

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE COACALCO

Título de Trabajo

Sistema de información web con PHP
y MySQL para la gestión de datos
logísticos en la empresa Quality Std
S.A de C.V.

Reporte Final de Residencias Profesionales

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Especialidad
Desarrollo de Software

Presenta(n):

- Téllez López Francisco.

Calificación: _____

Coacalco de Berriozábal, diciembre 2023

AGRADECIMIENTOS

Agradezco a mis padres que desde pequeño me inculcaron la disciplina de estudiar y luchar por los sueños, a mi hermano quien siempre creyó en mí, a mis profesores por su invaluable contribución a mi formación académica, a mis amigos por siempre apoyarme y sobre todo a dios quien ha sido mi guía en cada logro, desafío y por colocar a las personas correctas en mi vida en el momento adecuado.

RESUMEN

El sistema de información web con PHP y MySQL es un proyecto que se realiza con el fin de aportar a la empresa Quality Std S.A de C.V. una herramienta innovadora para simplificar la gestión y almacenamiento de datos esenciales en sus operaciones diarias. La implementación de este sistema busca optimizar y agilizar los procesos internos de la empresa, mejorando la eficiencia y proporcionando una plataforma más efectiva para el manejo de información.

Este sistema de información web permite a los usuarios interactuar de manera intuitiva con la base de datos a través de una interfaz web. PHP es un lenguaje de programación que se utiliza para el desarrollo de aplicaciones web, mientras que MySQL, un sistema de gestión de bases de datos relacionales que se utiliza para garantizar el almacenamiento, organización y acceso eficiente a los datos. La inclusión de Bootstrap en el proyecto proporciona una interfaz atractiva y responsiva, mejorando la experiencia del usuario.

La solución desarrollada permite a los usuarios realizar diversas acciones, como buscar, agregar, editar, y eliminar datos mediante formularios y vistas de datos. Además, el sistema es altamente personalizable para adaptarse a las necesidades específicas de Quality Std S.A de C.V., y ofrece la posibilidad de integrarse con otras aplicaciones web y sistemas existentes.

ÍNDICE

AGRADECIMIENTOS.....	ii
RESUMEN.....	iii
ÍNDICE DE FIGURAS.....	vii
ÍNDICE DE TABLAS.....	x
CAPITULO I GENERALIDADES DEL PROYECTO	11
1.1 Introducción	12
1.2 Problema a resolver.....	13
1.3 Objetivos	14
1.3.1 Objetivo general	14
1.3.2 Objetivos específicos.....	14
1.4 Justificación	15
CAPÍTULO II MARCO TEÓRICO.....	17
2.1 Antecedentes	18
2.2 Marco Conceptual.....	18
2.2.1 Sistema de información web.....	18
2.2.2 Servidor web.....	18
2.2.3 Metodología de desarrollo de software.	19
2.2.4 Metodología incremental.....	19
2.2.5 Lenguaje de programación.	19
2.2.6 Backend.	19
2.2.7 PHP.....	19
2.2.8 JavaScript.....	20
2.2.9 Frontend.....	20
2.2.10 HTML.....	20
2.2.11 CSS.....	21
2.2.12 Framework.....	21
2.2.13 Bootstrap.....	21
2.2.14 Base de datos.	22
2.2.15 Base de datos relacional	22

2.2.16 MySQL.....	22
CAPÍTULO III DESARROLLO.....	24
3.1 Diseño del sistema de información web	25
3.2 Requerimientos	26
3.2.1 Requerimientos funcionales	27
3.2.2 Requerimientos no funcionales:	28
3.3 Diseño del software.....	30
3.3.1 Metodología para el desarrollo de software (Incremental).....	30
3.3.2 Diseño de la arquitectura del sistema	31
3.3.3 Backend (Lógica de negocio)	31
3.3.4 Frontend (Interfaz de usuario)	32
3.3.5 Integración de tecnologías	32
3.3.6 Diseño de la base de datos.....	32
3.3.7 Diagramas de caso de uso general.....	34
3.4 Sistema de información.....	35
3.4.1 Módulo login	40
3.4.2 Módulo empleados	41
3.4.3 Módulo clientes:	43
3.4.4 Módulo proveedores:.....	44
3.4.5 Módulo productos:.....	46
3.4.6 Módulo compras:	47
3.4.7 Módulo cuentas por pagar:	49
3.4.8 Módulo ventas:	51
3.4.9 Módulo cuentas por cobrar:	53
3.4.10 Módulo envíos:	54
3.5 Estructuras de control	55
3.5.1 Sentencia WHILE.....	55
3.5.2 Sentencia IF ELSE.....	56
3.5.3 Sentencia FOR.....	56
3.5.3 Sentencia SWITCH	57
3.6 Trigger (disparador)	57
3.7 Pruebas.....	60

3.8 Documentación	61
CAPÍTULO IV RESULTADOS	62
4.1 Login	63
4.2 Interfaz inicio	64
4.3 Empleados	65
4.4 Clientes	68
4.5 Proveedores	71
4.6 Productos	74
4.7 Compras	78
4.8 Cuentas por pagar	82
4.9 Ventas	83
4.10 Cuentas por cobrar	86
4.11 Envíos	87
4.12 Pedidos	89
4.13 Prueba en cuanto a funcionalidad en uso	91
4.14 Implementación del proyecto en ambiente productivo:	91
CONCLUSIONES	93
COMPETENCIAS DESARROLLADAS	94
BIBLIOGRAFÍA	96

ÍNDICE DE FIGURAS

Figura 1. Cronograma de actividades del proyecto	23
Figura 2. Metodología incremental.....	31
Figura 3. Diseño modelo entidad-relación.....	33
Figura 4. Modelo entidad-relación phpmyadmin.....	34
Figura 5. Diagrama de caso de uso general.	35
Figura 6. Menú lateral.....	36
Figura 7. Referencia al header	36
Figura 8. CSS diseño a elementos.....	37
Figura 9. Conexión a la base de datos.....	38
Figura 10. Cerrar sesión de la base de datos.....	38
Figura 11. Paginación	39
Figura 12. Lógica para la paginación.....	39
Figura 13. Consulta para el buscador.....	40
Figura 14. Buscador	40
Figura 15. Caso de uso: login	40
Figura 16. Validaciones login	41
Figura 17. Login usuario o contraseña incorrecta.....	41
Figura 18. Caso de uso: empleados.....	42
Figura 19. Tabla empleados SQL	42
Figura 20. Módulo empleados	43
Figura 21. Caso de uso: clientes	43
Figura 22. Tabla clientes SQL	44
Figura 23. Módulo clientes	44
Figura 24. Caso de uso: proveedores	45
Figura 25. Tabla proveedores SQL	46
Figura 26. Módulo proveedores.....	46
Figura 27. Caso de uso: productos.....	47
Figura 28. Tabla productos SQL.....	47
Figura 29. Caso de uso: compras	48
Figura 30. Tabla compras SQL	48
Figura 31. Tabla detalles compra SQL	49
Figura 32. Módulo compras.....	49
Figura 33. Caso de uso: cuentas por pagar	50
Figura 34. Tabla cuentas por pagar SQL	51
Figura 35. Caso de uso: ventas	51
Figura 36. Tabla ventas SQL	52
Figura 37. Tabla detalles venta SQL.....	52
Figura 38. Módulo ventas	53
Figura 39. Caso de uso: cuentas por cobrar	53
Figura 40. Tabla cuentas por cobrar SQL	54

