- Motor de base de datos SQL Server

Es un sistema de gestión y motor de bases de datos relacionales desarrollado por Microsoft. Permite almacenar, organizar y gestionar datos en una base de datos relacional. Una base de datos relacional es un conjunto de datos organizados en tablas, que están relacionadas entre sí mediante relaciones. Utiliza el lenguaje de consulta estructurado para acceder, manipular y administrar los datos de la base de datos. SQL es un lenguaje estándar que se utiliza en la mayoría de los sistemas de gestión de bases de datos relacionales.

Se utiliza en una amplia gama de aplicaciones, que incluyen (Microsoft, 2023):

- Aplicaciones empresariales: sistemas de gestión de inventario, sistemas de gestión de clientes y sistemas de planificación de recursos empresariales.

- Aplicaciones web: sitios web de comercio electrónico y aplicaciones de medios sociales.
- Aplicaciones móviles: aplicaciones de juegos y aplicaciones de productividad.

SQL Server es un sistema de gestión de bases de datos potente y versátil que puede utilizarse para una amplia gama de aplicaciones como las listas anteriormente por ser algunas de las más populares.

A continuación, se presentan algunos beneficios de SQL Server:

- **Confiable:** SQL Server es un sistema de gestión de bases de datos relacional probado y confiable. Ha sido utilizado por empresas de todo el mundo durante más de 30 años.
- **Compatible:** SQL Server es compatible con una amplia gama de aplicaciones y entornos. Puede utilizarse en Windows, Linux y en la nube.
- **Escalable:** SQL Server es escalable y puede adaptarse a las necesidades de las empresas en crecimiento.
- **Seguro:** SQL Server es seguro y protege los datos de la base de datos

2.9 Patrones De Diseño

Desde el punto de vista de (Blancarte, 2016) los patrones de diseño representan soluciones comunes para problemas recurrentes en el desarrollo de software. Estos patrones, equivalentes a planos prediseñados, son adaptables y personalizables para resolver problemas específicos en el código.

No basta con replicar un patrón directamente en el programa; es necesario comprender los detalles del patrón y aplicar una solución adaptada a las necesidades del programa en cuestión.

Es importante diferenciar los patrones de diseño de los algoritmos. Mientras que un algoritmo prescribe un conjunto preciso de acciones para lograr un objetivo, un patrón de diseño es una descripción más general de una solución. El mismo patrón aplicado en dos programas diferentes puede tener implementaciones de código distintas.

Se puede hacer una analogía entre un algoritmo y una receta de cocina, ya que ambos siguen pasos claros para alcanzar un resultado específico. Por otro lado, un patrón de diseño se asemeja más a un plano, donde se especifican las características del resultado, pero el orden exacto de implementación depende del programador.

2.10 MVC

Según (MDN contributors, 2022) el Modelo Vista Controlador es un patrón de diseño de software que se caracteriza por separar la lógica de negocios, la interfaz de usuario y la lógica de control en una aplicación. Esta división de responsabilidades tiene como objetivo principal mejorar la modularidad, la mantenibilidad y la escalabilidad del software.

En términos técnicos, el modelo representa tanto los datos de la aplicación como la lógica de negocios asociada con esos datos. Es responsable de almacenar y gestionar los datos, así como de proporcionar los algoritmos y operaciones necesarios para manipular esa información.

La vista, por otro lado, corresponde a la interfaz de usuario de la aplicación. Su función principal es mostrar los datos de manera comprensible para el usuario y permitir la interacción del usuario con la aplicación. Mientras tanto, el controlador actúa como un intermediario entre el modelo y la vista. Se encarga de recibir las entradas del usuario y de actualizar tanto el modelo como la vista según corresponda.

La Figura 2 muestra el modelo de ejemplo sobre cada característica que conforma a MVC y enumera los pasos que suelen corresponder a la funcionalidad, es decir, la manera de tratar con los datos dentro del proceso de desarrollo:

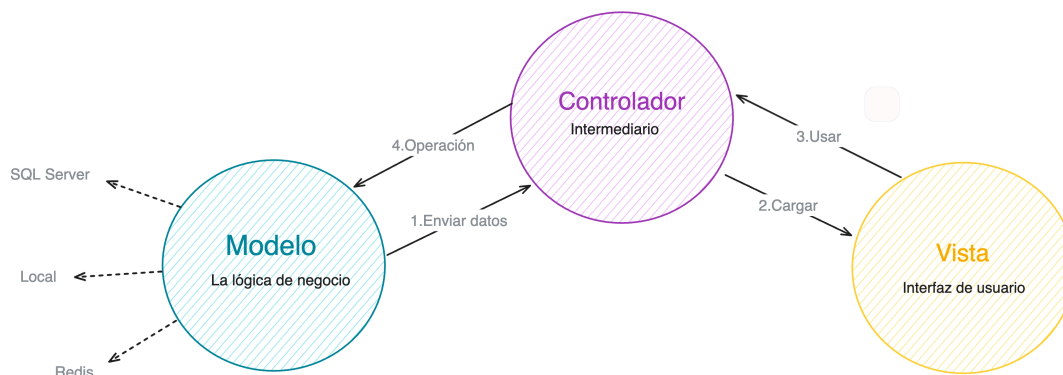


Figura 2. Esquema del patrón MVC. Fuente: Elaboración propia

Las ventajas de implementar el patrón MVC son significativas. La modularidad se logra mediante la independencia de los componentes, lo que permite su desarrollo y mantenimiento separados. La mantenibilidad se mejora gracias a la clara separación de preocupaciones, lo que facilita la comprensión y modificación del código.

2.11 Arquitecturas De Software

Las arquitecturas de software representan estructuras esenciales que establecen la organización y el diseño inherente a un sistema de software. Estas configuraciones sirven como cimientos para la edificación, implementación y sostenimiento de aplicaciones, ofreciendo una perspectiva integral de la distribución y la interrelación de los diversos componentes del software.

Algunas arquitecturas de software más utilizadas son (Maceda, 2016):

- La Arquitectura de Microservicios: Desglosa la aplicación en servicios autónomos y pequeños, operando de manera independiente y comunicándose mediante interfaces de programación de aplicaciones.
- Arquitectura sin servidor: Se enfoca en la ejecución de funciones en lugar de mantener servidores activos constantemente, ofreciendo una gestión más eficiente de recursos ya que todo el renderizado y una parte de la lógica ocurre en la vista.
- Arquitectura monolítica: Representa una aplicación única donde todos los componentes están interconectados, siendo menos modular que otras arquitecturas.
- La Arquitectura de capas: Divide la aplicación en estratos lógicos (presentación, lógica de negocio, acceso a datos) para una organización y gestión más efectiva del código.

2.12 Diseño De La Arquitectura

En el ámbito del desarrollo web, el diseño más común de las arquitecturas es (Lozada, 2021):

- Frontend: HTML, CSS, JavaScript, junto con frameworks como NextJS (para la librería React), Angular , Vue.js entre otros.

- Backend: Implica el uso de lenguajes como Node.js, Go, C# y frameworks como Django, Express entre otros.
- Base de datos: PostgreSQL, SQL Server, MongoDB, Redis, entre otros, para la administración de datos.
- Protocolos y herramientas de comunicación: Se utilizan protocolos HTTP/HTTPS que funcionan como puente de comunicación entre el frontend y el backend.

La Figura 3 muestra la arquitectura empleada en el proyecto, haciendo énfasis en cómo las aplicaciones web modernas tienden cada vez más a respetar la simplicidad, delegando responsabilidades a cada rol respectivamente.

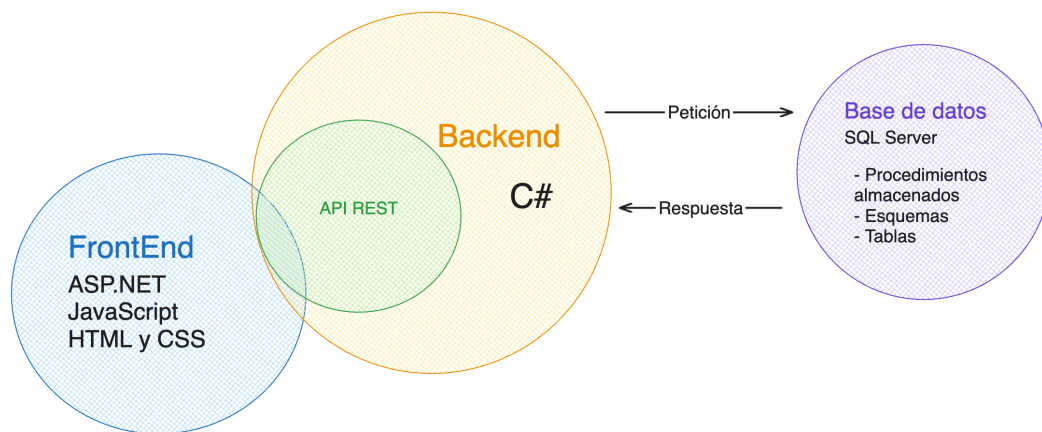


Figura 3. Esquema de arquitectura del sistema. Fuente: Elaboración propia

La elección de la metodología Scrum se fundamenta en su enfoque iterativo e incremental, posibilitando así la entrega continua de funcionalidades y la capacidad de adaptación a cambios en los requisitos del proyecto. Scrum promueve la colaboración estrecha entre los integrantes del equipo, facilitando una comunicación efectiva y la resolución ágil de problemas. Asimismo, la transparencia y la retroalimentación continua en Scrum concuerdan con la necesidad de la empresa de contar con un sistema flexible y adaptable a sus procesos empresariales en constante evolución.

Es conveniente utilizar esta metodología con el patrón MVC y la arquitectura cliente servidor, lo cual crea consistencia en cada una de las etapas, tanto de cómo se trabajan los datos, las operaciones de los módulos, la segmentación de código hasta las buenas prácticas de desarrollo y una integración moderna y satisfactoria para las necesidades de la empresa de acuerdo con los requisitos establecidos en los planes de trabajo.

Las tecnologías y herramientas mencionadas buscan complementar el desarrollo desde un punto de vista técnico, permitiendo conocer la documentación existente para dar solución a problemas que se presenten en etapas posteriores a la investigación y la referencia de las funciones, es decir cada etapa que presente un riesgo para el cumplimiento de los requisitos, debe ser subsanada bajo la investigación en proyectos similares u documentación oficial de la tecnología para facilitar el desarrollo.

Capítulo III

Desarrollo

A continuación, se presentan los procesos propuestos en el cronograma de actividades marcado al inicio del proyecto. En el entendido de que dichas actividades llevan un orden cronológico y son redactadas al momento de su desarrollo para considerar todos los detalles.

Cada etapa contiene una descripción detallada de cómo se lleva a cabo y las herramientas que se utilizan para cumplir el objetivo.

3.1 Análisis

Para llevar a cabo el análisis general del proyecto se llevan a cabo reuniones de manera virtual de aproximadamente 2 horas los jueves, por medio de las cuales se hace un levantamiento de requisitos con el cliente. Se detallan casos reales y escenarios donde el requisito debe atender de manera oportuna posibles flujos alternos y cuál es el fin de dicho planteamiento. De igual forma en la segunda sesión se firma el documento de “Contrato de confidencialidad” que detalla las responsabilidades de las personas involucradas en el desarrollo del sistema, tanto para el encargado como para el desarrollador. Cada uno de estos requisitos son puestos a detalle del desarrollador quien expone el entendimiento de cada flujo y lo corrobora con el cliente y el encargado de proyecto, quien toma la decisión final de si el proceso es comprendido de principio a fin y los requerimientos satisfacen las necesidades de calidad para el sistema. Una vez que todos los requisitos son descritos y aprobados se procede al modelado y la etapa de diseño.

3.1.1 Atributos De Calidad

La Tabla 2 muestra una exhaustiva exposición de los atributos de calidad incorporados en el diseño y desarrollo del sistema en cuestión. La metodología adoptada y siguiendo las buenas prácticas de ingeniería de software asegura la entrega de un producto que cumple con estándares rigurosos de rendimiento, confiabilidad y mantenibilidad.

Se presenta de manera precisa los atributos de calidad identificados, ofreciendo una descripción detallada de cada uno, así como la cantidad de recursos y esfuerzos dedicados a su implementación. Esta información proporciona una comprensión clara de cómo el

proyecto aborda los aspectos fundamentales de calidad, estableciendo una base sólida para la toma de decisiones y la evaluación continua del sistema.

Los principales aspectos que deben ser considerados para cubrir los estándares de calidad solicitados por la empresa en el levantamiento de requisitos son:

Tabla 2. Atributos de calidad. Fuente: Elaboración propia

Atributo	Descripción	Cantidad
Velocidad y tiempo de respuesta	El sistema debe ser capaz de cargar las operaciones de alta demanda en tiempos no mayores a un minuto	Menos de 60 segundos
Rendimiento	El sistema debe ser capaz de manejar simultáneamente múltiples solicitudes de los usuarios	Mínimo de 40 usuarios
Plataforma	El sistema debe ser compatible con los diferentes navegadores y motores de JavaScript	Google Chrome, Microsoft Edge (Chromium), Firefox, Opera, Safari, Brave, Arc entre otros
Usabilidad y accesibilidad	El sistema debe contar con la interfaz de usuario intuitiva, inclusiva y fácil de usar	Al menos 90% de prácticas recomendadas en informe de Lighthouse
Seguridad	El sistema debe contar con medidas solidas de seguridad para proteger la información confidencial de los clientes, ya que los datos son el activo más valioso del sistema	Autenticación, integridad y no repudio

3.2 Diseño

Para la etapa de diseño se utilizan la herramienta de diseño Excalidraw, el cual es un software de modelado y diseño que permiten plasmar los flujos de cada pantalla y requisito en espacio compartidos y seguros para un equipo de desarrollo.

Excalidraw es una aplicación web que permite con el uso de una pizarra blanca de diseño libre crear diagramas de todo tipo:

- Diagramas UML y de flujo
- Wireframes o interfaces de usuario de modelado ágil
- Modelos y esquemas

La Figura 4 muestra el diagrama de flujo muestra el proceso y las evaluaciones para el inicio de sesión del perfil de usuario en el sistema web, detalla que pasa cuando el perfil no concuerda con las credenciales y cuál es el flujo normal y alterno:

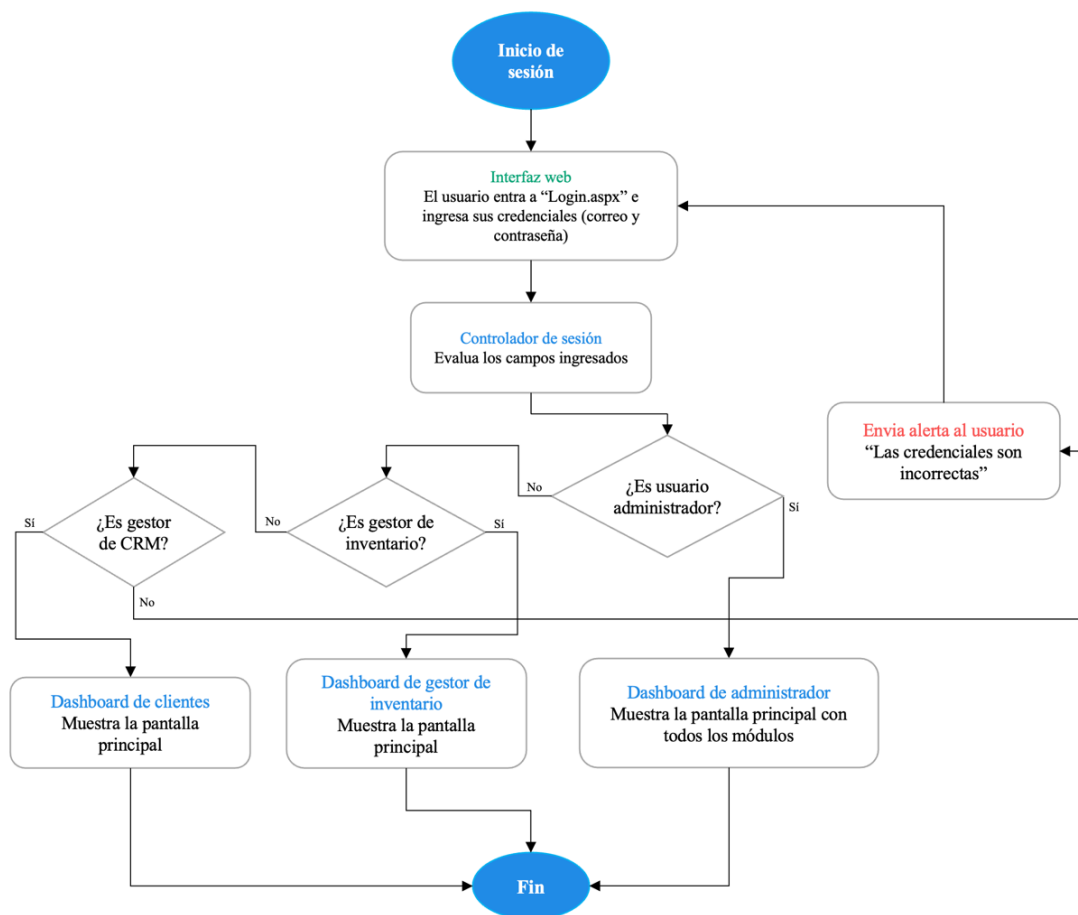


Figura 4. Diagrama de flujo inicio de sesión. Fuente: Elaboración propia

3.2.1 Diagramas UML

Una vez que ambos usuarios ingresan sus credenciales en la pantalla de inicio de sesión, el sistema evalúa estos parámetros y de ser correcto permite visualizar el dashboard (pantalla principal) del módulo que le corresponda (en caso de ser administrador, puede ver todos los módulos).