Figura 41. Caso de uso: envíos	54
Figura 42. Tabla envíos SQL	55
Figura 43. Módulo envíos	55
Figura 44. Sentencia WHILE	56
Figura 45. Sentencia IF ELSE	56
Figura 46. Sentencia FOR	57
Figura 47. Sentencia SWITCH	57
Figura 48. Trigger: actualizar total venta	58
Figura 49. Trigger: agregar cuenta por cobrar	59
Figura 50. Trigger: eliminar cuenta por cobrar	59
Figura 51. Trigger: agregar cuenta por pagar	60
Figura 52. Trigger: eliminar cuenta por pagar	60
Figura 53. Interfaz login	63
Figura 54. Interfaz login: usuario o contraseña incorrectos	64
Figura 55. Interfaz inicio	65
Figura 56. Interfaz empleados	65
Figura 57. Modal agregar empleado	66
Figura 58. Modal modificar empleado	66
Figura 59. Modal eliminar empleado	67
Figura 60. Buscador empleados	67
Figura 61. Base de datos empleados	68
Figura 62. Interfaz clientes	68
Figura 63. Modal agregar cliente	69
Figura 64. Modal modificar cliente	69
Figura 65. Modal eliminar cliente	70
Figura 66. Buscador clientes	70
Figura 67. Base de datos clientes	71
Figura 68. Interfaz proveedores	71
Figura 69. Modal agregar proveedor	72
Figura 70. Modal modificar proveedor	72
Figura 71. Modal eliminar proveedor	73
Figura 72. Buscador proveedores	73
Figura 73. Base de datos proveedores	74
Figura 74. Interfaz productos	74
Figura 75. Modal agregar producto	75
Figura 76. Modal modificar producto	75
Figura 77. Modal eliminar producto	76
Figura 78. Descargar excel con los productos de la base de datos	76
Figura 79. Excel con los productos de la base de datos	77
Figura 80. Cargar excel con productos	77
Figura 81. Buscador productos	77
Figura 82. Base de datos productos	78
Figura 83. Interfaz compras	78

Figura 84. Modal agregar nueva compra	79
Figura 85. Modal modificar estado de pago compra	79
Figura 86. Descargar factura compra PDF	80
Figura 87. Factura PDF compra.....	80
Figura 88. Filtrar compras.....	81
Figura 89. Base de datos compras.....	81
Figura 90. Interfaz cuentas por pagar	82
Figura 91. Base de datos cuentas por pagar	82
Figura 92. Interfaz ventas	83
Figura 93. Modal agregar nueva venta	84
Figura 94. Modal modificar estado de pago venta.....	84
Figura 95. Descargar factura venta PDF	85
Figura 96. Factura PDF venta	85
Figura 97. Base de datos ventas.....	86
Figura 98. Interfaz cuentas por cobrar	86
Figura 99. Base de datos cuentas por cobrar.....	87
Figura 100. Interfaz envíos	88
Figura 101. Modal agregar nuevo envío	88
Figura 102. Modal modificar estado de envío	89
Figura 103. Base de datos envíos.....	89
Figura 104. Interfaz del cliente: pedidos.....	90
Figura 105. Prueba de funcionalidad	91

ÍNDICE DE TABLAS

Tabla 1. Requerimientos funcionales: autenticación y autorización.	27
Tabla 2. Requerimientos funcionales: interfaz de usuario.	27
Tabla 3. Requerimientos funcionales: gestión de datos logísticos.	27
Tabla 4. Requerimientos funcionales: gestión de transacciones comerciales.....	28
Tabla 5. Requerimientos no funcionales: autenticación y autorización.	28
Tabla 6. Requerimientos no funcionales: interfaz de usuario.....	29
Tabla 7. Requerimientos no funcionales: gestión de datos logísticos.	29
Tabla 8. Requerimientos no funcionales: gestión de transacciones comerciales.....	29
Tabla 9. Competencias desarrolladas.....	94

CAPITULO I

GENERALIDADES DEL PROYECTO

1.1 Introducción

En el escenario empresarial, la gestión eficiente de datos logísticos es de suma importancia para asegurar la competitividad y sostenibilidad de las organizaciones. La interconexión global de mercados exige la implementación de sistemas informáticos robustos y especializados que faciliten la recolección, procesamiento y análisis de información clave. En este contexto, el presente informe documenta el desarrollo y la implementación de un Sistema de Información Web diseñado para la gestión integral de datos logísticos en la empresa Quality Std S.A de C.V.

La empresa Quality Std S.A de C.V., inmersa en el sector logístico, enfrenta desafíos diarios relacionados con la eficiencia operativa, la trazabilidad de productos y la toma de decisiones basada en información precisa y oportuna. Consciente de estas necesidades, el presente proyecto se enfoca en la creación de un sistema web que integra tecnologías como PHP y MySQL para ofrecer una plataforma ágil y eficaz en la administración de datos logísticos.

A lo largo de este informe, se detallará el proceso de diseño, desarrollo e implementación del sistema, destacando las metodologías y tecnologías empleadas. Asimismo, se analizarán los resultados obtenidos, evidenciando el impacto positivo del sistema en la gestión logística de Quality Std S.A de C.V.

A pesar de las ventajas evidentes de un sistema web, se reconocen los desafíos inherentes, especialmente en áreas críticas como la seguridad y el rendimiento del sistema. Se implementarán prácticas sólidas de seguridad y se realizarán pruebas regulares para abordar posibles vulnerabilidades. La optimización del código y la base de datos será una prioridad constante para mantener un rendimiento óptimo, y se considerará la escalabilidad desde las etapas iniciales para garantizar la capacidad del sistema ante futuros crecimientos.

1.2 Problema a resolver

Quality Std S.A de C.V., una empresa con experiencia en el sector de la Industria Química y Farmacéutica, se enfrenta a desafíos significativos en la gestión de datos logísticos que afectan directamente su eficiencia operativa y capacidad de servicio.

El problema central radica en la ausencia de un sistema de información dedicado para gestionar de manera integral los datos logísticos de la empresa. Quality Std S.A de C.V., al lidiar con una amplia gama de productos químicos y servicios especializados, se encuentra en la necesidad crítica de optimizar la planificación de rutas, el seguimiento de envíos, el control de inventario, la gestión de procesos logísticos complejos, entre otros. La falta de una herramienta especializada ha llevado a ineficiencias en la asignación de recursos, retrasos en la entrega, y en última instancia, a una experiencia del cliente que podría ser mejorada significativamente.

Los problemas a resolver específicos serían los siguientes:

1. Organización y eficiencia: Quality Std S.A de C.V., con su experiencia en la industria química y farmacéutica, maneja una cantidad considerable de datos logísticos, que van desde información de clientes, proveedores, hasta detalles de envíos y control de inventario. Un sistema web centralizado y estructurado sería la clave para organizar y gestionar estos datos de manera eficiente, evitando errores y redundancias que puedan afectar la calidad de los servicios ofrecidos.
2. Rastreo y monitoreo ineficiente: La carencia de un sistema centralizado impide a la empresa obtener una visión en tiempo real de la ubicación y estado de los envíos, lo que conlleva a dificultades en la toma de decisiones y una menor capacidad para abordar problemas logísticos de manera proactiva.
3. Complejidades en el control de inventario: La falta de una herramienta específica dificulta el seguimiento preciso de los niveles de inventario, lo que puede llevar a problemas de suministro y pérdida de oportunidades de venta.
4. Automatización y productividad: Un sistema web automatizado agiliza y optimiza los procesos administrativos y operativos de la empresa Quality Std S.A de C.V. La automatización de tareas como la generación de facturas, la gestión de inventario, la programación de envíos, entre otros, reduce la carga de trabajo manual y mejora la productividad.

5. Seguimiento para clientes: Los clientes carecen de acceso inmediato a información en tiempo real sobre el estado y ubicación de sus envíos. Esto impide la transparencia en la información, lo que puede afectar la confianza del cliente y generar preocupaciones respecto a la seguridad y puntualidad de sus pedidos.
6. Desafíos en la capacidad de almacenamiento y manejo de grandes volúmenes de datos: Con el crecimiento constante de la empresa, la capacidad para manejar grandes volúmenes de datos se convierte en un desafío. Un sistema web escalable podría abordar este problema al permitir la gestión eficiente de datos a gran escala, evitando cuellos de botella y garantizando la disponibilidad de información crucial en todo momento.

La solución propuesta, un "Sistema de Información Web con PHP y MySQL para la gestión de datos logísticos", se tiene como objetivo abordar estos desafíos específicos y proporcionar a Quality Std S.A de C.V. una herramienta que optimice sus procesos logísticos, mejore la eficiencia operativa y, en última instancia, eleve la calidad del servicio ofrecido a sus clientes.

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar un sistema de información web que permita optimizar la gestión de datos logísticos en la empresa Quality Std S.A de C.V., mejorando la eficiencia y precisión en el manejo de la información.