Para determinar el proceso anterior se debe hacer uso de los controles y entidades presentados en la Figura 5:

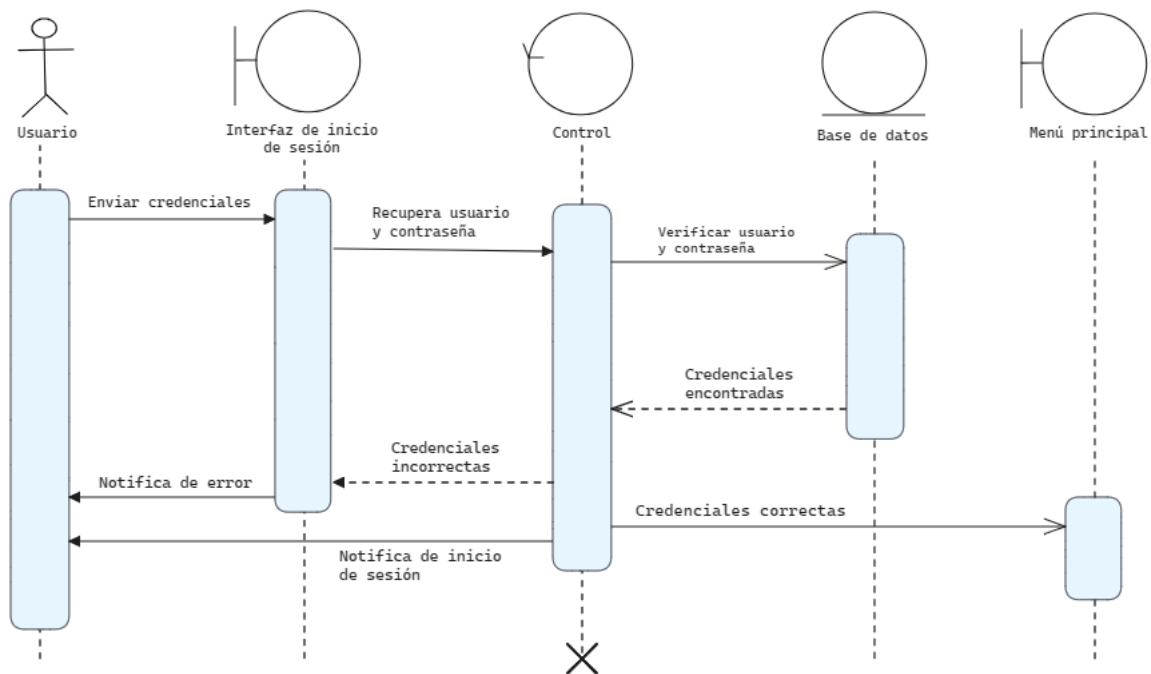


Figura 5. Diagrama de secuencia inicio de sesión. Fuente: Elaboración propia

El sistema será desarrollado para trabajar en un entorno web, lo que permitirá su utilización de forma rápida y multiplataforma, cabe recalcar que este es un sistema independiente, no necesita de otro sistema para funcionar de manera correcta y tampoco pertenece a un supra sistema. La Figura 6 presenta las posibles interacciones del usuario con los módulos considerando las evaluaciones de perfil previas del sistema.

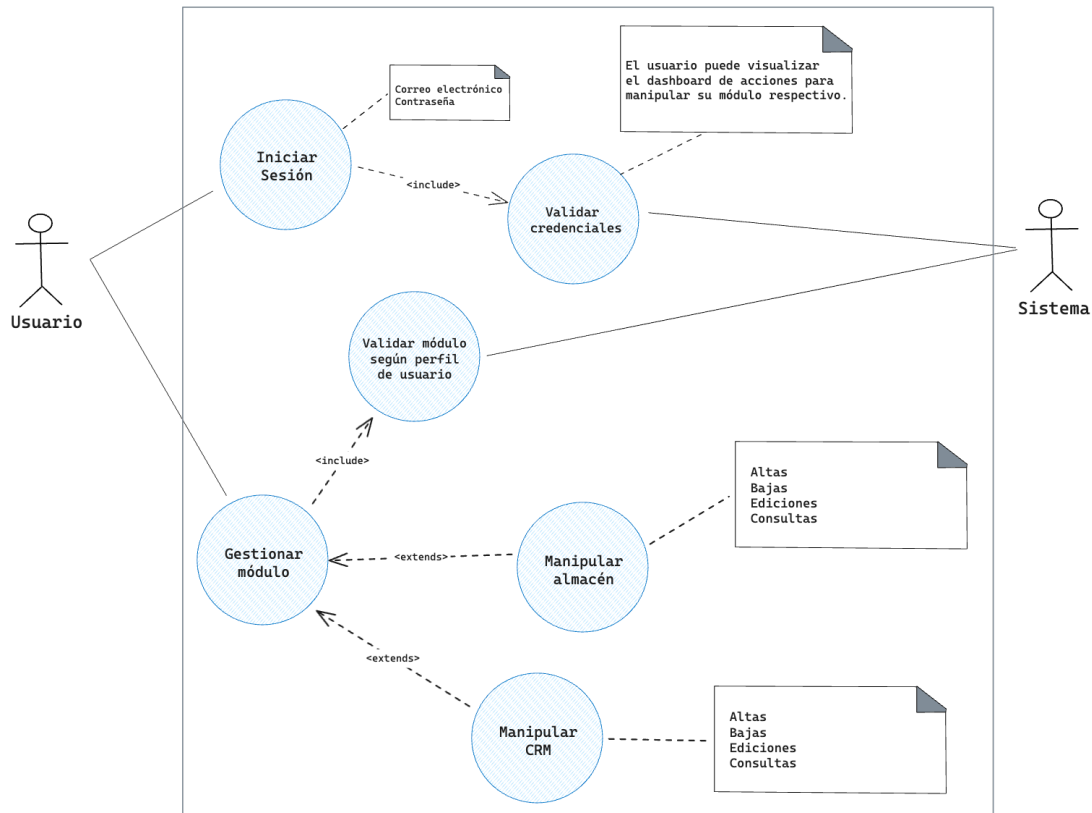


Figura 6. Diagrama de caso de uso general. Fuente: Elaboración propia

En la Figura 7 se muestra el caso de uso específico donde el usuario inicia sesión dentro del sistema y los datos que este debe ingresar:

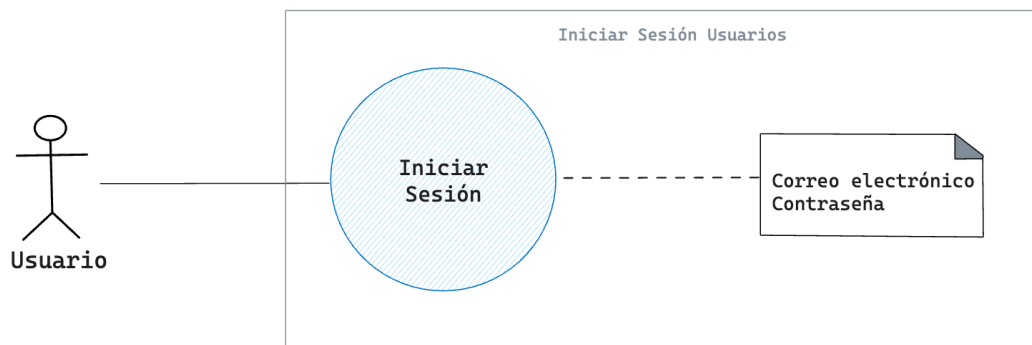


Figura 7. Diagrama de caso de uso inicio de sesión. Fuente: Elaboración propia

Una vez que el usuario con el perfil de gestor de inventario tiene acceso, puede realizar sus respectivas operaciones dentro del módulo de inventario, para lo que el sistema valida con un identificador único los movimientos en las pantallas que alteren la información de la base de datos, es por ello por lo que se controla el flujo en caso de errores u

modificaciones que son realizadas de manera exitosa, dando un *feedback* al usuario de como termina la operación. La Figura 8 muestra un ejemplo de operación y retroalimentación al usuario:

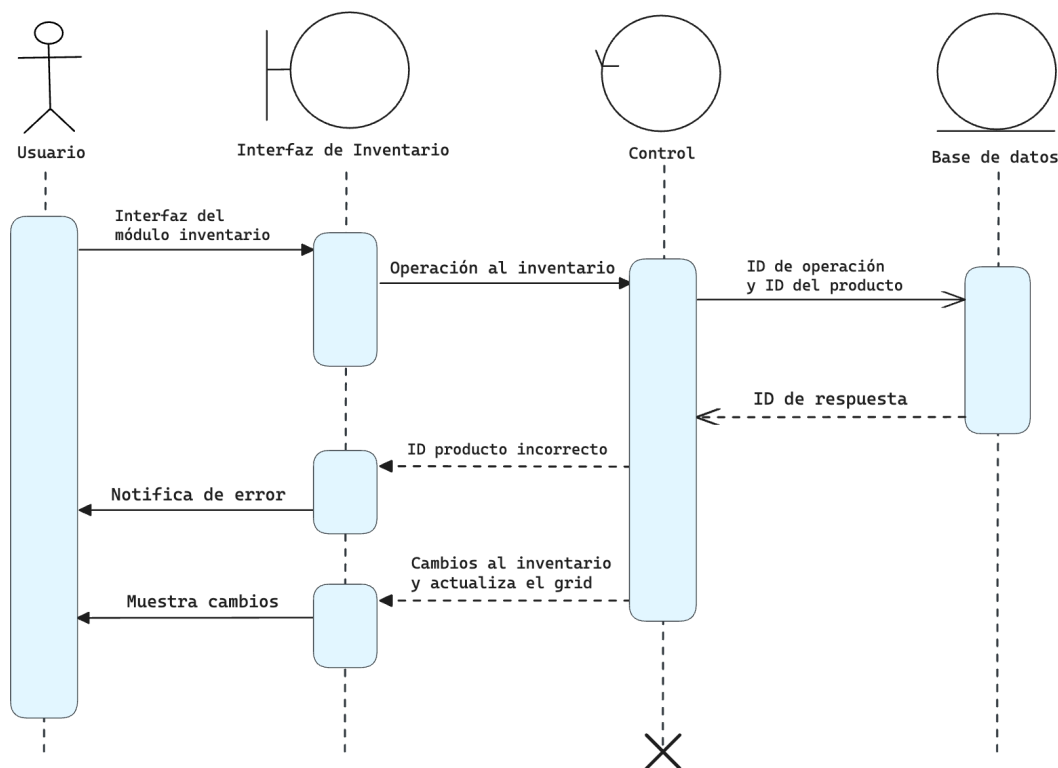


Figura 8. Diagrama de secuencia de operaciones al inventario. Fuente: Elaboración propia

3.3 Codificación

Una vez que el proyecto ha sido creado y la estructura de carpetas es la óptima para trabajar bajo las reglas del patrón MVC se puede comenzar con el proceso de codificación.

La primer etapa de codificación ocurre dentro de la base de datos, se crean los esquemas respectivos de cada módulo, es decir el esquema inventario contendrá todas aquellas tablas de referencia a esta sección, por ejemplo `[inventario].[CAT_MOTIVOS]` contiene una estructura, relaciones y permisos diferentes que `[crm].[CAT_MOTIVOS]` ya que pertenecen a diferentes esquemas.

Notar que las tablas tienen una estructura de nombre que sigue un modelo y reglas de buenas prácticas, para identificar su función dentro del esquema o la base de datos, y facilitar su indexado y mantenimiento por nuevos integrantes del equipo en el futuro, estas reglas son:

- CAT_NOMBRE: la palabra CAT hace referencia a tablas que contienen catálogos que están predefinidos o que ayudan a leer información, normalmente no son manipulables y cada registro tiene su identificador con relaciones a otras tablas.
- REL_NOMBRE: la palabra REL hace referencia a las tablas que crean una relación entre otras tablas del esquema, normalmente se utilizan para relacionar identificadores.
- T_NOMBRE: las tablas con la letra “T” al inicio, indican que es una tabla principal y que debe fungir como aquella tabla maestra donde van los registros de las vistas, estas tablas cuentan con validaciones especiales para proteger la integridad de los datos de ataques como la inyección de código SQL, algunas de estas medidas son:
 - Filtrado de entradas: Consiste en reemplazar los caracteres especiales, que pueden ser utilizados por los atacantes para inyectar código malicioso.
 - Uso de parámetros: el código SQL no se construye con la entrada del usuario, lo que hace que sea más difícil para los atacantes inyectar código.
 - Enmascaramiento de datos: Consiste en ocultar los datos sensibles, como las contraseñas, antes de almacenarlos en la base de datos.
- D_NOMBRE: estas tablas muestran los detalles de las tablas principales, para consultar características más específicas de ciertas columnas.

Al igual que las tablas, los procedimientos almacenados cuentan con una estructura de nombrado para proporcionar un mejor entendimiento de su función.

Las reglas de buenas prácticas para los nombre de los procedimientos son:

- INS_NOMBRE: procedimientos que agregar nuevos registros a tablas
- SEL_NOMBRE: procedimientos que listan registros de tablas
- UPD_NOMBRE: procedimientos que actualizan registros existentes en tablas
- DEL_NOMBRE: procedimientos que eliminan registros de tablas

La segunda etapa es el maquetado de interfaces de usuario, comúnmente esta etapa se suele confundir con una etapa de diseño. Sin embargo, el Front-end (que es exclusivo de la web) es dividido en tres roles y etapas de suma importancia:

1. Diseño UX/UI: Modelado de los flujos para las interfaces y cuál es la experiencia de usuario al manipular el sistema, atendiendo principios de usabilidad, accesibilidad, guías de estilo, semántica de componentes, metodología “Design Thinking” y cuidar el “Look and feel” de las pantallas.
2. Maquetado: Es traducir el diseño UI a código, esta etapa no es programación aún, se utilizan lenguajes de marcado y estilo como HTML y CSS
3. Programación Front-end: Aquí ya está la lógica de la vista y la interpretación de datos que vienen del backend, en la interfaz que ve el usuario final. En esta etapa se utilizan lenguajes de programación como JavaScript o tecnologías modernas y emergentes como WebAssembly para compilar otros lenguajes en el navegador como Rust, C#, entre otros, sin tener que pasar por JavaScript (recordando que es el único lenguaje de programación que puede interpretar el navegador).

Una vez que las interfaces han sido creadas necesitan ser pobladas con datos provenientes de sus respectivas tablas, por lo que la información debe ser manipulada. Para poder realizar las operaciones CRUD de los datos existen varias alternativas, entre las más populares es trabajar con operaciones HTTP (GET, POST, PUT, DELETE, entre otras). Estas permiten comunicar canales de diferentes fuentes al desarrollar una API, es decir se pueden usar para comunicar, por ejemplo, un sistema con front-end en NextJS (framework de JavaScript para trabajar con la librería React) y backend en C#.

Con el uso del patrón MVC implementar tecnologías .NET es aún más conveniente que el ejemplo anterior para un desarrollo armónico de las vistas y los modelos es por ello por lo que para lograr la comunicación entre la base de datos y las vistas ASPX se utiliza Entity Framework el cual según (Microsoft, 2023) es un framework de código abierto que permite trabajar con datos en forma de objetos. Esto simplifica el desarrollo de aplicaciones basadas en datos, ya que no es necesario crear APIs externas.

Los archivos .aspx contienen los componentes de ASP que permiten crear los componentes de la interfaz que ve el usuario del sistema, estos componentes pueden ser desde muy básicos hasta muy complejos, es decir cómo podemos utilizar las etiquetas `<asp:Label/>`, como se pueden utilizar algunas más complicadas de programar como el `<asp:Repeater/>`, que se utiliza para iterar los datos de un modelo y mostrarlos como piezas de interfaz de usuario con datos de una tabla en SQL.

Al momento de crear el archivo .aspx es importante relacionarlo a una página maestra o Site.master que no es más que, el layout padre, de todos los componentes, aquí van aquellos elementos que persisten en cada pantalla del sistema, tales como la barra de navegación, el footer, los estilos, los scripts y las importaciones de las librerías (en caso de utilizar CDN's).

Una vez que el maquetado esta completo, se corrobora que todos los elementos contengan un identificador único para poder ser manipulados, bien por la parte de C# con el atributo ID y la propiedad `runat="server"`, como por la parte de JavaScript con el identificador `ClientInstanceName`, gracias a estos identificadores se pueden manipular los elementos del DOM en un archivo separado con extensión .js (para código JavaScript) o ser utilizados en el archivo .aspx.cs que contiene toda la lógica y programación de la vista con el lenguaje C#.

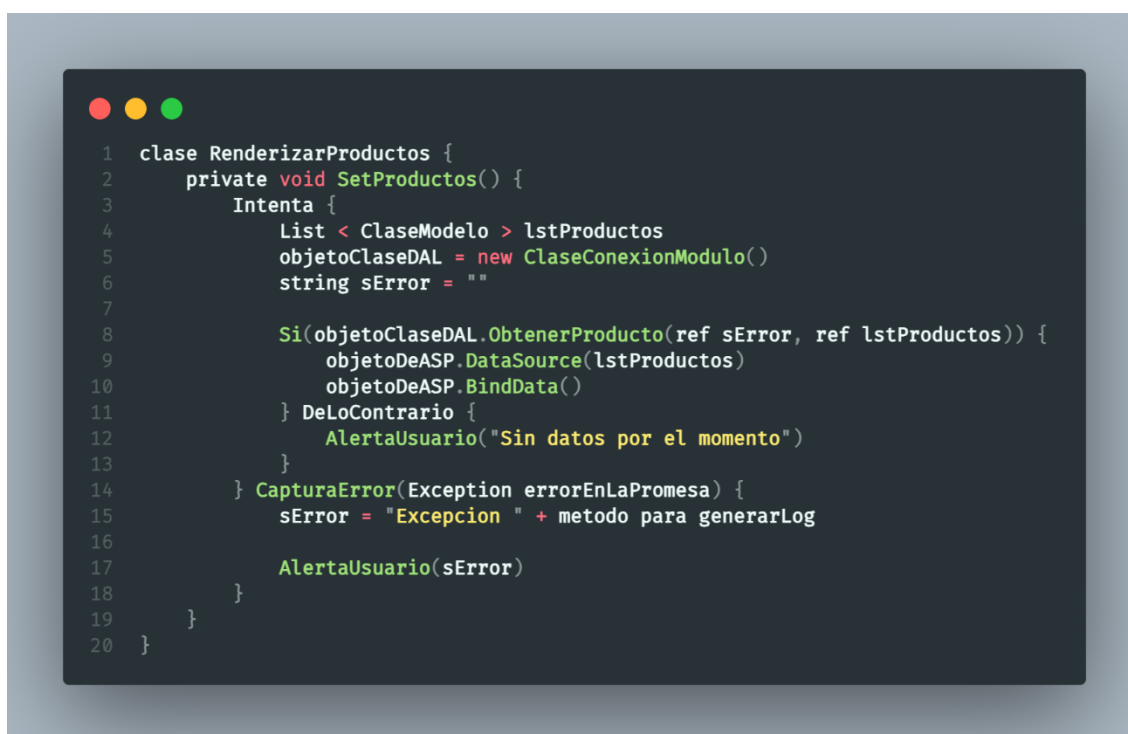
En el "code-behind" (archivo .aspx.cs) se pueden realizar diversas tareas relacionadas con la pantalla actual, una de ellas y en sí, el método principal de esa clase es el `Page_Load` que se encarga de ejecutar código al momento de cargar la página, normalmente este método se utiliza para realizar comprobaciones y resolver condicionales del tipo:

- ¿La sesión es nula?
- ¿Es la carga de la página un postback?
- ¿La carga de la página recibe datos por método POST o GET?
- ¿Es la carga de la página una llamada directa por la URL? y ¿Tiene permisos para acceder al contenido?
- ¿Se debe mandar a llamar a otro método de la clase al cargar?

Dentro del code-behind se pueden realizar otras tareas de suma importancia como:

- Mandar a llamar métodos de las clases DAL (Data Access Layer) para recuperar una respuesta de base de datos
- Iterar piezas de interfaz con datos de un modelo (previamente llenado con datos de base)
- Asignar valores a elementos del DOM (como el color de fondo o su fuente de datos)
- Renderizado condicional (crear un elemento u otro dependiendo de una evaluación)

La Figura 9 es un ejemplo de cómo se comportan las clases que se encargan de tratar los datos que provienen de la capa de datos.