1.3.2 Objetivos específicos

1. Analizar el problema a resolver.
2. Evaluar metodologías de desarrollo.
3. Diseñar diagramas entidad relación y de clases.
4. Diseñar e implementar una base de datos a través del diagrama entidad relación.
5. Implementar un sistema de autenticación y autorización que garantice la seguridad de la información almacenada en el sistema.
6. Diseñar e implementar un sistema con una interfaz amigable e intuitiva para el usuario.

7. Realizar pruebas exhaustivas para validar la funcionalidad, seguridad y rendimiento del sistema antes de su implementación en el entorno de producción.
8. Entregar manuales de usuario.
9. Implementar el sistema y proporcionar capacitación al personal de Quality Std S.A de C.V.

1.4 Justificación

La empresa Quality Std S.A de C.V. ha llevado a la identificación de desafíos críticos que actualmente afectan su eficiencia operativa y, en consecuencia, la calidad del servicio ofrecido. La empresa, especializada en la industria química y farmacéutica, se enfrenta a obstáculos significativos, como la falta de un sistema de información dedicado para la gestión integral de datos logísticos. En este contexto, la implementación de un "Sistema de Información Web con PHP y MySQL para la gestión de datos logísticos" se presenta como una solución estratégica y fundamental para superar estos retos y potenciar el desarrollo sostenible de la empresa.

El sistema propuesto no solo aborda la necesidad de organizar y gestionar eficientemente la gran cantidad de datos logísticos que maneja Quality Std S.A de C.V., sino que también busca mejorar la trazabilidad de productos, optimizar la planificación de rutas, y agilizar la gestión de procesos logísticos complejos. La ausencia de una herramienta especializada ha generado ineficiencias en la asignación de recursos, retrasos en las entregas y ha impactado la experiencia del cliente de manera significativa. La implementación de este sistema web no solo busca resolver estos desafíos, sino que aspira a elevar la competitividad y la calidad del servicio ofrecido por la empresa.

Entre las ventajas que este proyecto aportará se encuentran la centralización y accesibilidad de la plataforma desde cualquier ubicación, facilitando la recopilación, almacenamiento y acceso rápido a datos cruciales. La automatización de procesos manuales reducirá errores operativos y mejorará la eficiencia. Además, la agilización de tiempos de entrega, una mejor utilización de recursos y una reducción de costos operativos son beneficios tangibles. La transparencia y visibilidad proporcionadas por el sistema no solo benefician internamente a la empresa, sino que

también mejoran la experiencia del cliente al permitir un seguimiento en tiempo real de los envíos y una comunicación más fluida.

Para el desarrollo de este proyecto, se requerirá una comprensión profunda del contexto empresarial, así como un análisis detallado respaldado por metodologías de desarrollo y diagramas. La elección de herramientas de desarrollo, como PHP para el desarrollo de aplicaciones web del lado del servidor, MySQL como sistema de gestión de bases de datos relacionales, y la implementación de Bootstrap para crear interfaces web responsivas, creativas e intuitivas para el personal administrativo.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes

En el pasado, la gestión de datos logísticos se realizaba mediante métodos manuales y sistemas basados en papel. Empresas de diversos sectores, incluida la industria química y farmacéutica, dependían de procesos manuales, enfrentando desafíos como la dificultad de acceso rápido a la información, pérdida de documentos físicos y falta de coordinación entre las diferentes áreas de la empresa. La falta de un sistema centralizado dificulta el análisis y la toma de decisiones basadas en datos concretos. Los informes y análisis requeridos para mejorar las operaciones y la rentabilidad de la empresa son tediosos y consumen mucho tiempo cuando se realizan manualmente.

Hoy en día, se reconoce una relación directa entre los resultados exitosos de una compañía y la implementación efectiva de sistemas de información. Sin embargo, es crucial que una empresa no solo se llene de software, sino que analice cuidadosamente sus necesidades y operaciones antes de decidir implementar una estrategia, con el fin de asegurar que contribuya de manera efectiva a los objetivos de la empresa. (Gómez, 2023)

2.2 Marco Conceptual

2.2.1 Sistema de información web.

Es una aplicación informática o conjunto de aplicaciones interconectadas que se ejecutan en un entorno web y están diseñadas para gestionar, procesar y proporcionar acceso a la información a través de un navegador web. Estos sistemas utilizan tecnologías web para facilitar la interacción y el intercambio de datos a través de Internet o una red interna.

Se emplean en cualquier navegador web como Google Chrome, Mozilla Firefox, Internet Explorer, etc sin importar el sistema operativo. De hecho, se puede acceder a los navegadores mencionados a través de un servidor donde se aloje el sistema sin la necesidad de instalarlas en una computadora, laptop, etc. (CreaSystem, 2022)

2.2.2 Servidor web.

Un servidor web es un ordenador que almacena, procesa y entrega archivos de sitios web a los usuarios desde un navegador.

Los servidores web están formados por hardware y software que utilizan el Protocolo de Transferencia de Hipertexto (HTTP) para responder a las peticiones de los usuarios de la web realizadas a través de la World Wide Web. (HOSTINGER, 2023)

2.2.3 Metodología de desarrollo de software.

Una metodología de desarrollo de software es un conjunto de prácticas, técnicas y herramientas utilizadas por los equipos de desarrollo de software para planificar, diseñar, construir, probar y entregar software de alta calidad de manera eficiente y efectiva. (Valtx, s.f.)

2.2.4 Metodología incremental.

En esta metodología para el desarrollo de software se basa en la entrega progresiva y secuencial de partes funcionales del sistema. En cada etapa incremental se agrega una nueva funcionalidad, lo que permite ver resultados de una forma más rápida. (Santander, 2020)

Es particularmente adecuada para proyectos donde los requisitos pueden cambiar con el tiempo y se necesita una entrega temprana de funcionalidades clave. Además, proporciona una mayor visibilidad y control sobre el progreso del proyecto a medida que se desarrolla.

2.2.5 Lenguaje de programación.

Es un conjunto de reglas y sintaxis que le permite a un programador escribir instrucciones que una computadora puede entender y ejecutar. Estas instrucciones, conocidas como código fuente, son escritas en un formato legible por humanos y luego traducidas a un lenguaje que la computadora puede entender, conocido como código de máquina o código ejecutable. (UNAM, 2017)

2.2.6 Backend.

Es la parte lógica de un sitio, este se encarga de la lógica de negocio, de recibir y devolver datos procesados a las apps y sitios web, de forma que facilite la navegación y se garantice el funcionamiento y la seguridad de diferentes funciones. (Ken, 2023)

2.2.7 PHP.

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente diseñado para el desarrollo de aplicaciones web y la creación de páginas dinámicas. PHP se ejecuta en el servidor web y es un componente fundamental en el desarrollo de sitios web interactivos. (PHP, s.f.)

Algunas características de PHP son:

1. Embeddable, es decir, puede integrarse con código HTML permitiendo a los desarrolladores mezclar código estático y dinámico en una página web
2. Multiplataforma, es compatible con varios sistemas operativos y puede ejecutarse en servidores web.
3. Actualmente es de los lenguajes de programación más utilizados y es compatible con una amplia variedad de base de datos.

2.2.8 JavaScript

Es un lenguaje de programación de alto nivel, interpretado y orientado a objetos, utilizado principalmente en el desarrollo web, pensado para agregar potencial de interacción y dinamismo a las páginas web. (AWS, s.f.)

JavaScript proporciona una amplia gama de funcionalidades, como manipulación del DOM (Document Object Model) para interactuar con elementos de una página web, manejo de eventos, comunicación con servidores a través de AJAX, creación de animaciones, validación de formularios y mucho más.

2.2.9 Frontend.

Es el diseño de un sitio web, desde la estructura del sitio hasta los estilos como colores, fondos, tamaños hasta llegar a las animaciones y efectos. (García, 2021)

Es todo el código que se ejecuta en el navegador de un usuario, al que se le denomina una aplicación cliente, es decir, todo lo que el visitante ve y experimenta de forma directa.

2.2.10 HTML.

HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language). Es un estándar que se utiliza para la creación, el diseño de páginas web y es la base sobre la cual se construye la estructura de la mayoría de los contenidos web. (MDN Web Docs, s.f.)