A screenshot of a code editor with a dark background and light-colored text. The code is a pseudocode representation of a C# class named 'RenderizarProductos'. It includes a private method 'SetProductos()' which contains logic for fetching data from a database, handling errors, and displaying the results. The code is numbered from 1 to 20 on the left side.

```
1  clase RenderizarProductos {
2      private void SetProductos() {
3          Intenta {
4              List < ClaseModelo > lstProductos
5              objetoClaseDAL = new ClaseConexionModulo()
6              string sError = ""
7
8              Si(objetoClaseDAL.ObtenerProducto(ref sError, ref lstProductos)) {
9                  objetoDeASP.DataSource(lstProductos)
10                 objetoDeASP.BindData()
11             } DeLoContrario {
12                 AlertaUsuario("Sin datos por el momento")
13             }
14         } CapturaError(Exception errorEnLaPromesa) {
15             sError = "Excepcion " + metodo para generarLog
16
17             AlertaUsuario(sError)
18         }
19     }
20 }
```

Figura 9. Pseudocódigo del controlador. Fuente: Elaboración propia

Para el modelo se utilizan clases DAL (Capa de acceso a datos) las cuales se almacenan en un proyecto nuevo (dentro de la misma solución) denominado "Capa de negocio" estas clases contienen métodos que al establecer una conexión momentánea a base de datos y permiten realizar peticiones llamando a los procedimientos almacenados, si estos necesitan de parámetros se le otorgan por medio de los modelos (clases que contienen los getters y setters

de las propiedades que se desean enviar u obtener de la base de datos), con esto se obtiene una respuesta y se determina una acción.

Uno de los ejemplos más comunes de la función de clases DAL en el desarrollo del sistema, es utilizar un método de la clase de tipo bool para traer los atributos de un objeto y recorrer las filas que arroja un procedimiento almacenado, las filas llenan el modelo que es pasado por referencia y si todo el proceso es exitoso, retorna true, lo cual le permite al controlador realizar la lógica de la vista con esos datos, por mencionar algunas de las interpretaciones de estos, se encuentran:

- Poblar una tabla de datos
- Crear piezas de interfaz como tarjetas por cada registro
- Guardar datos de un formulario

A continuación, se muestra un ejemplo de cómo opera la clase de conexión a base de datos, sin embargo, se ejemplifican puntos esenciales, más no todas las validaciones, ya que dentro de cada clase y sus métodos existen más consideraciones importantes para evaluar los datos y capturar errores. El pseudocódigo de las clases DAL para generar el llenado del modelo es similar al de la Figura 10:

```

1
2  clase ConexionModulo {
3      metodo booleano ObtenerProducto(ref string sError, ref List < ClaseModelo > lstProductos) {
4          Intenta {
5              using(
6                  var resultadoBD = PeticionBaseDatos("NombreProcedimientoAlmacenado", argumentos)
7                  Mientras(resultadoBD["ClaveDeErrorBD"] = 0) {
8                      instancia ClaseModelo lst
9
10                     lstProductos.propiedad = resultadoBD["valorBD"]
11
12                     lst.Agregar(lstProductos)
13                 }
14                 return true
15             )
16         } CapturaError (Exception errorEnLaPromesa) {
17             sError = errorEnLaPromesa + metodo para generarLog
18             return false
19         }
20     }
21 }
    
```

Figura 10. Pseudocódigo de clase conexión DAL. Fuente: Elaboración propia

3.4 Pruebas

La implementación de pruebas end-to-end se lleva a cabo mediante Playwright para .NET. Este enfoque implica la automatización de interacciones de usuario simuladas para evaluar la funcionalidad integral del sistema.

En la redacción de los scripts de prueba en lenguaje .NET, se definen escenarios de usuario que abarcan áreas críticas del sistema. Estas pruebas abordan diversas facetas del sistema, desde la autenticación y navegación entre páginas hasta la interacción con formularios y la validación de datos. La ejecución de pruebas también se realiza en múltiples navegadores como Safari (para Windows y MacOS), Chrome y Firefox, para garantizar la coherencia en la experiencia del usuario final.

Particularmente, se verifica la funcionalidad específica del proyecto, como los módulos CRM y ERP, mediante la automatización de procesos empresariales clave, como la alta de pacientes y campañas y la gestión de inventario.

En términos de mantenimiento continuo, se enfatiza la necesidad de actualizar y ajustar las pruebas conforme la aplicación evoluciona, asegurando una cobertura exhaustiva y efectiva a lo largo del ciclo de vida del desarrollo del software. Este enfoque integral de pruebas end-to-end con Playwright para .NET se alinea con las mejores prácticas de aseguramiento de calidad, contribuyendo a la entrega de un sistema confiable y de alto rendimiento.

Capítulo IV

Resultados

A continuación, se desglosa la metodología aplicada al proyecto, describiendo las etapas conforme al marco metodológico propuesto. Esta etapa describe de manera detallada las fases a nivel práctico; tablas, diagramas, interfaces de usuario y seguimientos de cada requisito con base en la metodología de desarrollo y los tiempos dictaminados en el cronograma de actividades.

4.1 Seguimiento De Versiones

En la Tabla 3 se detalla el historial de cada una de las revisiones por el asesor de proyecto, para determinar los puntos de estado en cada fase del diseño, modelado y desarrollo, considerando la versión y las fechas de cada registro y donde estas empatan con el plan propuesto en el cronograma.

Tabla 3. Seguimiento de versiones del sistema. Fuente: Elaboración propia

Versión	Detalles	Fecha
1.0	Primera versión de planes de diseño y flujos del sistema (wireframes y diagramas) para su posterior codificación	27/09/2023
1.0	Primera versión del proyecto web que implementa la organización de carpetas y estructura los primeros archivos siguiendo la nomenclatura de orden que se detalla en el patrón MVC	04/10/2023
1.0	Primera versión del sistema web que cuenta con conexión a base de datos en SQL Server y un login de usuarios con validación de perfiles, flujo de recuperación de contraseña y controlador de módulos, que permite visualizar una sección u otra según sea el perfil de usuario	23/10/2023
1.0	Primera versión del módulo para la gestión de inventario con tarjetas que contengan los enlaces a los .aspx específicos	26/10/2023

Tabla 3 (continuación). Seguimiento de versiones del sistema. Fuente: Elaboración propia

Versión	Detalles	Fecha
2.0	Segunda versión del módulo para la gestión de inventario con apartados terminados	17/11/2023
1.0	Primera versión de módulo de gestión CRM con tarjetas que contengan los enlaces a los .aspx específicos	21/11/2023
2.0	Segunda versión de módulo de gestión CRM con apartados terminados	18/12/2023

4.2 Análisis, Diseño Y Modelado De La Estructura

4.2.1 Product Backlog

Se lleva a cabo una investigación de las necesidades que serán cubiertas por el sistema, platicando directamente con los empleados de la empresa, encargados de almacén y gestores de CRM para determinar cuál es el proceso actual y como beneficia la implementación de un sistema web.

- “Como empleados de una empresa que apunta al desarrollo ágil y sistematizado, es de suma importancia contar con una herramienta que permita a todo el equipo de trabajo poder atender procesos donde las tecnologías modernas jueguen un papel clave a la hora de llevar un control y aumentar la productividad”.

4.2.2 Sprint Backlog

Al realizar la investigación, se obtienen un listado de las actividades. Se comparan los requerimientos del sistema con base a los resultados de la investigación, se asignan tiempos, responsable y prioridades a las actividades.

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE COACALCO

Tabla 4. Spring backlog 1 del Análisis, diseño y modelado de la estructura. Fuente: Elaboración propia

Sprint 1				
Objetivo: Comprender las necesidades del cliente y plasmar las funciones del sistema con un levantamiento de requerimientos mediante el análisis				
Número	Requerimiento	Responsable	Tiempo	Prioridad
1.1	Junta con el cliente y el asesor de proyecto para conocer las reglas de negocio, los alcances y limitaciones con base a las necesidades planteadas	Cázares Pérez Josué Isaías	10 horas	Muy alta
1.2	Reunión con el asesor de proyecto para corroborar la lógica de negocio y funciones del sistema web	Cázares Pérez Josué Isaías	6 horas	Normal

Tabla 5. Spring backlog 2 del Análisis, diseño y modelado de la estructura. Fuente: Elaboración propia

Sprint 2				
Objetivo: Configurar la base de datos con el desarrollo de esquemas, tablas, uniones y permisos				
Número	Requerimiento	Responsable	Tiempo	Prioridad
2.1	Modelo entidad relación para la base de datos del esquema inventario	Valencia Arcega Luis Angel	1 día	Muy alta
2.2	Creación de las tablas para el esquema inventario	Valencia Arcega Luis Angel	3 días	Muy alta
2.3	Modelo entidad relación para la base de datos y creación de las tablas para el esquema crm	Valencia Arcega Luis Angel	4 días	Muy alta

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE COACALCO

Tabla 6. Spring backlog 3 del Análisis, diseño y modelado de la estructura. Fuente: Elaboración propia

Sprint 3				
Objetivo: Diseñar los flujos, casos de uso y secuencia de las funciones en las interfaces del sistema y aplicar los principios del patrón MVC para una organización de calidad en el sistema de archivos				
Número	Requerimiento	Responsable	Tiempo	Prioridad
3.1	Diagramas de secuencia y de caso de uso	Cázares Pérez Josué Isaías	2 días	Alta
3.2	Definición de wireframes (bocetos de UI) para las interfaces, flujos normales y alternos para el módulo de inventario	Valencia Arcega Luis Angel	3 días	Muy alta
3.3	Definición de wireframes (bocetos de UI) para las interfaces, flujos normales y alternos para el módulo de CRM	Valencia Arcega Luis Angel	3 días	Muy alta
3.4	Estructura de carpetas y archivos siguiendo la nomenclatura de orden que se detalla en el patrón MVC	Valencia Arcega Luis Angel	6 horas	Normal

Tabla 7. Spring backlog 4 del Análisis, diseño y modelado de la estructura. Fuente: Elaboración propia

Sprint 4				
Objetivo: Buscar la aprobación del cliente bajo sus estándares de calidad para el modelado de base de datos, interfaces y organización del proyecto para poder dar seguimiento a la siguiente etapa				
Número	Requerimiento	Responsable	Tiempo	Prioridad
4.1	Realizar una reunión con el cliente para dar el visto bueno del modelado de base de datos	Valencia Arcega Luis Angel	3 horas	Muy alta

Tabla 7 (continuación). Spring backlog 4 del Análisis, diseño y modelado de la estructura

Número	Requerimiento	Responsable	Tiempo	Prioridad
4.2	Realizar los cambios propuestos en la reunión, así como anotar las precauciones de cada uno de los modelos (restricciones de permisos y privilegios de usuarios en tablas)	Valencia Arcega Luis Angel	5 horas	Muy alta

4.2.3 Burndown Chart

En la Figura 11 se grafican la cantidad de días que fueron requeridos para realizar cada uno de los sprint, donde se visualiza que algunos de ellos tomaban diferente cantidad de días dependiendo del nivel de dificultad de cada tarea, así como de la prioridad.



Figura 11. Burndown chart Días por sprint. Fuente: Elaboración propia

4.2.4 Definition Of Done

Se realiza las definiciones del sistema al ser implementado, detallando, si se cumplieron los criterios, o si se cuentan con observaciones obtenidas durante el desarrollo.

- El trabajo realizado ha sido revisado por el encargado de proyecto
- Algunos detalles de modelado se corrigen al momento de la última revisión
- Los apartados de análisis y diseño cumplen con los estándares de la empresa

4.3 Codificación Del Módulo De Inventario

4.3.1 Product Backlog

Se lleva a cabo una investigación en el área de inventario y su gestión, platicando directamente con los empleados de la empresa encargados de almacén para determinar cuál es el proceso actual y como beneficia la implementación de un sistema web.

- “Como encargado del área de inventario debo llevar los registros de manera manual, todas las bitácoras y catálogos están escritas de manera física, el uso de una herramienta en mi computadora me ayuda a llevar a cabo mis tareas de manera más rápida y centralizada, porque no debo llevar todos los registros en físico a las evaluaciones, juntas o surtimientos y perder tiempo buscando algún registro en particular”.

4.3.2 Sprint Backlog

Al realizar la investigación, se obtienen un listado de las actividades. Se comparan los requerimientos del sistema con base a los resultados de la investigación, se asignan tiempos, responsable y prioridades a las actividades.

Tabla 8. Spring backlog 1 de Codificación del módulo de inventario. Fuente: Elaboración propia

Sprint 1				
Objetivo: Conectar el sistema a base de datos y preparar los procedimientos almacenados necesarios para ser consumidos posteriormente, así como maquetar y programar las funciones de las pantallas del primer apartado del módulo de inventario; Productos y el inicio de sesión				
Número	Requerimiento	Responsable	Tiempo	Prioridad
1.1	Realizar la conexión al motor de base de datos SQL Server con el proyecto de .NET y realizar las operaciones CRUD con ayuda de procedimientos almacenados	Valencia Arcega Luis Angel	3 días	Muy alta
1.2	Maquetado y programación del login de usuario con validaciones	Valencia Arcega Luis Angel	2 días	Muy alta
1.3	Maquetado y programación de la pantalla de productos la cual muestra una tabla de registros con todos los productos en almacén, esto incluye productos internos de la empresa, como también los productos proporcionados por los laboratorios	Valencia Arcega Luis Angel	1 día	Alta

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE COACALCO

Tabla 8 (continuación). Spring backlog 1 de Codificación del módulo de inventario. Fuente: Elaboración propia

Número	Requerimiento	Responsable	Tiempo	Prioridad
1.4	Maquetado y programación de la pantalla de líneas la cual muestra una tabla de registros con todas las líneas a las que puede pertenecer un producto, por ejemplo, si es una laptop, su línea es “Equipos de cómputo”	Valencia Arcega Luis Angel	1 día	Alta
1.5	Maquetado y programación de la pantalla de proveedores la cual muestra una tabla de registros con todas las fuentes (laboratorios) que proveen productos a los almacenes	Valencia Arcega Luis Angel	1 día	Alta
1.6	Maquetado y programación de la pantalla de almacenes la cual muestra una tabla de registros con los almacenes donde se guardan todos los productos de la empresa, su estatus, fecha de último movimiento y observaciones	Valencia Arcega Luis Angel	1 día	Alta

Tabla 9. Spring backlog 2 de Codificación del módulo de inventario. Fuente: Elaboración propia

Sprint 2				
Objetivo: Maquetar y programar las funciones de las pantallas del segundo apartado del módulo de inventario; Catálogo de conceptos				
Número	Requerimiento	Responsable	Tiempo	Prioridad
2.1	Maquetado y programación de la pantalla de movimiento entre almacenes la cual muestra una tabla de registros en forma de bitácora con todos los motivos de movimientos entre almacenes	Valencia Arcega Luis Angel	1 día	Alta

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE COACALCO

Tabla 9 (continuación). Spring backlog 2 de Codificación del módulo de inventario. Fuente: Elaboración propia

Número	Requerimiento	Responsable	Tiempo	Prioridad
2.2	Maquetado y programación de la pantalla de movimiento entre colaboradores la cual muestra una tabla de registros en forma de bitácora con todos los motivos de movimientos entre empleados	Valencia Arcega Luis Angel	1 día	Alta
2.3	Maquetado y programación de la pantalla de traslados internos la cual muestra una tabla de registros en forma de bitácora con los conceptos o motivo de traslado, fecha y su estatus	Valencia Arcega Luis Angel	8 horas	Alta

Tabla 10. Spring backlog 3 de Codificación del módulo de inventario. Fuente: Elaboración propia

Sprint 3				
Objetivo: Maquetar y programar las funciones de las pantallas del tercer apartado del módulo de inventario; Almacén				
Número	Requerimiento	Responsable	Tiempo	Prioridad
3.1	Maquetado y programación de la pantalla de existencias por almacén la cual muestra una tabla de registros con los productos que existen en los almacenes y permite editar parámetros como el stock mínimo y stock máximo de productos, muestra un semáforo de registros	Valencia Arcega Luis Angel	6 horas	Alta

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE COACALCO

Tabla 10 (continuación). Spring backlog 3 de Codificación del módulo de inventario. Fuente: Elaboración propia

Número	Requerimiento	Responsable	Tiempo	Prioridad
3.2	Maquetado y programación de la pantalla de existencias por colaborador la cual muestra una tabla de registros con los productos que tiene cada empleado de la empresa (permite cambiar el punto de reorden) y muestra un semáforo de registros	Valencia Arcega Luis Angel	1 día	Alta
3.3	Maquetado y programación de la pantalla de traspaso entre almacenes la cual muestra un formulario que permita realizar traspasos de productos de un almacén a otro	Valencia Arcega Luis Angel	1 día	Alta
3.4	Maquetado y programación de la pantalla de traspaso entre colaboradores la cual muestra un formulario que permita realizar traspasos de productos de un empleado a otro	Valencia Arcega Luis Angel	1 día	Alta

Tabla 11. Spring backlog 4 de Codificación del módulo de inventario. Fuente: Elaboración propia

Sprint 4				
Objetivo: Crear el flujo de recuperación de contraseña con ayuda de los servicios externos y reunir los avances con la empresa para correcciones futuras				
Número	Requerimiento	Responsable	Tiempo	Prioridad
4.1	Realizar las configuraciones necesarias para los servicios de correo y mensajes de texto que función en la recuperación de contraseñas y maquetar el flujo de sus pantallas y alertas	Valencia Arcega Luis Angel	3 días	Muy alta
4.2	Realizar una reunión con el cliente para obtener retroalimentación sobre el módulo y posibles mejoras de diseño o compatibilidad	Cázares Pérez Josué Isaías	6 horas	Normal

4.3.3 Burndown Chart

Se grafican la cantidad de días que fueron requeridos para realizar cada uno de los sprint, donde se visualiza que algunos de ellos tomaban diferente cantidad de días dependiendo del nivel de dificultad de cada tarea, las actividades delegadas en el primer sprint suelen ser las que llevan más tiempo de desarrollo, debido al proceso de adaptación, ya bien de la tecnología o de comprender el flujo que se detalla en la etapa de diseño, una vez que estas han sido desarrolladas las siguientes iteraciones tienden a ser más rápidas, a excepción de sprints con requisitos más complejos o tardados.



Figura 12. Burndown chart Días por sprint. Fuente: Elaboración propia

4.3.4 Definition Of Done

Se realiza las definiciones del sistema hasta el punto exacto de revisión, detallando, si se cumplieron los criterios, o si se cuentan con observaciones obtenidas durante el desarrollo de cada sprint.

- El trabajo realizado ha sido revisado por el encargado de proyecto y el encargado de almacén
- El módulo de inventario cumple con los requisitos establecidos en el plan de diseño y cumple con los estándares de calidad

4.4 Codificación Del Módulo De CRM

4.4.1 Product Backlog

Se lleva a cabo una investigación en el área de CRM, platicando directamente con los empleados de la empresa encargados de esta área para determinar cuál es el proceso actual y como beneficia la implementación de un sistema web.

- “Como encargada de ver la relación que tiene nuestra empresa con los clientes, es de suma importancia conocer los procesos de altas, trámites administrativos y movimientos internos (como interpretaciones medicas), es por ello por lo que al momento de recibir una llamada y comenzar el proceso de alta, debo registrar los datos del paciente que llama y asignarlo a las campañas existentes actualmente y relacionando los productos del inventario con las necesidades de la campaña en cuestión, así como tomar datos de su médico tratante y asignarle un médico especialista”.

4.4.2 Sprint Backlog

Al realizar la investigación, se obtienen un listado de requisitos. Se comparan los requerimientos del sistema con base a los resultados de la investigación, se asignan tiempos, responsable y prioridades a las actividades.

Tabla 12. Spring backlog 1 de Codificación del módulo de CRM. Fuente: Elaboración propia

Sprint 1				
Objetivo: Preparar los procedimientos almacenados necesarios para ser consumidos posteriormente, así como maquetar y programar las funciones de las pantallas del primer apartado del módulo de CRM; Paciente				
Número	Requerimiento	Responsable	Tiempo	Prioridad
1.1	Realizar las operaciones CRUD restantes con ayuda de procedimientos almacenados	Valencia Arcega Luis Angel	4 días	Muy alta
1.2	Maquetado y programación de la pantalla de alta la cual muestra un formulario que permite al personal encargado de área a dar de alta a un nuevo paciente al momento de la llamada	Valencia Arcega Luis Angel	1 día	Muy alta

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE COACALCO

Tabla 12 (continuación). Spring backlog 1 de Codificación del módulo de CRM. Fuente: Elaboración propia

Número	Requerimiento	Responsable	Tiempo	Prioridad
1.3	Maquetado y programación de la pantalla de expediente digital, la cual muestra una tabla en forma de bitácora con todos los estudios y campañas a los que ha pertenecido un paciente	Valencia Arcega Luis Angel	1 día	Normal

Tabla 13. Spring backlog 2 de Codificación del módulo de CRM. Fuente: Elaboración propia

Sprint 2				
Objetivo: Maquetar, poblar de datos y programar las funciones de las pantallas del segundo apartado del módulo de CRM; Catálogo				
Número	Requerimiento	Responsable	Tiempo	Prioridad
2.1	Maquetado y programación de la pantalla de consultor la cual muestra una tabla que lista a los consultores encargados de la campaña y la asignación de pacientes a campañas	Valencia Arcega Luis Angel	1 día	Normal
2.2	Maquetado y programación de la pantalla de médico especialista la cual muestra una tabla con todos los registros de los médicos que trabajan con mercurio para interpretar los estudios	Valencia Arcega Luis Angel	1 día	Alta
2.3	Maquetado y programación de la pantalla de médico tratante la cual muestra una tabla con todos los registros de los médicos externos que mandar a realizar los estudios	Valencia Arcega Luis Angel	1 día	Normal

Tabla 14. Spring backlog 3 de Codificación del módulo de CRM. Fuente: Elaboración propia

Sprint 3				
Objetivo: Maquetar y programar las funciones de la pantalla del segundo apartado en el módulo de CRM; Catálogo – Campañas, así como validar los requisitos del módulo y comenzar las pruebas del sistema				
Número	Requerimiento	Responsable	Tiempo	Prioridad
3.1	Maquetado y programación de la pantalla de campañas la cual muestra una tabla que lista todas las campañas disponibles de los clientes de la empresa	Valencia Arcega Luis Angel	3 días	Muy alta
3.2	Junta con el cliente y el asesor de proyecto para la validación de requisitos y retroalimentación de mejoras en el módulo	Valencia Arcega Luis Angel	6 horas	Alta

Tabla 15. Spring backlog 4 de Codificación del módulo de CRM. Fuente: Elaboración propia

Sprint 4				
Objetivo: Realizar pruebas de funcionamiento al sistema, comenzando con el flujo normal de las pantallas, con énfasis en el inicio de sesión, recuperación de credenciales y validaciones de páginas				
Número	Requerimiento	Responsable	Tiempo	Prioridad
4.1	Pruebas a nivel de vista; cada pantalla se revisa a detalle con el objetivo de puntualizar temas de diseño, colores, fuentes, consistencia de la interfaz y flujos de cada sección, así como <i>feedback</i> del usuario como alertas o flujos alternos y normales	Valencia Arcega Luis Angel	3 días	Muy alta
4.2	Pruebas de estrés; cada flujo pasa por un proceso donde se someten a las páginas a una gran cantidad de peticiones de manera simultánea, para comprobar su velocidad de carga y capacidad para albergar múltiples usuarios al mismo tiempo, sin perder el estado, el contexto y el rendimiento	Valencia Arcega Luis Angel	2 días	Muy alta
4.3	Pruebas de acceso a la información; los flujos donde los usuarios ingresan actualizan o recuperan su información, deben tener las validaciones adecuadas para generar robustez y seguridad en cada proceso, por ejemplo; <i>Las pantallas de inicio de sesión donde se deben probar diferentes perfiles y credenciales incorrectas, así como intentos de acceso a rutas protegidas.</i>	Valencia Arcega Luis Angel	2 días	Muy alta

4.4.3 Burndown Chart

En la Figura 13 se grafican la cantidad de días que fueron requeridos para realizar cada uno de los sprint del mes, en el caso de esta etapa, el ultimo sprint toma más tiempo debido a que se comienza la etapa de pruebas donde se validan todas y cada una de las pantallas y flujos de manera visual e interna, como pruebas de rendimiento, una primera etapa de pruebas de seguridad y consistencia de las interfaces y los accesos.



Figura 13. Burndown chart Días por sprint. Fuente: Elaboración propia

4.4.4 Definition Of Done

Se realiza las definiciones del sistema hasta el punto exacto de revisión, detallando, si se cumplieron los criterios, o si se cuentan con observaciones.

- El trabajo realizado ha sido revisado por el encargado de proyecto y el encargado del área de CRM
- El módulo CRM logró cumplir de manera acertada con los requisitos establecidos en el plan de diseño y cumple con los estándares de calidad y robustez
- Los resultados se midieron con base a juntas de trabajo donde se detallaban los requisitos completados de cada tarea

4.5 Pruebas, Implementación Y Capacitación

4.5.1 Product Backlog

Se lleva a cabo una investigación sobre el funcionamiento del sistema para garantizar la calidad y su correcta entrega a los usuarios finales sin enfrentar problemas típicos de compatibilidad o mala interpretación de su uso.

- El software debe ser probado de forma exhaustiva antes de su implementación.
- La implementación del software debe ser realizada de forma correcta y segura.
- Los usuarios finales deben ser capacitados en el uso del software.

4.5.2 Sprint Backlog

Al realizar la investigación, se obtienen un listado de requisitos. Se comparan los requerimientos del sistema con base a los resultados de la investigación, se asignan tiempos y se contabilizan para el registro de días, un responsable por tarea y prioridades a cada una de las actividades.

Tabla 16. Spring backlog 1 de Pruebas, implementación y capacitación. Fuente: Elaboración propia

Sprint 1				
Objetivo: Realizar las pruebas de base de datos (procedimientos almacenados e integridad de las tablas) y servicios externos				
Número	Requerimiento	Responsable	Tiempo	Prioridad
1.1	Realizar pruebas de SQL a nivel de base de datos, para comprobar la robustez de las tablas y procedimientos, esto con tareas y rutinas de inserción, modificación y lectura de datos, para probar peticiones a las tablas y su velocidad de respuesta e integridad.	Valencia Arcega Luis Angel	4 días	Alta

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE COACALCO

Tabla 16 (continuación). Spring backlog 1 de Pruebas, implementación y capacitación. Fuente: Elaboración propia

Número	Requerimiento	Responsable	Tiempo	Prioridad
1.2	Realizar pruebas del servicio de correos electrónicos que permite a los usuarios restablecer su contraseña y el WebHook (un tipo de API que genera un disparador cuando cambia su estado) de llamadas	Valencia Arcega Luis Angel	2 días	Muy alta

Tabla 17. Spring backlog 2 de Pruebas, implementación y capacitación. Fuente: Elaboración propia

Sprint 2				
Objetivo: Realizar las pruebas del sistema considerando aspectos básicos de un sistema web, tales como SEO, usabilidad, accesibilidad y compatibilidad				
Número	Requerimiento	Responsable	Tiempo	Prioridad
2.1	Realizar pruebas end-to-end con Playwright para canalizar datos de SEO, usabilidad, accesibilidad y buenas prácticas generales	Valencia Arcega Luis Angel	3 días	Muy alta
2.2	Realizar pruebas con Lighthouse y diagnosticar parámetros faltantes como descripciones en los metadatos, textos alternativos y consideraciones de diseño como colores con buen contraste de lectura y un sitio web inclusivo	Valencia Arcega Luis Angel	3 días	Alta
2.2	Realizar pruebas de compatibilidad del sitio, considerando la funcionalidad y la consistencia del diseño en múltiples navegadores, ya que no todos utilizan el mismo motor de JavaScript y así evitar problemas de <i>cross-browsing</i>	Valencia Arcega Luis Angel	1 día	Normal

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE COACALCO

Tabla 18. Spring backlog 3 de Pruebas, implementación y capacitación. Fuente: Elaboración propia

Sprint 3				
Objetivo: Realizar la implementación del sistema desplegando el sitio en el servidor y dejar un acceso directo o marcar en favoritos el enlace, en los equipos de cómputo de los trabajadores de la empresa, encargados de gestionar el ERP				
Número	Requerimiento	Responsable	Tiempo	Prioridad
3.1	Generar los archivos estáticos (HTML, CSS, JavaScript y contenido multimedia) del proyecto, esto con la finalidad de poder subir dichos archivos al servidor y hacer el despliegue	Valencia Arcega Luis Angel	6 horas	Muy alta
3.2	Guardar el sitio web como favorito en los navegadores de los empleados encargados de gestionar el ERP	Valencia Arcega Luis Angel	4 horas	Alta

Tabla 19. Spring backlog 4 de Pruebas, implementación y capacitación. Fuente: Elaboración propia

Sprint 4				
Objetivo: Realizar la capacitación de los usuarios de la empresa sobre el manejo de sistema y exponer todas las ventajas y requisitos resueltos en el proceso de desarrollo				
Número	Requerimiento	Responsable	Tiempo	Prioridad
4.1	Capacitar a los usuarios en todos los detalles esenciales del sistema, su uso, sus alcances y limitaciones	Valencia Arcega Luis Angel	4 días	Muy alta
4.2	Resolver dudas particulares y generales del sistema	Valencia Arcega Luis Angel	5 horas	Muy alta

4.5.3 Burndown Chart

Se grafican la cantidad de días que fueron requeridos para realizar cada uno de los sprint de la última etapa, los dos primeros sprints tomaron más tiempo, debido a que son la continuación de pruebas para el sistema, donde se probaron servicios externos, procedimientos almacenados y detalles finales de experiencia de usuario, el tercer sprint representa la etapa de implementación y despliegue la cual no toma tanto tiempo como la capacitación o las pruebas.



Figura 14. Burndown chart Días por sprint. Fuente: Elaboración propia

4.5.4 Definition Of Done

Se realiza las revisiones del sistema hasta el punto exacto de la última etapa, detallando, si se cumplieron los criterios y los requisitos planteados.

- El trabajo realizado ha sido revisado por el encargado de proyecto
- Los módulos y el sistema en general lograron cumplir de manera acertada con los requisitos establecidos en el plan de diseño y cumple con los estándares de calidad
- Los resultados se midieron con base a juntas y reuniones de capacitación para lograr la adaptación del sistema y empatar las necesidades con el producto final

4.6 Interfaces De Usuario

La primer pantalla que se muestra al cargar el sistema es para iniciar sesión, aquí se debe ingresar el correo electrónico y la contraseña que se le asignan al trabajador al momento de la alta, esto con la finalidad de poder ingresar al sistema y seguir el flujo para la manipulación del módulo, ya sea de inventario o CRM según sea su perfil.

Además del inicio de sesión el usuario puede dar clic al enlace de ‘No recuerdo mi contraseña’ para ir al proceso de restablecimiento en caso de ser necesario. Por último, están los enlaces a las redes sociales de la empresa.



Figura 15. Interfaz de inicio de sesión. Fuente: Elaboración propia

En caso de que el usuario al querer iniciar sesión olvide su contraseña, da clic en el enlace mencionado anteriormente y carga la siguiente página:

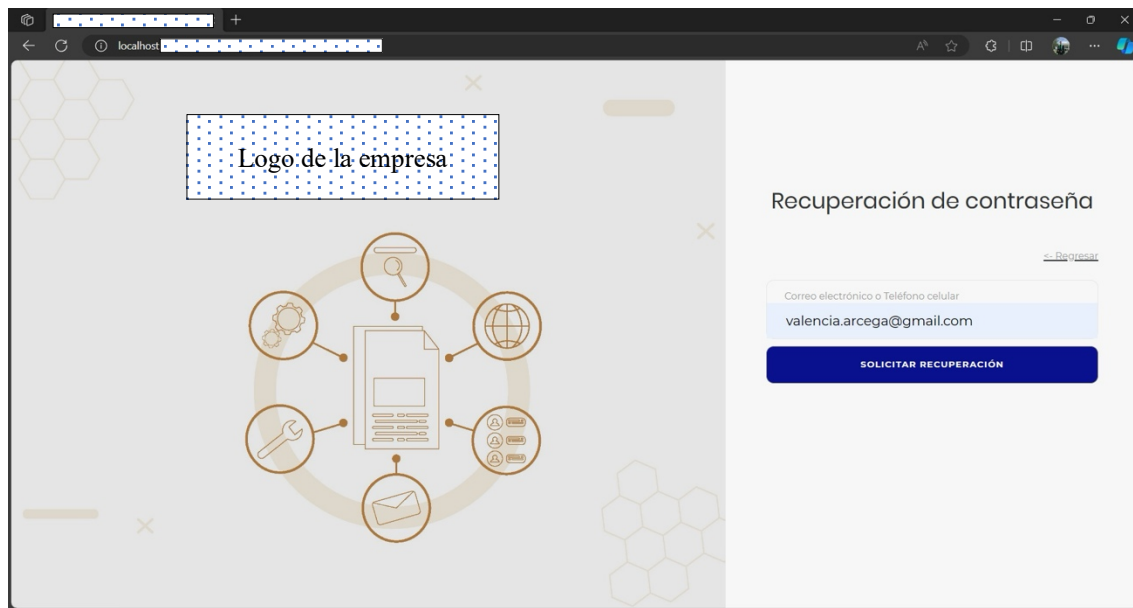


Figura 16. Interfaz para enviar código de recuperación. Fuente: Elaboración propia

Al dar clic al botón de ‘SOLICITAR RECUPERACIÓN’ envía un correo electrónico o un mensaje de texto al usuario con el código para restablecer su contraseña y muestra una alerta de los pasos a seguir:

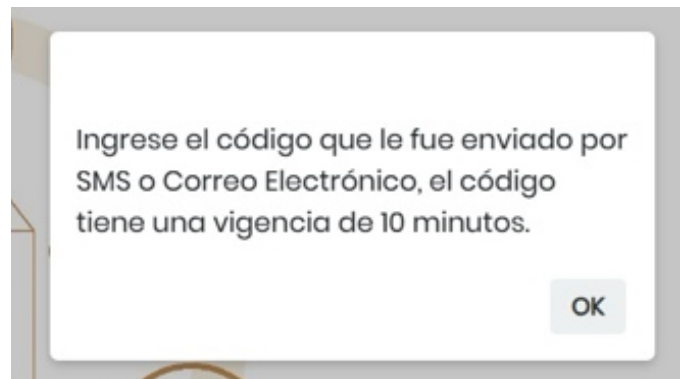


Figura 17. Alerta de código enviado al usuario. Fuente: Elaboración propia

Llega el código al correo y es este, el que debe ser ingresado en la nueva pantalla

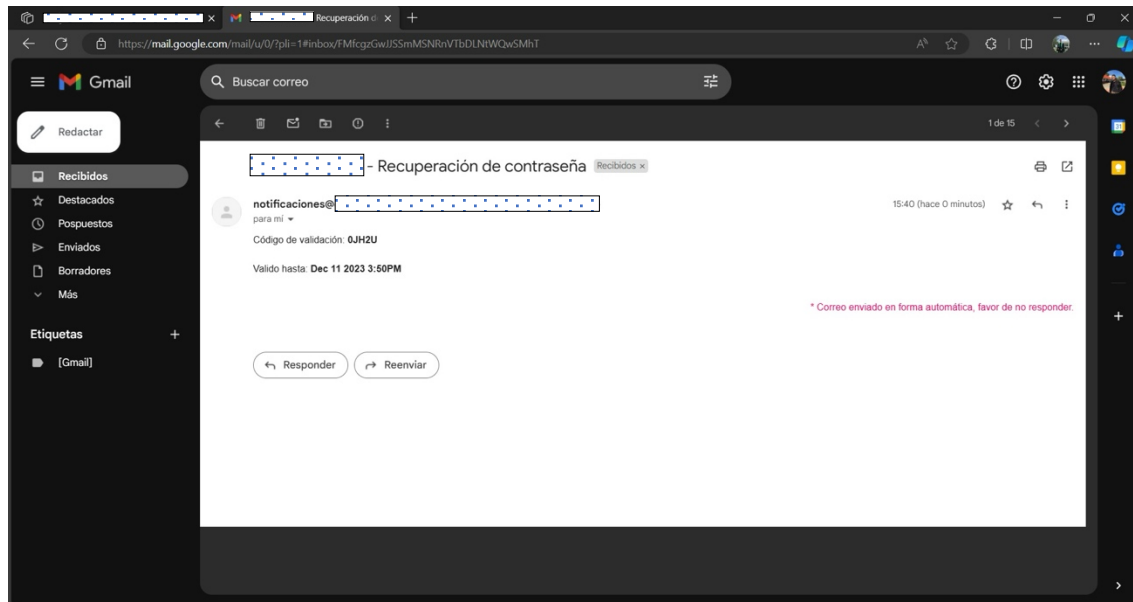


Figura 18. Bandeja de correo con el código de recuperación. Fuente: Elaboración propia

Una vez que el usuario verifica el código de seguridad y da en el botón ‘VALIDAR CÓDIGO’ le muestra la siguiente alerta y un nuevo campo para ingresar su nueva contraseña:

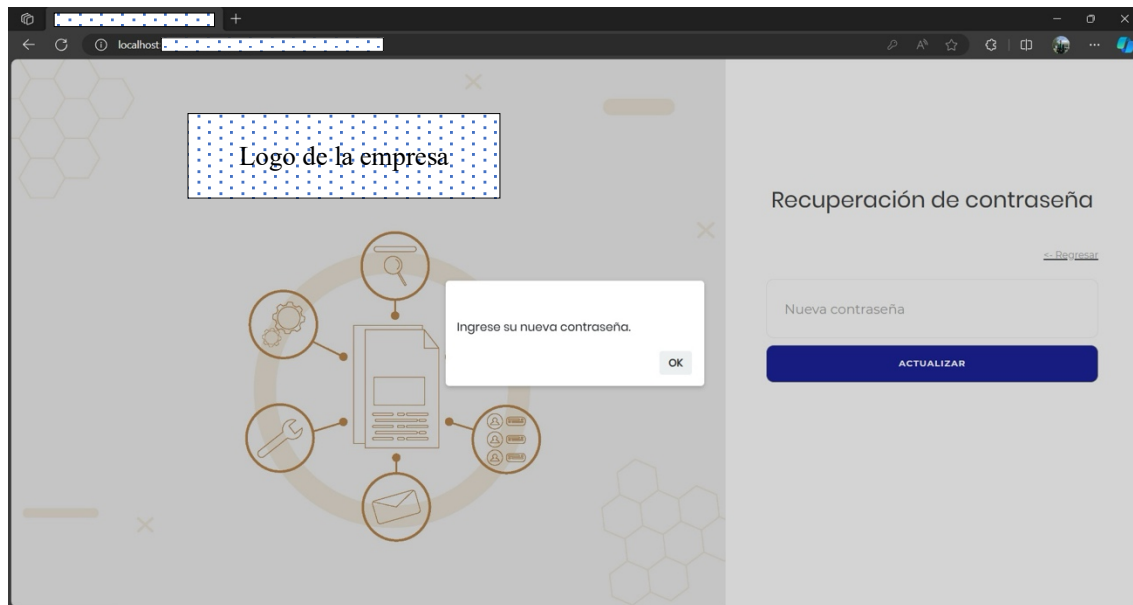


Figura 19. Alerta de código valido y campo de nueva contraseña. Fuente: Elaboración propia

Al dar clic en el botón ‘ACTUALIZAR’, si la contraseña cumple con los parámetros mínimos de seguridad le muestra una alera de éxito con los pasos a seguir:

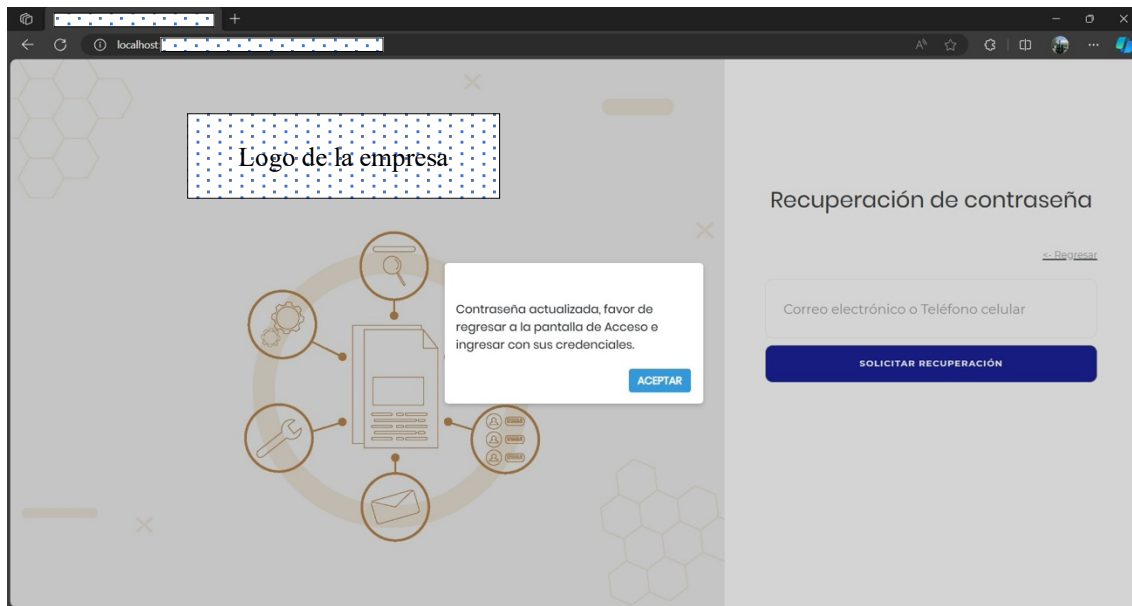


Figura 20. Alerta contraseña restablecida exitosamente. Fuente: Elaboración propia

Al momento de iniciar sesión, en caso de que las credenciales empaten con un usuario que tiene el perfil de administrador de inventario, la SideBar muestra el botón de ‘Cloud’:

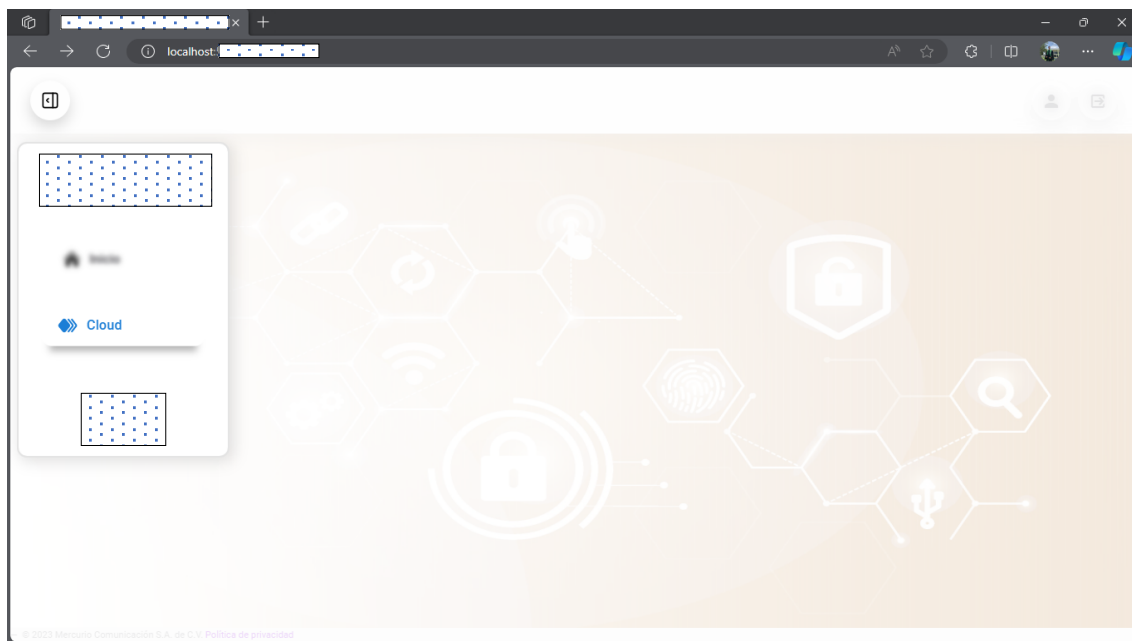


Figura 21. Interfaz de pantalla principal, gestor de inventario. Fuente: Elaboración propia

Nota. La razón por la que el módulo de inventario se denomina como “Cloud” es debido a la imperante necesidad de extender este sistema a largo plazo, teniendo como resultado la anexión de futuros módulos al ERP tales como el control y la gestión de finanzas, recursos humanos, entre otros. Por lo que, en vez de directamente llevar al módulo de control de inventario exista una pantalla intermedia que nos permita agregar más módulos según sea la necesidad a futuro. Pasa algo similar con el botón que nos permite ir al control del módulo CRM, debido a que el enfoque está dirigido en mayor medida a los clientes de la empresa, es importante considerar que cada uno de ellos debe tener su sección específica, por lo que debe tener un accionador que lleve a su dashboard particular. En este caso, el laboratorio Novartis que pertenece a uno de los clientes de la empresa aparece en la SideBar, de ser el caso de otro de los clientes como Bayer se anexaría un nuevo botón de igual manera.

El mismo escenario donde las credenciales correspondan al perfil de gestor de CRM se muestra la SideBar de un usuario el cual es capaz de entrar y manipular solo el módulo de CRM y los diferentes clientes que se encuentren activos.

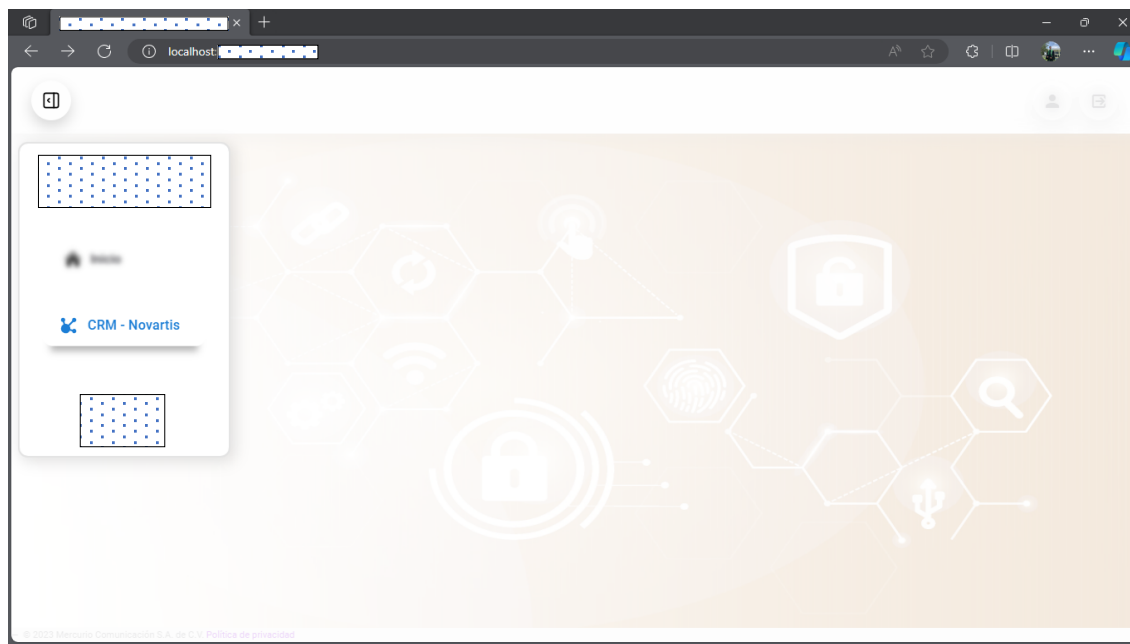


Figura 22. Interfaz de pantalla principal, gestor de CRM. Fuente: Elaboración propia

Una vez que se tiene acceso al dashboard de gestión de inventario, el encargado de esta área puede manipular las secciones del módulo, las cuales incluyen catálogos, bitacoras y formularios para manipular la información de los productos dentro de los almacenes:

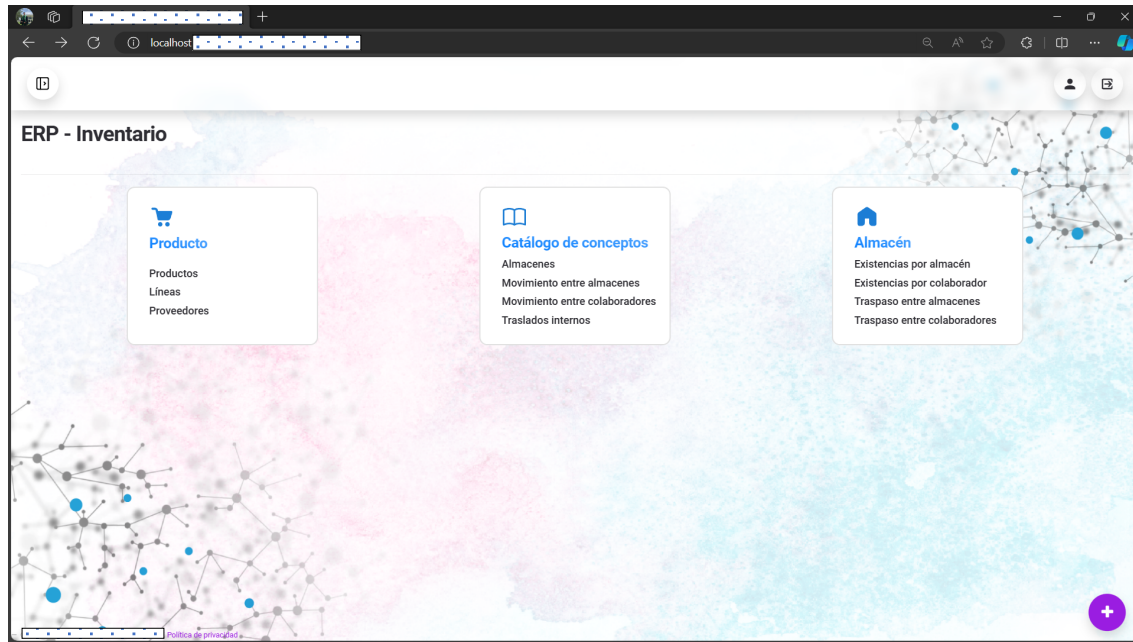
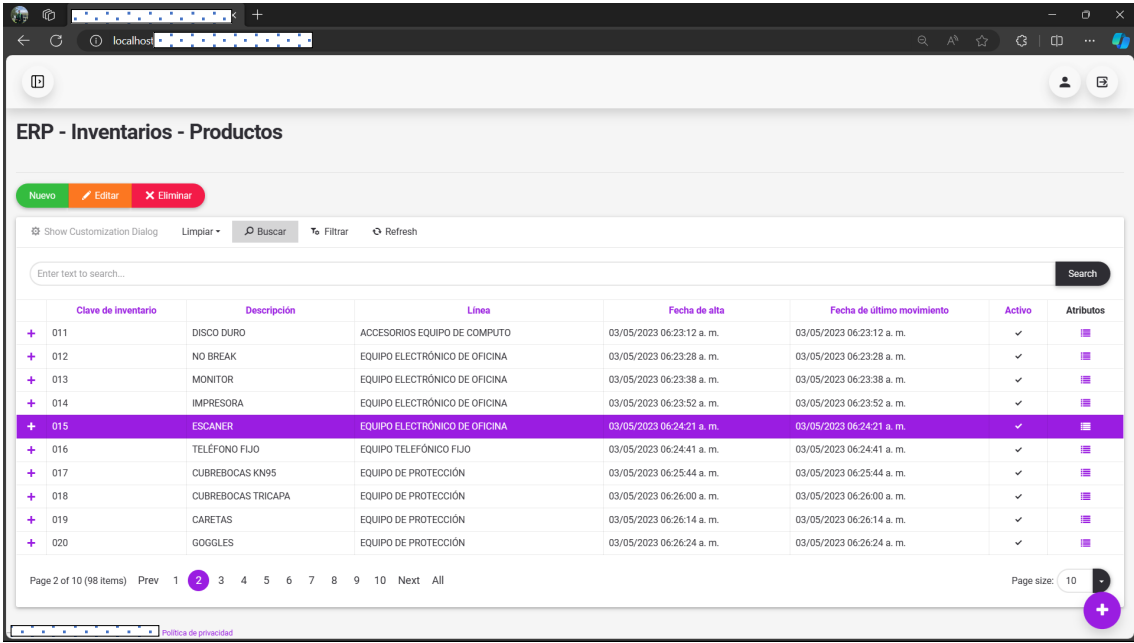


Figura 23. Interfaz de dashboard inventario. Fuente: Elaboración propia

La primer pantalla muestra un grid donde se listan todos los productos de los inventarios, la línea a la que pertenecen, fechas, estatus y una columna para ver más detalles del producto:



ERP - Inventarios - Productos

Nuevo Editar Eliminar

Show Customization Dialog Limpiar Buscar Filtrar Refresh

Enter text to search... Search

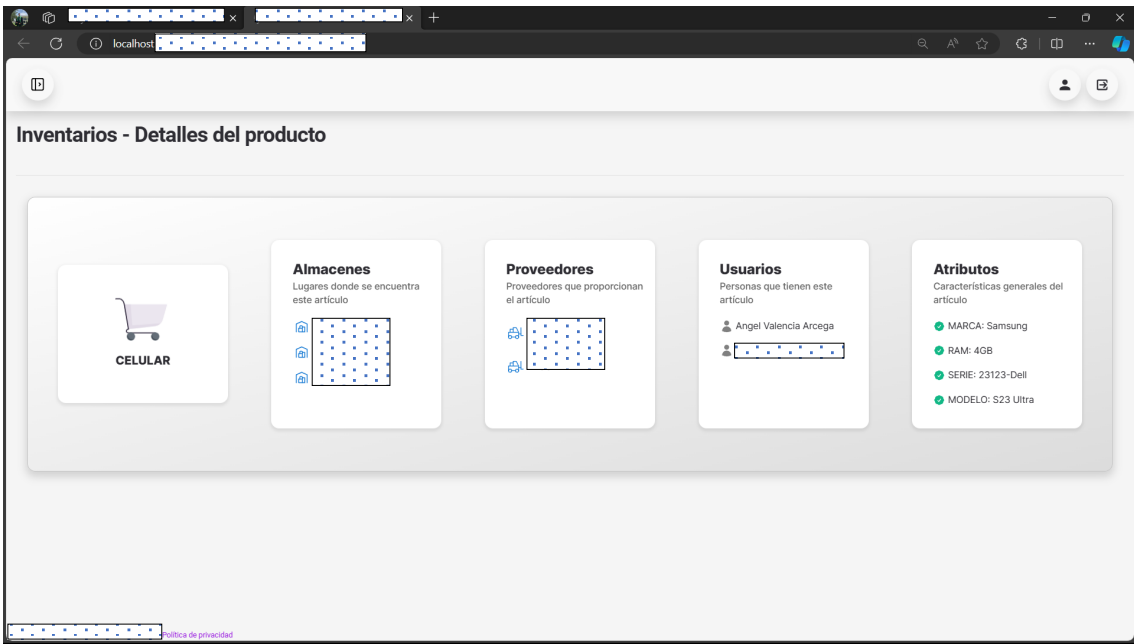
	Clave de inventario	Descripción	Línea	Fecha de alta	Fecha de último movimiento	Activo	Atributos
+	011	DISCO DURO	ACCESORIOS EQUIPO DE COMPUTO	03/05/2023 06:23:12 a. m.	03/05/2023 06:23:12 a. m.	✓	
+	012	NO BREAK	EQUIPO ELECTRÓNICO DE OFICINA	03/05/2023 06:23:28 a. m.	03/05/2023 06:23:28 a. m.	✓	
+	013	MONITOR	EQUIPO ELECTRÓNICO DE OFICINA	03/05/2023 06:23:38 a. m.	03/05/2023 06:23:38 a. m.	✓	
+	014	IMPRESORA	EQUIPO ELECTRÓNICO DE OFICINA	03/05/2023 06:23:52 a. m.	03/05/2023 06:23:52 a. m.	✓	
+	015	ESCANNER	EQUIPO ELECTRÓNICO DE OFICINA	03/05/2023 06:24:21 a. m.	03/05/2023 06:24:21 a. m.	✓	
+	016	TELÉFONO FIJO	EQUIPO TELEFÓNICO FIJO	03/05/2023 06:24:41 a. m.	03/05/2023 06:24:41 a. m.	✓	
+	017	CUBREBOCAS KN95	EQUIPO DE PROTECCIÓN	03/05/2023 06:25:44 a. m.	03/05/2023 06:25:44 a. m.	✓	
+	018	CUBREBOCAS TRICAPA	EQUIPO DE PROTECCIÓN	03/05/2023 06:26:00 a. m.	03/05/2023 06:26:00 a. m.	✓	
+	019	CARETAS	EQUIPO DE PROTECCIÓN	03/05/2023 06:26:14 a. m.	03/05/2023 06:26:14 a. m.	✓	
+	020	GOGGLES	EQUIPO DE PROTECCIÓN	03/05/2023 06:26:24 a. m.	03/05/2023 06:26:24 a. m.	✓	