HTML proporciona un conjunto de etiquetas (tags) que describen la estructura y el contenido de una página web. Cada elemento en una página web, como títulos, párrafos, imágenes, enlaces, formularios, entre otros, se define mediante estas etiquetas. Las etiquetas le dicen al navegador cómo mostrar el contenido y cómo organizarlo.

2.2.11 CSS.

Las siglas CSS (Cascading Style Sheets) significan “Hojas de estilo en cascada” permiten crear páginas web atractivas.

Se le denomina estilos en cascada porque se lee, procesa y aplica el código desde arriba hacia abajo (siguiendo un patrón denominado herencia que trataremos más adelante) y en el caso de existir ambigüedad (código que se contradice), se siguen una serie de normas para resolver dicha ambigüedad. (Manz, s.f.)

2.2.12 Framework.

Es una estructura conceptual y un conjunto de herramientas que proporciona una base para el desarrollo de software. Este conjunto de herramientas incluye bibliotecas de código, estándares, convenciones y componentes predefinidos que facilitan la creación y el mantenimiento de aplicaciones.

Son usados por programadores porque permiten acelerar el trabajo y favorecer que este sea colaborativo, reducir errores y obtener un resultado de más calidad. (Unir, 2022)

2.2.13 Bootstrap.

Es un framework CSS de desarrollo web gratuito y de código abierto. Está diseñado para facilitar el proceso de desarrollo de los sitios web responsivos proporcionando una colección de sintaxis para diseños de plantillas. (HOSTINGER, 2023).

Proporciona una serie de estilos predefinidos, componentes y utilidades que permiten a los desarrolladores crear interfaces de usuario atractivas y compatibles con dispositivos de diferentes tamaños, como computadoras de escritorio, tabletas y teléfonos móviles.

2.2.14 Base de datos.

Es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Los datos de los tipos más comunes de bases de datos en funcionamiento actualmente se suelen utilizar como estructuras de filas y columnas en una serie de tablas para aumentar la eficacia del procesamiento y la consulta de datos. Así, se puede acceder, gestionar, modificar, actualizar, controlar y organizar fácilmente los datos. (ORACLE, s.f.)

2.2.15 Base de datos relacional

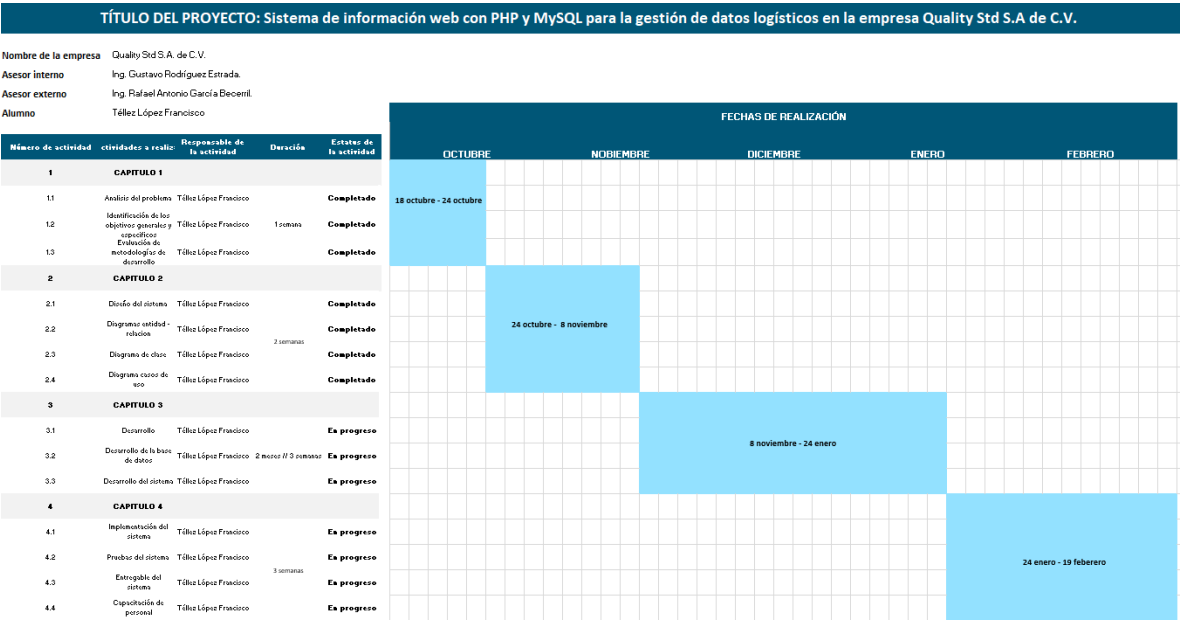
Una base de datos relacional es un conjunto de información que organiza los datos en relaciones predefinidas donde se almacenan en una o más tablas (o "relaciones") de columnas y filas, lo que ayuda a ver y comprender la relación entre las distintas estructuras de datos. Las relaciones son una conexión lógica entre diferentes tablas, que se basa en la interacción entre ellas. (Google Cloud, s.f.)

2.2.16 MySQL.

Es un sistema de gestión de bases de datos relacional basado en el lenguaje de consulta estructurado SQL. Es ampliamente utilizado en el desarrollo de aplicaciones web y otras aplicaciones que requieren el almacenamiento y recuperación eficientes de datos. Ofrece características como la capacidad de manejar grandes volúmenes de datos, soporte para transacciones, seguridad de datos y una comunidad activa que proporciona soporte y mejoras continuas.

MySQL es una opción razonable para ser usado en ámbito empresarial. Al estar basado en código abierto permite a pequeñas empresas y desarrolladores disponer de una solución fiable y estandarizada para sus aplicaciones. (Robledano, 2019)

Figura 1. Cronograma de actividades del proyecto



CAPÍTULO III

DESARROLLO

3.1 Diseño del sistema de información web

El diseño del sistema web de gestión de información se basará en los requerimientos identificados durante el diagnóstico realizado en conjunto con las áreas operativas de la empresa y las respuestas obtenidas a preguntas clave formuladas en el departamento de logística.

¿Cómo podría mejorar la eficiencia operativa de la empresa?

La implementación de un sistema de información web permitirá automatizar procesos administrativos y operativos, reduciendo la carga de trabajo manual. Esto agilizará la gestión de datos logísticos, optimizando la planificación de rutas, seguimiento de envíos y control de inventario, lo que contribuirá significativamente a mejorar la eficiencia operativa de la empresa.

¿Cómo podría reducir costos y aumentar la rentabilidad?

El sistema propuesto optimizará la asignación de recursos, agilizará tiempos de entrega y reducirá errores operativos, lo que conducirá a una disminución de costos operativos. Además, la automatización de tareas administrativas y la mejora en la eficiencia contribuirán a aumentar la rentabilidad de la empresa.

¿Cómo podría optimizar la gestión de inventario y recursos?

El sistema permitirá un seguimiento preciso de inventario en tiempo real y facilitará la planificación de recursos mediante la automatización de procesos. La centralización de datos logísticos contribuirá a una gestión más eficiente de inventario y recursos, evitando problemas de suministro y pérdida de oportunidades de venta.

¿Cómo podría aumentar la precisión y calidad de los datos?

La implementación de un sistema de información web con PHP y MySQL garantizará la integridad y consistencia de los datos logísticos. La utilización de formularios y vistas de datos permitirá a los usuarios agregar, editar y eliminar información de manera controlada, mejorando así la precisión y calidad de los datos almacenados.

¿Cómo podría asegurar la seguridad y privacidad de los datos?

Se implementarán prácticas sólidas de seguridad, incluyendo un sistema de autenticación y autorización. Además, se realizarán pruebas para abordar posibles vulnerabilidades. La seguridad de la información almacenada en el sistema será una prioridad constante para garantizar la confidencialidad y privacidad de los datos.

¿Cómo facilitaría el acceso y la colaboración con datos en tiempo real?

El sistema de información web permitirá a los usuarios acceder a la información desde cualquier lugar con conexión a internet, facilitando la colaboración y toma de decisiones en tiempo real. La interfaz intuitiva y responsiva, junto con la centralización de datos, mejorará la accesibilidad y agilizará la colaboración entre diferentes áreas de la empresa

¿Cómo podría adaptarse al crecimiento y cambios de la empresa?