Page 2 of 10 (98 items) Prev 1 2 3 4 5 6 7 8 9 10 Next All Page size: 10

Política de privacidad

Figura 24. Interfaz de productos. Fuente: Elaboración propia

De manera particular, si se hace clic en el botón de detalles de un determinado producto, abre una página donde muestra un panel de características más detalladas y específicas de ese producto, características tales como, los almacenes donde se encuentra el producto, sus proveedores, los usuarios que lo tienen en uso y algunos de sus atributos



Inventarios - Detalles del producto

CELULAR

Almacenes
Lugares donde se encuentra este artículo

Proveedores
Proveedores que proporcionan el artículo

Usuarios
Personas que tienen este artículo

Angel Valencia Arcega

Atributos
Características generales del artículo

- MARCA: Samsung
- RAM: 4GB
- SERIE: 23123-Dell
- MODELO: S23 Ultra

Política de privacidad

Figura 25. Interfaz de detalle de producto. Fuente: Elaboración propia

Al regresar a la pantalla de productos, se tiene la opción de crear nuevo registro (para el caso específico de esta pantalla se muestra la alta de un producto, pero todas aquellas pantallas que cuentan con el mismo botón en la parte superior izquierda hacen la misma función)

Atributo	Valor
MARCA	APPLE
MODELO	IPHONE XR
NO. DE SERIE	2139128390
IMEI	232131238978712837
CARGADOR	SI
FECHA DE COMPRA	23-12-23

Figura 26. Interfaz de alta de producto. Fuente: Elaboración propia

De igual manera el gestor de inventario puede editar algunos de los detalles de un producto como la línea a la que pertenece, su estatus y descripción:

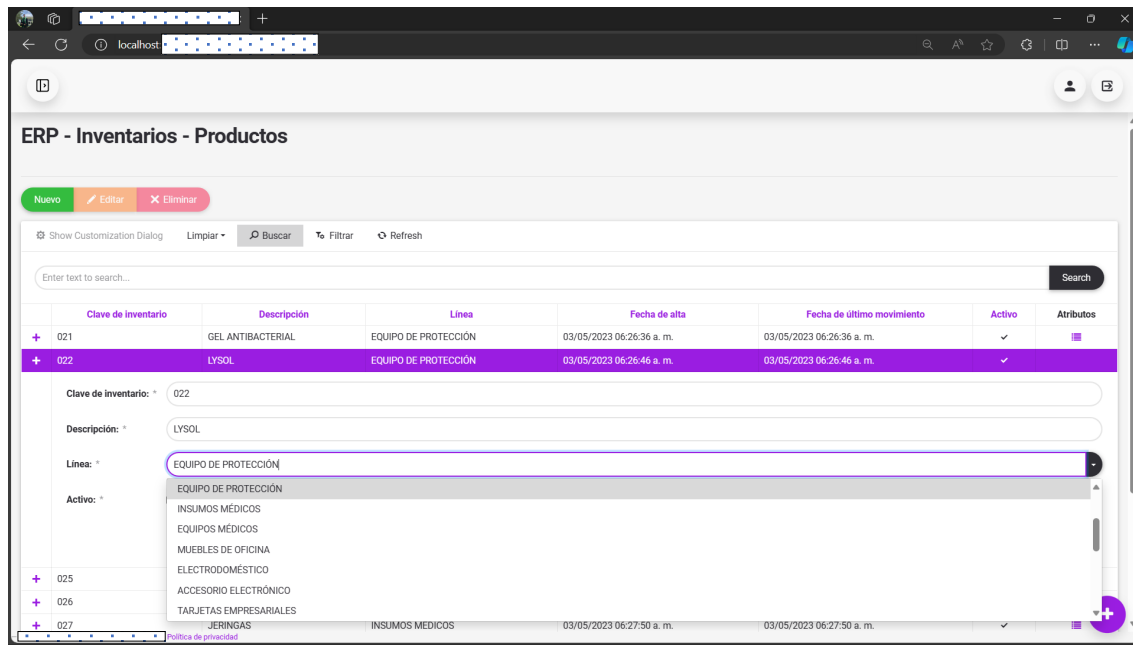


Figura 27. Interfaz de edición de productos. Fuente: Elaboración propia

La ultima sección del dashboard inventario corresponde a los almacenes

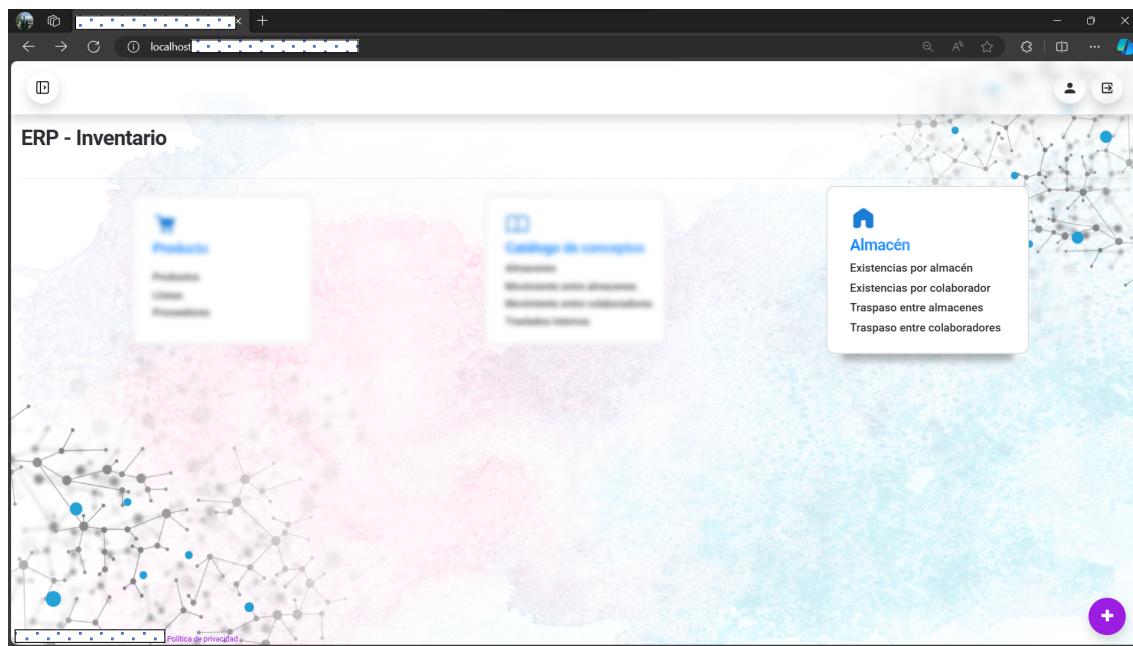


Figura 28. Interfaz de dashboard inventario - hover en almacén. Fuente: Elaboración propia

La pantalla de existencias por almacen muestra de todos aquellos productos que se encuentran en distintos almacenes, su cantidad total, lo mínimo y maximo de ese producto que debe haber en existencia y el calculo de punto de reorden (es un nivel predefinido de un artículo específico, que indica cuándo se debe solicitar más o menos unidades de un artículo)

ERP - Inventarios - Existencias por almacén

Semáforo de registros
 ● La cantidad está en orden
 ● La cantidad supera el inventario máximo
 ● La cantidad no alcanza el inventario mínimo

Editar Punto Reorden

Show Customization Dialog Limpiar Buscar Filtrar Refresh

Drag a column header here to group by that column

Almacén	Semáforo	Inventario	Cantidad	Inventario mínimo	Inventario máximo	Punto de reorden	Fecha de alta	Atributos
027 - JERINGAS	●	027 - JERINGAS				0	28/09/2023 12:00 a. m.	
020 - GOGGLES	●	020 - GOGGLES				0	28/09/2023 12:00 a. m.	
26359866 - LAPTOP	●	26359866 - LAPTOP				0	28/09/2023 11:51 a. m.	
26359866 - LAPTOP	●	26359866 - LAPTOP				0	28/09/2023 11:52 a. m.	
003 - CELULAR	●	003 - CELULAR				0	28/09/2023 12:06 p. m.	
006 - TECLADO	●	006 - TECLADO				0	28/09/2023 02:39 p. m.	
009 - DIADEMA	●	009 - DIADEMA				74	29/09/2023 09:26 a. m.	
003 - CELULAR	●	003 - CELULAR				0	29/09/2023 10:12 a. m.	
003 - CELULAR	●	003 - CELULAR				0	29/09/2023 10:26 a. m.	
027 - JERINGAS	●	027 - JERINGAS				0	29/09/2023 11:01 a. m.	

Page 1 of 5 (46 items) Prev 1 2 3 4 5 Next All

Política de privacidad

Page size: 10

Figura 29. Interfaz de existencias por almacén. Fuente: Elaboración propia

Para el módulo de CRM se presentan las dos secciones a las que puede acceder el gestor de este módulo:

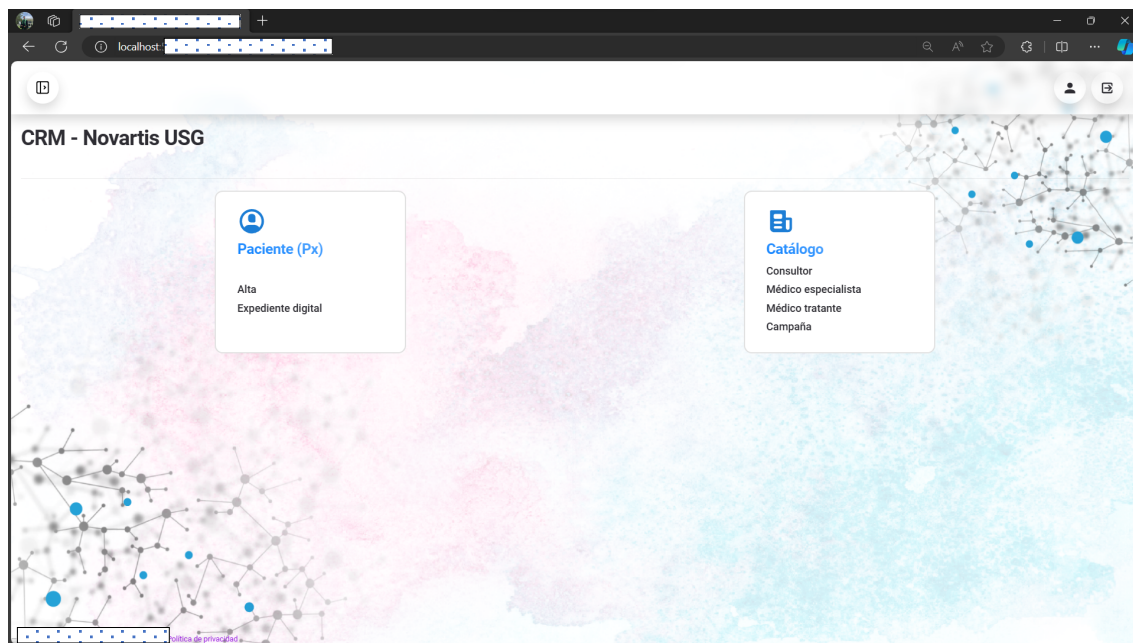


Figura 30. Interfaz de dashboard principal gestor de CRM. Fuente: Elaboración propia

La pantalla de alta permite a la coordinadora de CRM agregar un nuevo paciente, para lo que debe llenar todo el formato con los datos que le proporcionen durante la llamada, esta pantalla

es la responsable de realizar todas las uniones de la información del usuario, por detrás, la lógica de negocio en la programación con C# y los procedimientos almacenados con SQL Server permiten realizar estas rutinas que acortan el tiempo del administrador, de tener que llenar todo a mano sin poder llegar a tener clara la relación entre el paciente, sus estudios, el médico y la campaña involucrada.

Las figuras 31, 32, 33, 34 y 35 muestran el flujo de la alta de un paciente y se asignan su respectivas relaciones con base a las preguntas que realiza el consultor al momento de tomar la llamada, los pacientes brindan su información para poder ser dados de alta en el sistema.

The screenshot displays a web application titled "CRM - Novartis USG - Paciente". At the top, a status message indicates: "El registro aun se encuentra en proceso de captura por parte del Especialista Coordinador USG." Below this, a navigation bar includes tabs for "Estudio", "Documentos", "Perfil Articular de inicio (PsA)", "Perfil Axial (MASEI)", and "Facturación". The main form is divided into several sections for data entry:

- Número de paciente:** A field labeled "Campo no editable".
- Folio interno de estudio:** A field labeled "Campo no editable".
- Nombre(s): *** Input field containing "KARLA CELESTE".
- Apellido paterno: *** Input field containing "PANTOJA".
- Apellido materno: *** Input field containing "HERNANDEZ".
- Iniciales: *** Input field containing "PHKC".
- Sexo: *** A dropdown menu currently showing "FEMENINO".
- Fecha de nacimiento: *** A date picker showing "23/09/2001".
- Edad:** An input field showing "22 años".
- Teléfono paciente (principal): *** Input field containing "5578347634".
- Teléfono paciente (secundario):** Input field containing "5634091234".
- Correo electrónico: *** Input field containing "karla.celeste@gmail.com".
- Observaciones:** A large text area for additional notes.

At the bottom left, there is a link for "Política de privacidad". A purple circular button with a white plus sign is located at the bottom right of the form.

Figura 31. Interfaz de alta de paciente información básica. Fuente: Elaboración propia

Es después de la información básica donde entran en juego las campañas a las que puede tener acceso un paciente dependiendo del folio o nombre que le haya proporcionado su médico tratante:

OTRO Hospital-Clinica-Consultorio:

Afiliación médica: IMSS

Sector de la afiliación médica: Público

Campaña:

USG002

Duración: 2 días
Mínimo de Px: 5, Máximo de Px: 14
Pacientes registrados: 0, Cupo disponible: 14
Estatus de la campaña: PENDIENTE, Tipo de campaña: PRIVADA
Hospital: CONSULTORIO LOMAS

USG-0001

Duración: 2 días
Mínimo de Px: 10, Máximo de Px: 28
Pacientes registrados: 2, Cupo disponible: 26
Estatus de la campaña: PENDIENTE, Tipo de campaña: PÚBLICA
Hospital: HOSPITAL LA RAZA

Selecciona una opción

Figura 32. Interfaz de alta de paciente información de clínica. Fuente: Elaboración propia

De igual manera cada combo box que seleccione permite ir realizando cambios a los siguientes con validaciones personalizadas de manera que los procesos sean aún más exactos, un ejemplo de este comportamiento es que, al seleccionar una opción en *Campaña*, el combo box de *Tipo de cupón* solo mostrara los tipos a los que puede pertenecer esa campaña con base al catálogo de campañas y su tipo.

OTRO Hospital-Clinica-Consultorio:

Afiliación médica: IMSS Sector de la afiliación médica: Público

Campaña: USG002

Médico tratante:

Tipo de cupón: Campaña Pública Folio cupón: ID de activación: Campo no editable

Consultor:

Diagnóstico presuntivo: ARTRITIS PSORIASICA - M07X

Calendar: enero de 2024

	dom.	lun.	mar.	mié.	jue.	vie.	sáb.
01	31	1	2	3	4	5	6
02	7	8	9	10	11	12	13
03	14	15	16	17	18	19	20
04	21	22	23	24	25	26	27
05	28	29	30	31	1	2	3
06	4	5	6	7	8	9	10

Today Clear

Figura 33. Interfaz de alta de paciente información clínica de campañas. Fuente: Elaboración propia

Causa de solicitud:

Select All

☒ DOLOR ARTICULAR PERIFÉRICO

☐ DOLOR AXIAL

☐ DOLOR EN SITIOS DE ENTESIS

☐ PSORIASIS EN PIEL CABELLUDA, UÑAS O ZONAS INVERTIDAS

☐ PSORIASIS DE LARGA EVOLUCIÓN (>8 AÑOS)

Otra causa de solicitud:

Zona de ultrasonido:

Select All

☒ MUÑECA

☐ CODO

☒ RODILLA

Otra zona de ultrasonido:

Médico especialista:

Fecha de estudio: 17/01/2024

Horario cita USG: 12:00 p. m.

Lugar del estudio:

Figura 34. Interfaz de alta de paciente información médica. Fuente: Elaboración propia

Por último, se llena la sección que contiene datos del estudio y al dar en *Guardar* el sistema en automático (en caso de no existir flujos alternos; como algún campo vacío, de ser el caso, mostrara una alerta visual) guarda la información en la base de datos.

The screenshot shows a web browser window with a URL bar displaying 'localhost'. The application interface is a form for entering patient study information. It includes the following fields and controls:

- Utra causa de solicitud:** A text input field.
- Utra zona de ultrasonido:** A text input field.
- Médico especialista:** A dropdown menu with a search icon (magnifying glass) and a close icon (X).
- Fecha de estudio:** A date picker showing '17/01/2024'.
- Horario cita USG:** A time picker showing '12:00 p. m.'.
- Lugar del estudio:** A text input field containing 'Coacalco de Berriozabal'.
- Fecha de confirmación de estudio:** A date picker showing '14/01/2024'.
- Fecha de envío carta confirmación cita:** A date picker showing '15/01/2024'.
- Estatus del estudio:** A dropdown menu showing 'Programado'.
- Motivo por el cual no se realiza el estudio:** A dropdown menu showing 'No aplica'.

At the bottom of the form, there are two buttons: a red 'Cancelar' button and a green 'Guardar' button. A purple circular button with a plus sign is located in the bottom right corner of the form area. The browser's address bar and a 'Política de privacidad' link are visible at the very bottom.

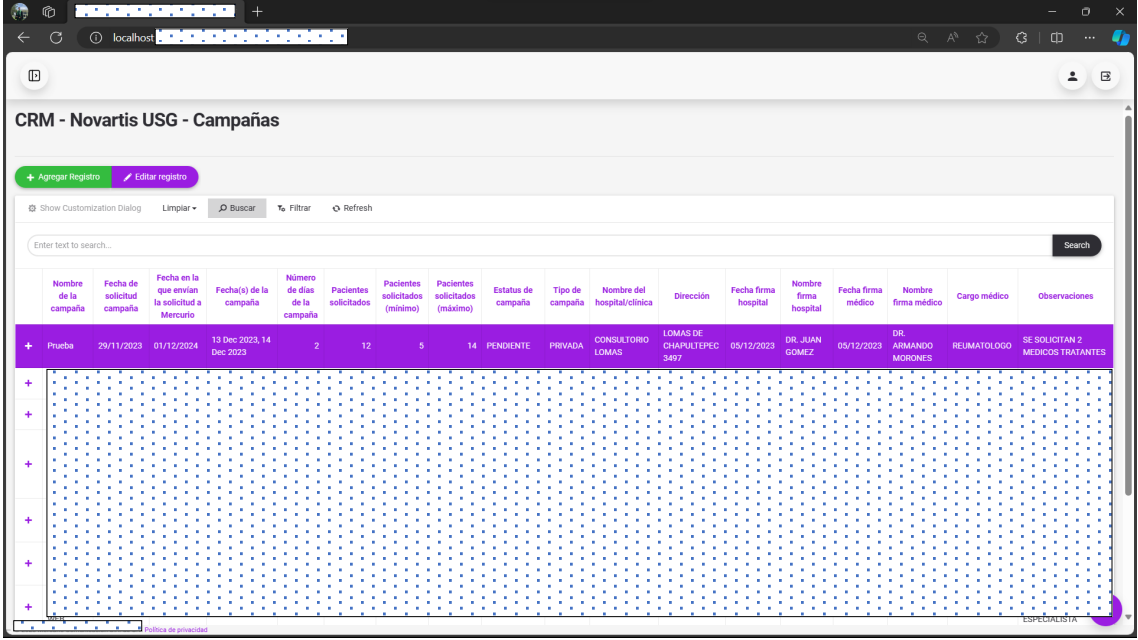
Figura 35. Interfaz de alta de paciente información de estudio. Fuente: Elaboración propia

El Kardex de médico especialista muestra la información de los médicos con los que cuenta la empresa para realizar los estudios de cada paciente dependiendo de la campaña:



Figura 36. Interfaz de médico especialista. Fuente: Elaboración propia

La pantalla de campañas que muestra el core de todos los estudios, las campañas son el punto de encuentro entre los laboratorios, los médicos y el paciente y es gracias a ellas que se pueden crear las relaciones con otros módulos del ERP, como es el caso del módulo de inventario, debido a que si una campaña requiere de determinados artículos para operar será el gestor de inventario y el encargado de CRM de congeniar esos registros para que exista armonía entre los artículos disponibles y los pacientes registrados de cada campaña:



The screenshot shows a web application interface titled "CRM - Novartis USG - Campañas". It features a search bar, a table with 17 columns, and a large grid of blue dots below the table. The table contains the following data:

Nombre de la campaña	Fecha de solicitud campaña	Fecha en la que envían la solicitud a Mercurio	Fecha(s) de la campaña	Número de días de la campaña	Pacientes solicitados	Pacientes solicitados (mínimo)	Pacientes solicitados (máximo)	Estatus de campaña	Tipo de campaña	Nombre del hospital/clínica	Dirección	Fecha firma hospital	Nombre firma hospital	Fecha firma médico	Nombre firma médico	Cargo médico	Observaciones
Prueba	29/11/2023	01/12/2024	13 Dec 2023, 14 Dec 2023	2	12	5	14	PENDIENTE	PRIVADA	CONSULTORIO LOMAS	LOMAS DE CHAPULTEPEC 2457	05/12/2023	DR. JUAN GOMEZ	05/12/2023	DR. ARMANDO MORONES	REUMATOLOGO	SE SOLICITAN 2 MEDICOS TRATANTES

Figura 37. Interfaz de campañas. Fuente: Elaboración propia

4.7 Comparación De Procesos

El tiempo de vida promedio de las operaciones generales en la empresa antes de implementar el sistema, comprendían alrededor de 20 días en el área de gestión de clientes, debido a que el alta de las campañas, la relación de médicos tratantes y especialistas con dichas campañas y la asignación de productos a diferentes pacientes requerían de más días para ser registrados en las bitácoras. Por el lado de la gestión de inventario hacer el alta de un nuevo producto, relacionarlo a una línea, conocer su proveedor y saber los almacenes donde estaría, así como realizar todas las operaciones con ese activo tomaba entre 16 y 18 días para poder ser catalogado como un registro en bitácora “estable”.

Con la implementación del sistema de información web los tiempos en términos de días se han visto reducidos en 70.59% en el módulo o área de CRM y un 55% en el módulo de inventario, produciendo una mayor tasa de eficiencia y mejorando la productividad de ambos módulos del ERP en la empresa.

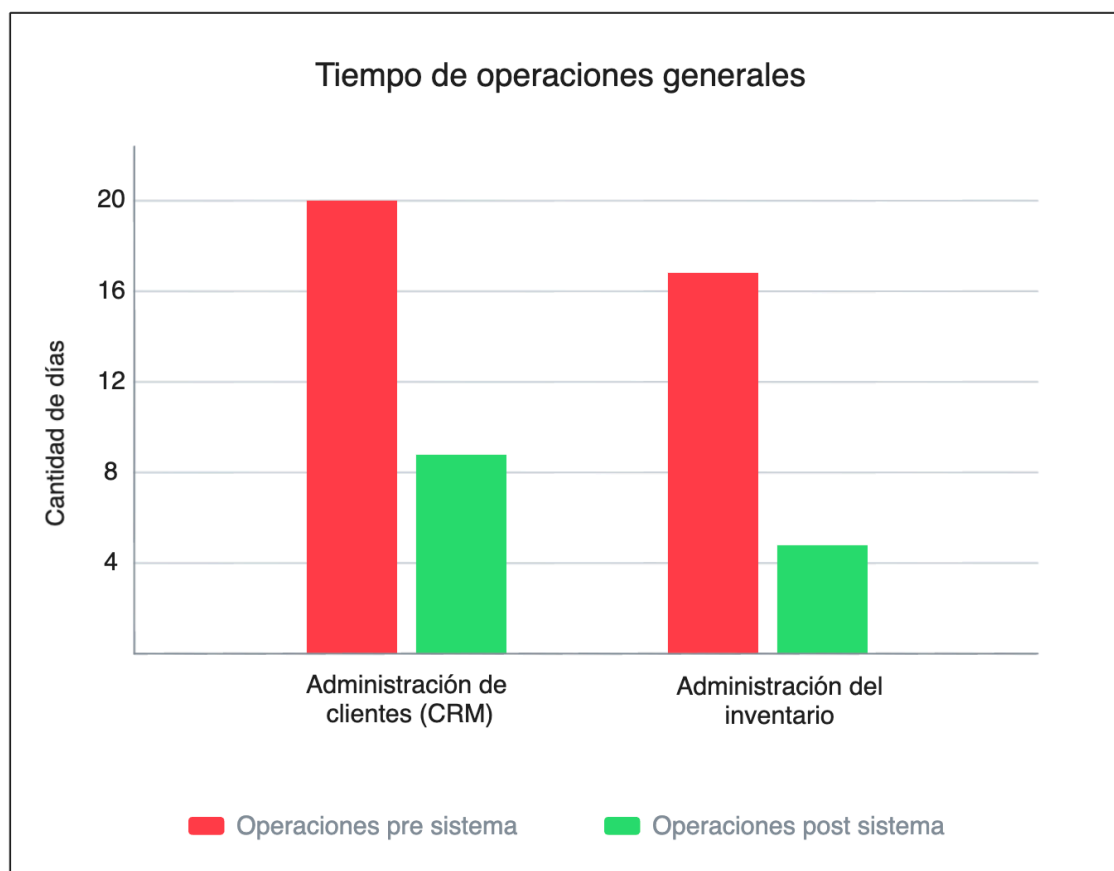


Figura 38. Comparación de días en operaciones generales pre sistema y post sistema. Fuente: Elaboración propia

Para ambos módulos la parte de altas es la carga con más gasto de recursos, tanto de esfuerzo humano como en costes de tiempo, por lo que atajar la optimización de esta fase es sumamente relevante, las altas no únicamente pertenecen a la acción de integrar un nuevo registro a una tabla de base de datos, si no que desencadena una serie de relaciones para poder tener integridad en cada una de las tablas que solicitan una información específica, un claro ejemplo de este comportamiento es:

- No se puede agregar a un nuevo paciente que no tiene el nombre de la campaña, ya que no tendría sentido agregar datos de una persona, asignarle productos de inventario si el vínculo con el laboratorio no existe.

Gracias a la implementación del sistema los flujos de altas y relaciones en términos de horas se han visto reducidos en un 79.17% para la parte de CRM en las operaciones como altas de pacientes y relaciones con campañas y un 83.33% en operaciones de alta de productos y relaciones con el paciente, almacenes, proveedores y detalles.

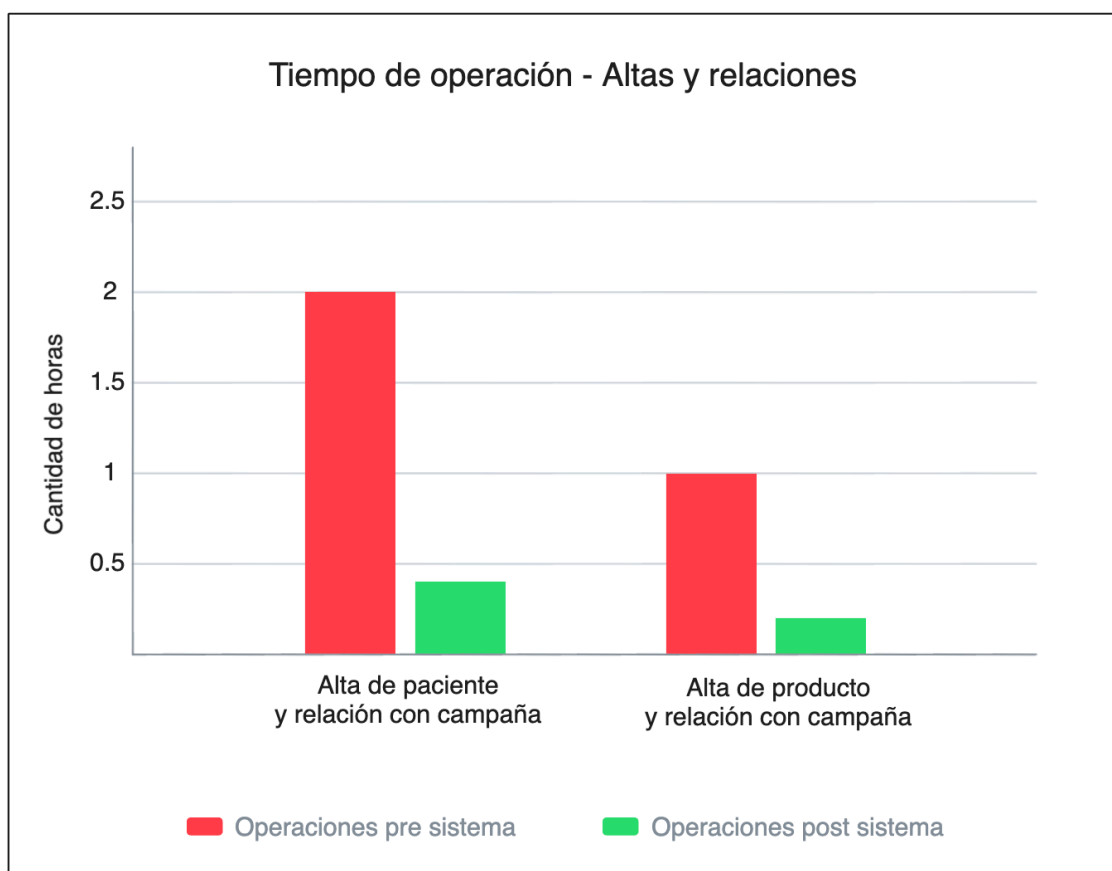


Figura 39. Comparación de horas en operaciones generales pre sistema y post sistema. Fuente: Elaboración propia

Conclusiones

Con base al trabajo presentado se puede concluir que el sistema desarrollado tendrá efectos positivos en la empresa, uno de ellos es la alta de productos y pacientes en ambos módulos del ERP, las cuales permiten que el flujo de relaciones y gestión de inventario y procesos médicos sea más ágil y eficiente, para culminar la implementación fue necesario que la empresa diera apertura a espacios internos para la carga del sistema en los equipos de cómputo de cada trabajador responsable de área, además de la capacitación del personal (tanto del sistema en general como de módulos respectivos), por mencionar algunos de los beneficios de la implementación, son la velocidad en la que se realiza una edición, una baja o un alta ya bien de datos médicos como de registros de inventario

Se implementa el sistema de información web el cual mejora todos los procesos de ambos módulos designados del ERP; CRM y gestión de inventario, los ejemplos de los procesos más importantes que se ven beneficiados considerablemente (en tiempo), son la alta de campañas que organizan los laboratorios, la relación entre los pacientes y dichas campañas, la administración de los artículos del inventario y su adecuada relación con los pacientes y las campañas. Fue gracias a este desarrollo que se pueden generar enlaces de manera más certera y consistente comparado a cuando se realizaban de manera manual.

Para el área de inventario y almacén, se libera el módulo de gestión de inventario, de manera que se puede llevar un control exhaustivo de los productos del almacén y todas las vertientes que lo complementan, como las líneas los proveedores y los colaboradores. Gracias a este módulo, la generación de reportes, creación de nuevos registros, relaciones y administración en general de los artículos, son tareas que se verán beneficiadas por el ahorro de tiempo y de recursos de la empresa Mercurio Comunicación al realizar convenios con otros laboratorios farmacéuticos.

Para el sector de clientes de la empresa se deja en funcionamiento el módulo CRM, permitiendo a los encargados de esta área acortar el tiempo de sus tareas y realizar un registro más robusto sobre la información de los pacientes, los médicos y los programas actuales de cada laboratorio. Ayuda a corregir errores que suelen presentarse al utilizar un método manual y se automatizan los procesos del área gracias al uso de tecnologías web que cada

vez representan un mayor impacto en cualquier área empresarial, permitiendo gestionar de manera muy versátil y segura.

Para comunicar ambas áreas y poder crear relaciones armónicas se implementa una arquitectura de jerarquías, donde cada nodo padre son los módulos del ERP y cuentan con fragmentos más pequeños dentro, que conviven y se comunican entre sí, pero sin olvidar la relación entre los nodos padre y poder delegar tareas más específicas a cada nodo hijo y así evitar conflictos entre operaciones a los módulos.

Al final, el desarrollo de este proyecto permite desempeñar las competencias adquiridas durante el periodo de estudio en el Tecnológico. El conocer cómo funciona y estar inmerso en el entorno laboral desde su interior, es de suma importancia para poder conocer la lógica de negocio y saber dónde atajar las necesidades de una empresa con un sistema computacional que proporcione seguridad, escalabilidad, eficiencia y confianza.

Competencias Desarrolladas

Tabla 20. Competencias desarrolladas. Fuente: Elaboración propia

Asignatura	Competencia
Fundamentos de Programación	Aplica algoritmos y lenguajes de programación para diseñar e implementar soluciones a problemáticas del entorno
Fundamentos de Investigación	Habilidades para buscar, procesar y analizar información procedente de diversas fuentes
Taller de Ética	Fundamente la práctica ética del ejercicio profesional en la toma de decisiones para la solución de problemas en las instituciones y organizaciones
Programación Orientada a Objetos	Aplica la programación orientada a objetos para resolver problemas reales y de ingeniería
Estructura de Datos	Conoce, comprende y aplica eficientemente estructuras de datos, métodos de ordenamiento y búsqueda para la optimización del rendimiento de soluciones a problemas del mundo real
Cultura Empresarial	Analiza los conceptos fundamentales del contexto de la empresa para identificar áreas de oportunidad proponiendo planes de negocio
Fundamentos de Base de Datos	Conoce y aplica el modelo relacional para la generación de esquemas de base de datos con el fin de organizar la información y atender necesidades del entorno
Programación Web	Desarrolla aplicaciones web dinámicas del lado cliente y del servidor, considerando la conectividad a orígenes de datos, la interconectividad entre aplicaciones y cómputo en la nube

Tabla 20 (continuación). Competencias desarrolladas. Fuente: Elaboración propia

Asignatura	Competencia
Sistemas Operativos	Aplica los paradigmas de diseño de los sistemas operativos actuales y emergentes, para el manejo de los recursos del sistema
Taller de Base de Datos	Implementa bases de datos para apoyar la toma de decisiones considerando las reglas de negocio
Arquitectura de Computadoras	Conoce e identifica los componentes y el funcionamiento en diferentes modelos de arquitectura
Taller de Investigación I	Elabora un protocolo de investigación en el que presenta soluciones científico - tecnológicas a problemáticas relacionadas con su campo profesional en diversos contextos
Fundamentos de Ingeniería de Software	Realiza el análisis de un proyecto de software, a partir de la identificación del modelo de negocios de la organización que permita alcanzar estándares y métricas de calidad
Administración de Base de Datos	Comprende las actividades de la administración de bases de datos, identifica las funciones del DBA y las características de los diferentes SGBD con el fin de dimensionar su importancia en las organizaciones
Programación Lógica y Funcional	Realiza una aplicación dando solución a un problema del entorno usando el paradigma de la programación funcional
Ingeniería de Software	Desarrolla soluciones de software, considerando la metodología y herramientas para la elaboración de un proyecto aplicativo en diferentes escenarios

Tabla 20 (continuación). Competencias desarrolladas. Fuente: Elaboración propia

Asignatura	Competencia
Gestión de Proyectos de Software	Aplica metodologías e instrumentos, para garantizar la gestión adecuada de un proyecto de software
Ingeniería de Requerimientos	Analiza el proceso de requerimientos identificando sus modelos, actores, administración, soporte de procesos, calidad y mejora del proceso
Arquitectura y Diseño de Software	Identificar el impacto de las decisiones arquitectónicas en relación con los atributos de calidad con el uso de lenguajes y la utilización de los modelos de proceso y diseño
Verificación y Validación de Software	Manejar técnicas para verificaciones y validaciones al software, considerando los aspectos de revisión y pruebas como parte del ciclo de vida para detectar fallas en el mismo

Fuentes De Información

- Arciniegas, C. A. (2019). *Desarrollo de los módulos del sistema de información FACTURASSES para la empresa SSES LTDA*. Obtenido de Red UAO: <https://red.uao.edu.co/bitstream/handle/10614/11249/T08641.pdf?sequence=5>
- Blancarte, O. (2016). *Introducción a los Patrones de Diseño*. Ciudad de México, México.
- Bos, B. (28 de Octubre de 2023). *Cascading Style Sheets home page*. Obtenido de w3c: <https://www.w3.org/Style/CSS/Overview.en.html>
- Casañ, A. M. (2020). *Desarrollo de software ERP para empresa de montajes*. Obtenido de riunetupv: <https://riunet.upv.es/bitstream/handle/10251/174813/Albert%20-%20Desarrollo%20de%20software%20ERP%20para%20empresa%20de%20montajes.pdf?sequence=1>
- Chang, W. F. (Septiembre de 2017). *Desarrollo de Aplicación Web para el control de inventario y ventas de la empresa "Casa Automotriz Esteven"*. Obtenido de DSPACE: <https://dSPACE.ups.edu.ec/bitstream/123456789/14893/1/UPS-GT002018.pdf>
- Claro, H. (2010). En *Estándares de Diseño Web. Ciencias de la Información* (págs. 69-71).
- Cruz, G. C. (2013). *Desarrollo de un sistema CRM para la empresa inmobiliaria ACMINMOBILIARIA apoyado en el uso de herramientas web*. Obtenido de repositoriouisrael: <http://repositorio.uisrael.edu.ec/bitstream/47000/480/1/UISRAEL-EC-SIS-378.242-248.pdf>
- Díaz, A. (2005). *Implantación de un sistema ERP en una organización*.
- Escudero, C. J. (Noviembre de 2017). *Diseño de sistema de control y gestión de proveedores y agentes para empresa inmobiliaria*. Obtenido de uaeh: https://www.uaeh.edu.mx/docencia/Tesis/huejutla/licenciatura/2018/Sistema_de_Gestion_de_Proveedores.pdf
- Fairhurst, P. (2001). e-CRM. En *Journal of Database Marketing & Customer Strategy Management* 8 (págs. 137–142).
- Garma, S. L. (2018). *Desarrollo de un Sistema Web para el Control de la Producción de Calzado en la Empresa Laguna*. Obtenido de Repositorio UCV: https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/33122/Valle_GSL.pdf?sequence=1&isAllowed=y
- Gomez, X. P. (2020). *Implementación de un sistema ERP para mejorar el control de inventario de la Librería Bazar "Diamante Azul" en el distrito de El Tambo en el año 2020*. Huancayo.