Se considerará la escalabilidad desde las etapas iniciales del desarrollo del sistema. Esto garantizará que el sistema pueda manejar grandes volúmenes de datos a medida que la empresa crezca. La flexibilidad y personalización del sistema permitirán adaptarse a cambios en las operaciones y necesidades específicas de la empresa.

¿Cómo contribuiría a una mejor toma de decisiones?

La disponibilidad de información en tiempo real, la centralización de datos logísticos y la mejora en la precisión de los datos proporcionarán una base sólida para la toma de decisiones informadas. La empresa contará con análisis y reportes más rápidos y precisos, facilitando la identificación de áreas de mejora y la implementación de estrategias efectivas.

3.2 Requerimientos

Para desarrollar un sistema de información web eficiente que satisfaga las necesidades de gestión de datos logísticos de Quality Std S.A de C.V., se tiene que identificar y definir claramente los requerimientos del sistema. A continuación, recopilaremos los requerimientos funcionales y no funcionales para la gestión de datos logísticos en la empresa Quality Std S.A de C.V:

3.2.1 Requerimientos funcionales

Tabla 1. Requerimientos funcionales: autenticación y autorización.

Módulo: Autenticación y autorización		
ID REQ	REQUERIMIENTO	DESCRIPCIÓN
RF1	Registro de usuarios	Permitir que los usuarios se registren en el sistema proporcionando información básica como nombre de usuario, contraseña, y datos de contacto.
RF2	Inicio de sesión	Permitir que los usuarios inicien sesión en el sistema utilizando sus credenciales previamente registradas.
RF3	Gestión de roles	Permitir la asignación de roles a los usuarios, como administrador, gerente, y empleado, para controlar sus permisos de acceso.
RF4	Control de acceso	Garantizar que los usuarios solo puedan acceder a las funciones y datos para los que tienen permisos autorizados según su rol.

Tabla 2. Requerimientos funcionales: interfaz de usuario.

Módulo: Interfaz de usuario		
ID REQ	REQUERIMIENTO	DESCRIPCIÓN
RF5	Diseño responsivo	Desarrollar una interfaz de usuario que se adapte de manera fluida solamente a computadoras de escritorio y laptops.
RF6	Navegación intuitiva	Diseñar una interfaz de usuario intuitiva y fácil de usar, con menús y botones claramente etiquetados para facilitar la navegación del usuario.
RF7	Interacción con datos	Permitir a los usuarios interactuar con los datos de manera eficiente mediante formularios de entrada, tablas de visualización y opciones de búsqueda y filtrado.

Tabla 3. Requerimientos funcionales: gestión de datos logísticos.

Módulo: Gestión de datos logísticos		
ID REQ	REQUERIMIENTO	DESCRIPCIÓN

RF8	Registro de envíos	Permitir a los usuarios registrar nuevos envíos en el sistema, proporcionando detalles como origen, destino, fecha estimada de entrega y estado del envío.
RF9	Seguimiento de envíos	Permitir a los usuarios rastrear el estado y la ubicación de los envíos en tiempo real utilizando números de seguimiento únicos.
RF10	Gestión de inventario	Facilitar la gestión de inventario permitiendo a los usuarios realizar un seguimiento de los productos almacenados, niveles de stock y movimientos de inventario.

Tabla 4. Requerimientos funcionales: gestión de transacciones comerciales.

Módulo: Gestión de transacciones comerciales		
ID REQ	REQUERIMIENTO	DESCRIPCIÓN
RF11	Generar facturas	Permitir la generación de facturas para registrar transacciones de ventas realizadas a clientes.
RF12	Elaboración de cotización o presupuestos	Permitir la creación de cotizaciones o presupuestos para presentar ofertas a clientes potenciales.
RF13	Órdenes de compras	Permitir la creación de órdenes de compras para registrar transacciones de compra de productos a proveedores.

3.2.2 Requerimientos no funcionales:

Tabla 5. Requerimientos no funcionales: autenticación y autorización.

Módulo: Autenticación y autorización		
ID REQ	REQUERIMIENTO	DESCRIPCIÓN
RNF1	Seguridad	Implementar medidas de seguridad, como encriptación de contraseñas y protección contra ataques, para garantizar la confidencialidad de los datos de los usuarios.
RNF2	Rendimiento	Optimizar el rendimiento del sistema para garantizar tiempos de respuesta rápidos, incluso durante periodos de alta carga de usuarios.
RNF3	Escalabilidad	Diseñar el sistema con la capacidad de escalar para manejar un aumento en el

	número de usuarios y la cantidad de datos sin pérdida de rendimiento.
--	---

Tabla 6. Requerimientos no funcionales: interfaz de usuario.

Módulo: Interfaz de usuario		
ID REQ	REQUERIMIENTO	DESCRIPCIÓN
RNF4	Compatibilidad del navegador	Garantizar que la interfaz de usuario sea compatible con una amplia gama de navegadores web modernos, como Google Chrome, Mozilla Firefox y Microsoft Edge.
RF5	Accesibilidad	Cumplir con estándares de accesibilidad web para garantizar que el sistema sea utilizable

Tabla 7. Requerimientos no funcionales: gestión de datos logísticos.

Módulo: Gestión de datos logísticos		
ID REQ	REQUERIMIENTO	DESCRIPCIÓN
RNF6	Consistencia de datos	Implementar controles de integridad de datos para garantizar la consistencia y precisión de la información almacenada en la base de datos.
RNF7	Respaldo de datos	Establecer rutinas de respaldo regulares para proteger los datos del sistema contra pérdidas, garantizando la disponibilidad continua de la información.

Tabla 8. Requerimientos no funcionales: gestión de transacciones comerciales.

Módulo: Gestión de transacciones comerciales		
ID REQ	REQUERIMIENTO	DESCRIPCIÓN
RNF8	Eficiencia en el procesamiento de transacciones	Garantizar que el sistema pueda procesar un alto volumen de transacciones de manera eficiente y sin retrasos significativos.
RNF9	Fiabilidad y disponibilidad	Asegurar que el sistema esté disponible en todo momento y que sea confiable para completar las transacciones sin pérdida de datos.
RNF10	Auditoría y registro de transacciones	Implementar mecanismos de auditoría que registren todas las transacciones

		realizadas en el sistema para cumplir con requisitos legales y de seguridad.
--	--	--

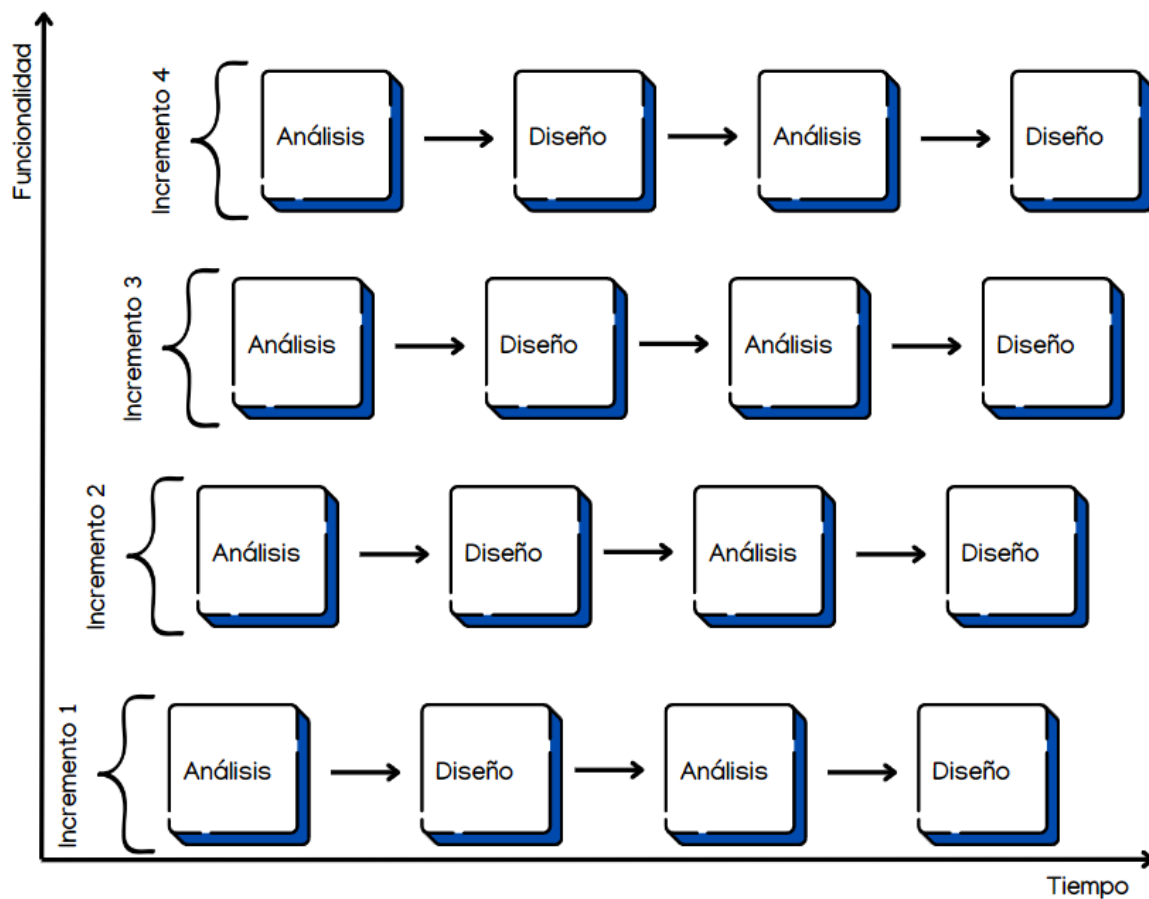
3.3 Diseño del software

El diseño del software para el sistema de información web destinado a la gestión de datos logísticos en Quality Std S.A de C.V. se basará en la aplicación de la metodología incremental. Esta metodología fue seleccionada debido a su capacidad para adaptarse a los cambios en los requisitos del proyecto, su enfoque iterativo y su capacidad para proporcionar entregas tempranas y continuas de funcionalidades. También se utilizarán herramientas y tecnologías específicas para garantizar un diseño eficiente y cumplir con los objetivos establecidos.

3.3.1 Metodología para el desarrollo de software (Incremental)

La metodología incremental es un enfoque de desarrollo de software que divide el proceso en una serie de iteraciones o incrementos. Cada iteración resulta en una versión operativa del sistema que incluye una funcionalidad adicional o mejorada. A diferencia de los enfoques tradicionales, como la metodología en cascada, la metodología incremental permite una mayor flexibilidad y adaptabilidad a medida que los requisitos del proyecto evolucionan.

El desarrollo del sistema se dividirá en múltiples iteraciones, cada una de las cuales seguirá un ciclo de vida que abarca desde la planificación hasta la implementación y la entrega de funcionalidades específicas. Las iteraciones se ejecutarán de manera secuencial, con cada una construyendo sobre el trabajo realizado en las iteraciones anteriores.

Figura 2. Metodología incremental

Al finalizar cada iteración, se espera obtener un conjunto de entregables que representen una versión funcional y operativa del sistema. Estos entregables pueden incluir módulos específicos completados, actualizaciones de la interfaz de usuario, funcionalidades adicionales implementadas y pruebas realizadas para garantizar la calidad del software.

3.3.2 Diseño de la arquitectura del sistema

Se utilizará una arquitectura de tres capas, dividiendo el sistema en capas de presentación, lógica de negocio y acceso a datos. Esto permitirá una mayor modularidad y mantenimiento del sistema.

3.3.3 Backend (Lógica de negocio)

Se utilizará PHP como lenguaje de programación del lado del servidor para la implementación de la lógica de negocio.

Se desarrollarán controladores y modelos para gestionar la interacción entre la interfaz de usuario y la base de datos.

Se implementarán las prácticas de la metodología incremental para facilitar la entrega progresiva de funcionalidades.

3.3.4 Frontend (Interfaz de usuario)

Se utilizará HTML para la estructura básica de las páginas web, asegurando la compatibilidad y accesibilidad.

Se empleará CSS para el diseño y la presentación de la interfaz de usuario.

Se integrará el framework Bootstrap para facilitar el desarrollo de interfaces web y mejorar la experiencia del usuario.

También se hará uso de JavaScript para hacer un poco más dinámico el sistema web, además que nos va a permitir hacer algunas peticiones con el servidor.

3.3.5 Integración de tecnologías

Se integrarán las tecnologías PHP, JavaScript, MySQL, HTML, CSS y Bootstrap de manera coherente para lograr una aplicación web funcional y atractiva.

Se asegurará la compatibilidad del sistema con diferentes navegadores web para garantizar una experiencia consistente para los usuarios.

3.3.6 Diseño de la base de datos

El diseño de la estructura de la base de datos se ha concebido estratégicamente para optimizar la eficiencia en la búsqueda y recuperación de información, priorizando la adaptabilidad futura y la escalabilidad del sistema. Se ha eliminado datos redundantes, así como en la preservación de la integridad de los datos para generar consultas claras.

La creación de las tablas se ha basado en los requisitos específicos del sistema, implementando técnicas de normalización para estandarizar la estructura de las mismas. Para la gestión de la base de datos, se ha elegido MySQL, una solución de código abierto reconocida por su rendimiento excepcional y su eficiencia en el consumo de recursos. La optimización de los