- Granja, S. P. (2017). *Desarrollo de una aplicación móvil integrada al ERP ekon, para el almacenamiento, envío y gestión de los reportes diarios de obra de los proyectos de la empresa SMARTPRO S.A.* Obtenido de dspace: <https://dspace.udla.edu.ec/bitstream/33000/8087/1/UDLA-EC-TIS-2017-11.pdf>
- Guaman, M. P. (2021). *Desarrollo de un sistema web para la gestión de inventarios en el almacén "MADERKA" utilizando el framework Laravel.* Obtenido de dspace: <http://dspace.esPOCH.edu.ec/bitstream/123456789/19270/1/18T00859.pdf>
- Hernández, G. R. (2017). *Desarrollo de un módulo de inventario de un sistema ERP para la empresa "Electro sur CIA. lta" telefonía & electricidad.* Obtenido de Repositorio UG: <https://repositorio.ug.edu.ec/server/api/core/bitstreams/d5b6a640-cbbd-4bf4-bcb6-a153ff09c4a7/content>
- Idrobo, K. L. (2016). *Análisis y desarrollo de un sistema web y móvil para el monitoreo de los carros tanques de una empresa de Guayaquil dedicada a la venta de alcohol.* Obtenido de Repositorio ucsg: <http://repositorio.ucsg.edu.ec/bitstream/3317/5163/3/T-UCSG-PRE-ING-CIS-135.pdf>
- Legua, C. D. (2015). *Implementación de un sistema web para el control de las ventas, almacén y garantías de una empresa.* Obtenido de Repositorio URP: https://repositorio.urp.edu.pe/bitstream/handle/20.500.14138/2033/rivera_cd-torrejón_bl.pdf?sequence=1&isAllowed=y
- López, C. A. (2020). *Desarrollo de un sistema de información web para la gestión de los procesos administrativos de una tienda de calzado.* Obtenido de Udistrital: <https://repository.udistrital.edu.co/bitstream/handle/11349/28423/CarlosAndresQui-mbayLopez2021.pdf?sequence=2>
- Lozada, A. (2 de Octubre de 2021). *¿Qué es SQL?* Obtenido de EDteam: <https://ed.team/blog/que-es-sql>
- Maceda, H. C. (2016). *Arquitectura de Software.*
- Manayalle, I. M. (2021). *Implementación de un sistema de facturación electrónica para costa rica.* Obtenido de Repositorio UNPRG: https://repositorio.unprg.edu.pe/bitstream/handle/20.500.12893/10618/Ivan_Morales_Manayalle.pdf?sequence=1&isAllowed=y
- MDN contributors. (5 de Diciembre de 2022). *MVC.* Obtenido de MDN web docs: <https://developer.mozilla.org/es/docs/Glossary/MVC>
- Mejía, K. I. (Mayo de 2012). *Construcción de sitio web de administración de información de laboratorios caso de aplicación: laboratorio de sistemas digitales de la UACJ.* Obtenido de Erecursos UACJ: http://erecursos.uacj.mx/bitstream/handle/20.500.11961/3173/Entrega_Final_revisada02.pdf?sequence=1&isAllowed=y

- Mercurio (2023). *INFORMACIÓN MÉDICA ESTRATÉGICA*. Obtenido de Mercurio Comunicación: <https://mercuriocomunicacion.net/>
- Microsoft. (15 de 02 de 2023). *Documentación de .NET*. Obtenido de Microsoft: <https://learn.microsoft.com/es-es/dotnet/>
- Patarroyo, C. F. (2020). *Software para la gestión y análisis de la información (SGAI - PYME)*. Obtenido de Repository libertadores: https://repository.libertadores.edu.co/bitstream/handle/11371/4201/Neira_Fandi%C3%B1o_2021.pdf?sequence=1
- Pecharromán, A. U. (2015). *Aplicación de la metodología de desarrollo ágil Scrum para el desarrollo de un sistema de gestión de empresas*. Obtenido de uc3m: https://e-archivo.uc3m.es/bitstream/handle/10016/23750/TFG_Aitor_Urteaga_Pecharroman.pdf?sequence=1&isAllowed=y
- Perez, M. (2011). *Microsoft SQL Server motor de base de datos y administración*. RC Libros.
- Playwright. (2023). *playwright.dev*. Obtenido de Playwright for .NET: <https://playwright.dev/dotnet/docs/intro>
- Príncipe, N. T. (2021). *Implementación de un sistema web para la gestión de ventas e inventario*. Obtenido de repositoriosil: <https://repositorio.usil.edu.pe/server/api/core/bitstreams/85353a5a-5a65-4b03-a6ba-ab7ca514fab9/content>
- Sandoval, K. W. (30 de Abril de 2021). *Implementación del módulo de Compras del Sistema KOHINOR DBS utilizando Visual Studio y ASP.NET*. Obtenido de Repositorio USFQ: <https://repositorio.usfq.edu.ec/bitstream/23000/10554/1/136086.pdf>
- Santander. (21 de 12 de 2020). *Metodologías de desarrollo de software: ¿qué son?* Obtenido de Santander: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>
- Schmedtmann, J. (s.f.). *The Complete JavaScript Course*. Algarve, Portugal.
- TC39. (s.f.). *ECMAScript® 2023 Language Specification*. Obtenido de <https://tc39.es/ecma262/2023/#sec-intro>
- Zhou, J. X. (7 de Abril de 2020). *Desarrollo de aplicación de gestión empresarial (ERP) sobre entorno MySQL/PHP*. Obtenido de Riunet: <https://m.riunet.upv.es/bitstream/handle/10251/143045/Xia%20-%20DESARROLLO%20DE%20UNA%20APLICACI%C3%93N%20DE%20GESTI%C3%93N%20EMPRESARIAL%20EN%20UN%20ENTORNO%20PHP/MYSQL.pdf?sequence=1&isAllowed=y>

Anexos

Contrato De Confidencialidad

**BAACC SYSTEMS S.A. DE C.V.**

CONTRATO DE CESION DE DERECHOS Y CONFIDENCIALIDAD QUE CELEBRAN EL C. LUIS ANGEL VALENCIA ARCEGA, A QUIEN EN LO SUCESIVO SE LE DESIGNARA EL "CEDENTE", Y POR LA OTRA PARTE; BAACC SYSTEMS S.A. DE C.V. A LA QUE EN LO SUCESIVO SE LE DENOMINARA EL "CESIONARIO", REPRESENTADA EN ESTE ACTO POR EL C. ALEJANDRO MONTEERRUBIO HERNÁNDEZ, AL TENOR DE LAS DECLARACIONES Y CLAUSULAS SIGUIENTES:

DECLARACIONES

I. DECLARA EL "CEDENTE":

1. Es una persona física de nacionalidad MEXICANA en pleno uso de su capacidad legal.
2. Acredita su personalidad mediante INE, que se adjunta como Anexo 1 al presente contrato.
3. Acredita no ser propietario de ningún producto de software (programas, sistemas, etc.) desarrollados por el CEDENTE durante las fechas que van del **24 de Julio de 2023 al 31 de Diciembre de 2023**, y ceder los Derechos de Autor que posea sobre cualquier software desarrollado en esas fechas, por estar comprometida su total disponibilidad en exclusividad con el CESIONARIO, debido al tipo de actividades necesarias para el desarrollo de software y actos relacionados con ello, para con el CESIONARIO y para con sus diferentes clientes en esas fechas, durante las cuales hizo uso de un cubículo e instalaciones eléctricas, de red alámbrica e inalámbrica y telefónicas, así como implementos de oficina, software, etc. con conexiones a Internet para el buen cumplimiento de su trabajo, pagados por el CESIONARIO, en las oficinas del CESIONARIO, situadas en la dirección: **Ébanos 107 Col. Villa de las Flores, Coacalco de Berriozábal, Estado de México, C.P. 55710.**
4. Acuerda entregar cualquier copia u original de documentos en cualquier formato sea este electrónico, escrito, grabado, etc. que tengan alguna relación con el CESIONARIO o con alguno de los clientes del CESIONARIO.
5. Se compromete a no divulgar de ninguna forma hablada, escrita, electrónica, etc. los datos, procedimientos, acuerdos, contactos del CESIONARIO o de alguno de sus CLIENTES.
6. Es su deseo celebrar el presente contrato de Cesión y Confidencialidad libre de vicio alguno.
7. La Cesión de Derechos y Confidencialidad se realiza con motivo de que libremente se incorporó a los proyectos del CESIONARIO y que estuvo presente en reuniones con clientes del CESIONARIO y en reuniones con el CESIONARIO donde se detallaron procesos, procedimientos, datos, nombres de usuarios, contraseñas, etc. de la operación de los clientes del CESIONARIO y del CESIONARIO, y que leyó y tuvo en su poder los programas de software, scripts de bases de datos en lenguaje SQL, diseño de bases de datos, bases de datos con estructuras y datos, propiedad y con registro de derechos de autor del CESIONARIO, del **24 de Julio de 2023 al 31 de Diciembre de 2023.**
8. Que se compromete a no tocar, ver, copiar, divulgar, prestar, comentar o usar ninguno de los documentos, procedimientos, datos, programas, etc. no importando su formato, ya sea escrito, electrónico, grabado, etc. que se encuentren en las instalaciones, computadoras, servidores, etc. del CESIONARIO o de alguno de los clientes del CESIONARIO, o que sean producto de las reuniones con los clientes del CESIONARIO o de las reuniones internas con el CESIONARIO, ni firmarse o introducirse en ninguna de sus instalaciones, computadoras, servidores, etc. del CESIONARIO o de alguno de sus CLIENTES a partir e incluido el día **24 de Julio de 2023.**
9. Que, para los efectos legales derivados de este contrato, señala como su domicilio legal: **FCO VILLA LT 5 EDIF B10 D103 GUADALUPE VICTORIA ECATEPEC DE MORELOS ESTADO DE MEXICO C.P. 55010**

Camelia 64, Col. Bosque de los Remedios, Naucalpan de Juárez, Estado de México, C.P. 53458
Oficina **(55) 8624 7021** y **(55) 7577 8805** * WhatsApp Business **(55) 6263 7194**
contacto@baacc-systems.mx * www.baacc.mx

Página 1 | 3



BAACC SYSTEMS S.A. DE C.V.

II. DECLARA EL "CESIONARIO":

1. Es una SOCIEDAD ANONIMA DE CAPITAL VARIABLE constituida de conformidad con las leyes de los Estados Unidos Mexicanos.
2. Tiene como objeto:
 - i. Análisis de requerimiento de negocios para diseñar, desarrollar e implementar soluciones de infraestructura de redes, comunicaciones, sistemas operativos, aplicaciones y bases de datos.
 - ii. Diseñar, desarrollar e implementar soluciones de software centralizadas, distribuidas, cliente servidor y en la web.
 - iii. Diseñar, desarrollar e implementar modelos de gestión, así como lineamientos, normas, procesos, procedimientos y estándares en tecnologías de la información y de las comunicaciones.
 - iv. Servicios de posicionamiento en buscadores de páginas web. Compra y venta al mayor y menor de dominios, espacio de almacenamiento en la web, banda ancha y certificados de autenticidad.
 - v. Instalaciones de redes integradas de cómputo con cableado estructurado, cableado de fibra óptica, cableado coaxial.
 - vi. Venta y arrendamiento de equipos de cómputo, dispositivos periféricos, gabinetes para concentradores y ruteadores. Mantenimiento preventivo y correctivo a redes, equipos en red, equipos de cómputo, dispositivos periféricos.
 - vii. Prestación, contratación, subcontratación de todo tipo de servicios informáticos, de telecomunicaciones y de consultoría en tecnologías de la información y de las comunicaciones.
 - viii. Elaboración, edición, producción, coproducción, rodaje, publicación y comercialización de toda clase de obras audiovisuales, incluyendo spots publicitarios; doblaje, sonorización, montaje, edición y postproducción de obras audiovisuales.
 - ix. Servicios de diseño gráfico, realización de proyectos editoriales, de artes gráficas o de cualquier otra naturaleza. Publicación electrónica, imagen digital, impresión, servicios publicitarios, agencia de publicidad. Gestión, realización, comercialización y distribución de servicios de publicidad a través de uno o varios medios de difusión.
 - x. Ejecutar, celebrar y cumplir con toda clase de actos mercantiles, civiles, laborales, administrativos, y de cualquier otra naturaleza relacionado con las actividades de la sociedad en forma directa o indirecta por lo que así mismo, podrán adquirir, poseer, arrendar usar, enajenar, y en general ser parte de cualquier contrato relativo a la posesión, arrendamiento, uso y enajenación de los bienes muebles.
 - xi. Solicitar y obtener préstamos, suscribir y endosar, aceptar, toda clase de títulos de crédito y garantizar obligaciones de la sociedad, con la intervención de las instituciones específicas por la ley cuando ello sea necesario, directamente relacionados con los objetos de la sociedad.
 - xii. Comercializar, crear, ampliar, fusionar, enajenar, operar y administrar empresas dedicadas a cualquier actividad anexa o conexas, establecer, arrendar, subarrendar y negociar en todas sus formas con fábricas, plantas industriales, almacenes, tiendas, oficinas y en general con todo tipo de personas físicas.
 - xiii. Manufacturar, importar, exportar, reconstruir, distribuir, ensamblar y en general negociar en cualquier forma con los accesorios, artículos, refacciones, suministros, materia prima, productos terminados o semiterminados y con aquellos bienes relacionados directa o indirectamente con los de esta sociedad.
 - xiv. Establecer oficinas, agencias, representaciones y sucursales en la República Mexicana y en el extranjero para el desarrollo de las actividades de la empresa.
 - xv. La adquisición por cualquier medio legal de acciones, partes sociales y en general participaciones y/o intereses en otras sociedades de objeto análogo, similar o vinculado directa o indirectamente con los de esta sociedad.
 - xvi. Representar o ser agente en los Estados Unidos Mexicanos de empresas industriales y comerciales, nacionales o extranjeras.

Camelia 64, Col. Bosque de los Remedios, Naucalpan de Juárez, Estado de México, C.P. 53458

Oficina (55) 8624 7021 y (55) 7577 8805 * WhatsApp Business (55) 6263 7194

contacto@baacc-systems.mx * www.baacc.mx



BAACC SYSTEMS S.A. DE C.V.

- xvii. Comprar, vender, arrendar, hipotecar y en general comercializar en cualquier forma legal con bienes de muebles e inmuebles.
 - xviii. Establecer sucursales y subsidiarias en la República Mexicana y/o en el extranjero.
 - xix. Solicitar, obtener, comprar, arrendar, ceder, o de cualquier otra forma, disponer y adquirir patentes de invención, registro de modelos de utilidad, diseños industriales, marcas, avisos y nombres comerciales, licencias de explotación y de uso de estos: declaraciones de protección de denominaciones de origen y autorizaciones para uso y derechos de autor ya sean en los Estados Unidos Mexicanos o en el extranjero.
 - xx. Dar y tomar dinero en préstamo con y sin garantía; emitir bonos y cedulas; suscribir, aceptar, endosar y avalar títulos de crédito; garantizar obligaciones propias y de terceros, con o sin garantías específicas; emitir obligaciones y toda clase de valores con intervención en su caso, de las instituciones requeridas por la Ley.
 - xxi. Otorgar garantías propias a favor de terceros, fianzas, confianzas, así como obligarse solidariamente ante instituciones de crédito y otras entidades financieras y particulares a favor de las sociedades y otras personas morales y físicas aun cuando se trate de personas extrañas a la sociedad.
 - xxii. En general realizar toda clase de actos comerciales, industriales y celebrar toda clase de contratos, convenios y operaciones, ya sean civiles o mercantiles permitidos por la Ley y llevar a cabo todos los actos legales necesarios para el desarrollo de su objeto social.
3. El C. ALEJANDRO MONTEERRUBIO HERNÁNDEZ tiene poderes y facultades suficientes para celebrar el presente contrato, mismas que hasta la fecha no le han sido revocadas, modificadas y/o limitadas.
4. Es su deseo celebrar el presente contrato de Cesión de Derechos y Confidencialidad, libre de vicio alguno.
5. Que, para los efectos legales derivados de este contrato, señala como su domicilio legal: **CALLE CAMELIA 64 COL. BOSQUE DE LOS REMEDIOS, NAUCALPAN DE JUÁREZ, ESTADO DE MÉXICO, C.P. 53458.**

CLAUSULAS

PRIMERA. En caso de cualquier controversia que pudiera suscitarse respecto al cumplimiento o interpretación del presente instrumento, las partes acuerdan someterse a las leyes aplicables, competencia y jurisdicción de los tribunales del Estado de México, por lo que a partir de este momento renuncia a cualquier otra jurisdicción que por virtud del domicilio presente o futuro o por cualquier otra circunstancia pudiese corresponderles.

Leído el presente contrato y enteradas las partes de su valor y contenido legal, lo firman al calce de la última hoja y al margen de las anteriores, en Coacalco de Berriozábal, Estado de México, a los VENTICUATRO días del mes de JULIO del 2023.

POR EL "CEDENTE"

LUIS ANGEL VALENCIA ARCEGA

POR EL "CESIONARIO"

ALEJANDRO MONTEERRUBIO HERNÁNDEZ

Camelia 64, Col. Bosque de los Remedios, Naucalpan de Juárez, Estado de México, C.P. 53458
Oficina (55) 8624 7021 y (55) 7577 8805 * WhatsApp Business (55) 6263 7194
contacto@baacc-systems.mx * www.baacc.mx