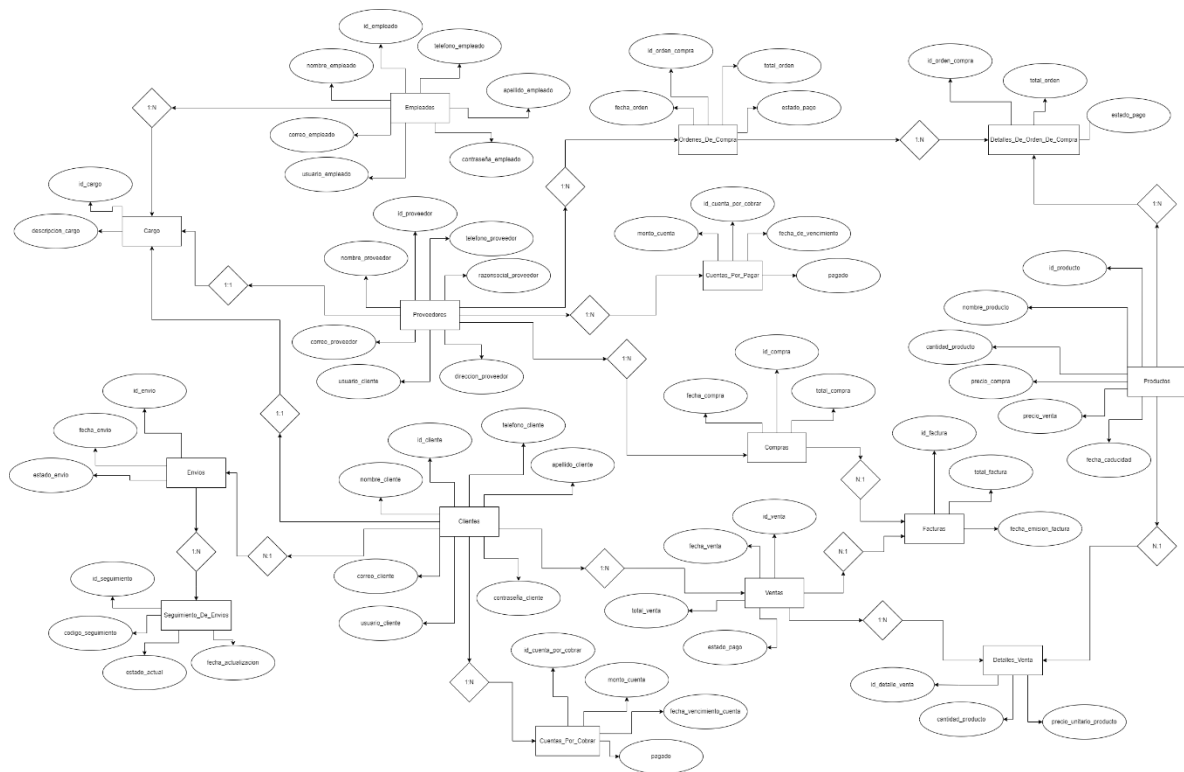
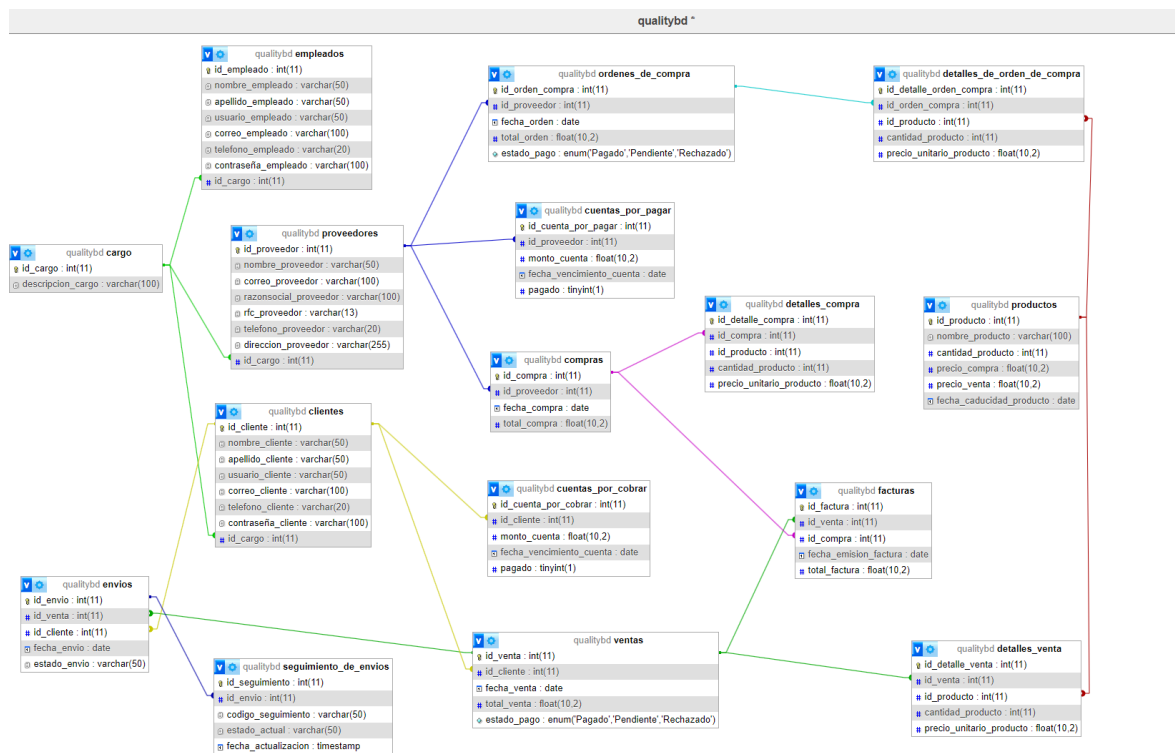


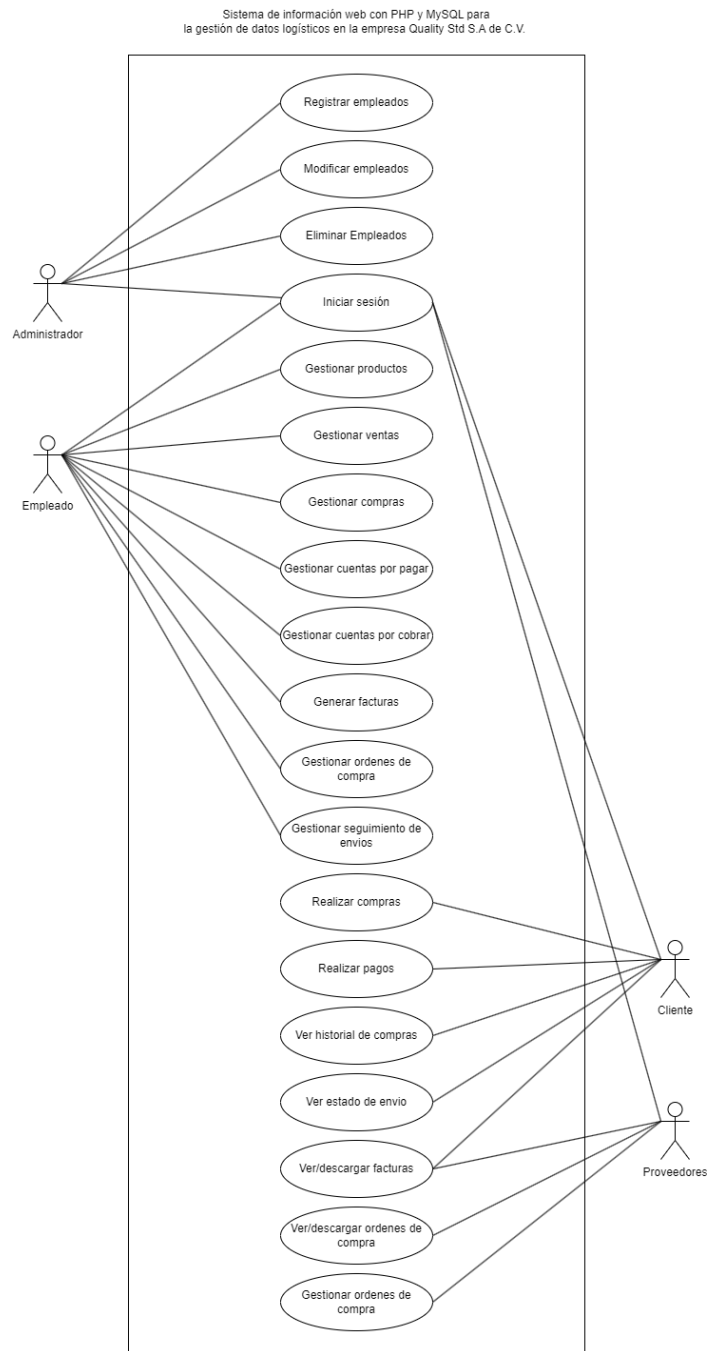
Figura 4. Modelo entidad-relación phpmyadmin



3.3.7 Diagramas de caso de uso general.

Los diagramas de caso de uso muestran el comportamiento esperado del sistema, con la que se representan procesos, sistemas, programación, donde todos los objetos involucrados se estructuran y se relacionan entre sí. Aquí se verán los roles de los actores, la simbología y los casos del sistema.

Figura 5. Diagrama de caso de uso general.



3.4 Sistema de información.

Tomando en cuenta las tecnologías a utilizar, para el frontend, se define un diseño general para todas las páginas que tendrá el sistema de información, esto con el fin de aprovechar nuestro lenguaje de programación backend “PHP” para poder modularizar y reutilizar el código.

Este diseño tiene establecido las referencias para poder implementar correctamente CSS y el framework bootstrap para el buen funcionamiento y se aprecien los estilos aplicados. Este diseño se compone de un encabezado, un menú lateral y un pie de página.

Figura 6. Menú lateral

```
<!-- Menu lateral -->
<div id="sidebarMenu" class="collapse d-lg-block sidebar">
<div class="position-sticky">
<div class="list-group list-group-flush mx-3 mt-4">
<a class="nav-item brand" href="#">

</a>
<a href="<php echo $base_url; >?/paginas/index.php" class="list-group-item list-group-item-action py-2 ripple <php echo basename($_SERVER['PHP_SELF']) == 'index.php' ? 'active' : ''>>

<span>Inicio</span>
</a>
<a href="<php echo $base_url; >?/paginas/clientes/clientes.php" class="list-group-item list-group-item-action py-2 ripple <php echo basename($_SERVER['PHP_SELF']) == 'clientes.php' ? 'active' : ''>>

<span>Clientes</span>
</a>
<a href="<php echo $base_url; >?/paginas/proveedores/proveedores.php" class="list-group-item list-group-item-action py-2 ripple <php echo basename($_SERVER['PHP_SELF']) == 'proveedores.php' ? 'active' : ''>>

<span>Proveedores</span>
</a>
<a href="<php echo $base_url; >?/paginas/empleados/empleados.php" class="list-group-item list-group-item-action py-2 ripple <php echo basename($_SERVER['PHP_SELF']) == 'empleados.php' ? 'active' : ''>>

<span>Empleados</span>
</a>
<a href="<php echo $base_url; >?/paginas/productos/productos.php" class="list-group-item list-group-item-action py-2 ripple <php echo basename($_SERVER['PHP_SELF']) == 'productos.php' ? 'active' : ''>>

<span>Productos</span>
</a>
<a href="<php echo $base_url; >?/paginas/compras/detalles_compras.php" class="list-group-item list-group-item-action py-2 ripple <php echo basename($_SERVER['PHP_SELF']) == 'detalles_compras.php' ? 'active' : ''>>

<span>Compras</span>
</a>
<a href="<php echo $base_url; >?/paginas/ventas/detalles_ventas.php" class="list-group-item list-group-item-action py-2 ripple <php echo basename($_SERVER['PHP_SELF']) == 'detalles_ventas.php' ? 'active' : ''>>

<span>Ventas</span>
</a>
<a href="<php echo $base_url; >?/paginas/envios/detalles_envios.php" class="list-group-item list-group-item-action py-2 ripple <php echo basename($_SERVER['PHP_SELF']) == 'detalles_envios.php' ? 'active' : ''>>

<span>Envios</span>
</a>
<a href="<php echo $base_url; >?/paginas/cuentasporcobrar/cuentas_cobrar.php" class="list-group-item list-group-item-action py-2 ripple <php echo basename($_SERVER['PHP_SELF']) == 'cuentas_cobrar.php' ? 'active' : ''>>

<span>Cuentas por cobrar</span>
</a>
<a href="<php echo $base_url; >?/paginas/cuentasporpagar/cuentas_pagar.php" class="list-group-item list-group-item-action py-2 ripple <php echo basename($_SERVER['PHP_SELF']) == 'cuentas_pagar.php' ? 'active' : ''>>

<span>Cuentas por pagar</span>
</a>
</div>
</div>
</nav>
```

Con este diseño base podemos reutilizar código para no ponerlo en cada una de nuestras páginas, simplemente hacemos uso de código PHP en nuestros módulos para poder mandar a llamar nuestro diseño.

Figura 7. Referencia al header

```
<?php include("../includes/header.php"); ?>
```

Por otra parte, en CSS se establece el estilo visual del sistema web. Definiendo el color de fondo del cuerpo de la página, el diseño y la apariencia del menú lateral y el estilo de ciertos elementos como las tablas. Además, se asegura que el menú lateral se mantenga visible en la parte izquierda de la pantalla incluso al hacer desplazamiento hacia abajo en la página.

Figura 8. CSS diseño a elementos

```
body {
  background-color: #F4F4F4 !important;
}
@media (min-width: 991.98px) {
  main {
    padding-left: 240px;
  }
}

.sidebar {
  background-color: #FFFFFF;
  position: fixed;
  top: 0;
  bottom: 0;
  left: 0;
  padding: 58px 0 0;
  box-shadow: 0 2px 5px 0 #000000 / 5%, 0 2px 10px 0 #000000 / 5%;
  width: 240px;
  z-index: 600;
}

@media (max-width: 991.98px) {
  .sidebar {
    width: 100%;
  }
}

.sidebar .active {
  border-radius: 5px;
  box-shadow: 0 2px 5px 0 #000000 / 16%, 0 2px 10px 0 #000000 / 12%;
}

.sidebar-sticky {
  position: relative;
  top: 0;
  height: calc(100vh - 48px);
  padding-top: 0.5rem;
  overflow-x: hidden;
  overflow-y: auto;
}

.fondo-tabla {
  position: relative;
  z-index: 1;
  background-color: #FFFFFF;
  padding: 50px;
  height: 70vh;
  border-radius: 10px;
}
```

La implementación del framework bootstrap en este sistema Bootstrap nos proporciona un conjunto de componentes preestablecidos y estilos CSS que puedes utilizar fácilmente en tu

proyecto. Esto facilita la creación de una interfaz de usuario atractiva y coherente sin tener que escribir mucho código desde cero.

Para la conexión a la base de datos creamos una clase especial llamada “bd.php” donde igual vamos a reutilizar código, esto con el fin de no crear la conexión en todas nuestras clases.

Figura 9. Conexión a la base de datos

```
1  <?php
2  $conexion=mysqli_connect("localhost","root","","qualitybd");
3  ?>
```

Se creo la variable \$conexion para almacenar la conexión a la base de datos, con la función mysqli_connect con cuatro parámetros, pero por cuestiones de seguridad no se pusieron los datos reales, como el nombre del servidor de la base de datos, el usuario para conectarse a la base de datos, la contraseña, solamente se dejó “qualitybd” ya que es la base de datos de la empresa.

Para cerrar sesión se creó una clase en específico para no repetir código en los módulos, donde nos permite destruir las variables de la sesión y redirige al usuario a la página para poder iniciar sesión (login).

Figura 10. Cerrar sesión de la base de datos

```
1  <?php
2  // Inicia la sesión
3  session_start();
4
5  session_destroy();
6
7  header("Location:../paginas/login.php");
8  exit;
```

En general se implementó funcionalidades de paginación y búsqueda para mejorar la experiencia del usuario al interactuar con grandes conjuntos de datos. Estas características se aplican de manera generalizada en varios módulos del sistema.

La paginación se realiza de manera eficiente, dividiendo los resultados utilizando un número fijo de registros por página con el fin de no sobrecargar la página y hacerla más lenta. Calculamos automáticamente el número total de páginas en función del total de registros y el número de registros por página, lo que permite una navegación fluida entre páginas mediante enlaces de paginación

Figura 11. Paginación

```
// registros por página
$registros_por_pagina = isset($_GET['num_registros']) ? $_GET['num_registros'] : 10;

// Obtener el total de registros
$total_registros_query = "SELECT COUNT(*) AS total FROM Empleados";
$total_resultado = mysqli_query($conexion, $total_registros_query);
$total_fila = mysqli_fetch_assoc($total_resultado);
$total_registros = $total_fila['total'];

// Calcular el número total de páginas
$total_paginas = ceil($total_registros / $registros_por_pagina);

// Obtener el número de página actual
$pagina_actual = isset($_GET['pagina']) ? $_GET['pagina'] : 1;

// Calcular el offset para la consulta SQL
$offset = ($pagina_actual - 1) * $registros_por_pagina;
```

Además, se requiere de estructura HTML con PHP para poder mostrar enlaces a diferentes páginas de resultados.

Figura 12. Lógica para la paginación

```
<nav aria-label="Page navigation">
  <ul class="pagination">
    <?php for ($i = 1; $i <= $total_paginas; $i++) : ?>
      <li class="page-item <?php echo $i == $pagina_actual ? 'active' : ''; ?>">
        <a class="page-link" href="?pagina=<?php echo $i; ?>"><?php echo $i; ?></a>
      </li>
    <?php endfor; ?>
  </ul>
</nav>
```

Por otro lado, la funcionalidad de búsqueda proporciona a los usuarios la capacidad de encontrar rápidamente los registros deseados. Se construyeron consultas SQL que aplican la condición de búsqueda según los términos ingresados por el usuario. Utilizamos el operador LIKE para realizar búsquedas que coincidan con los campos de la tabla necesarios ya que esto depende de cada módulo.

Figura 13. Consulta para el buscador

```
// Obtener los registros
$consulta_base = "SELECT * FROM Empleados";

// Verificar si se ha enviado un término de búsqueda
if (isset($_GET['buscar']) && !empty($_GET['buscar'])) {
    $buscar = mysqli_real_escape_string($conexion, $_GET['buscar']);
    // Condición de búsqueda
    $consulta_base .= " WHERE nombre_empleado LIKE '%$buscar%' OR apellido_empleado LIKE '%$buscar%' OR usuario_empleado LIKE '%$buscar%' OR correo_empleado LIKE '%$buscar%'";
}

// Agregar la cláusula LIMIT para paginación
$consulta_base .= " LIMIT $offset, $registros_por_pagina";

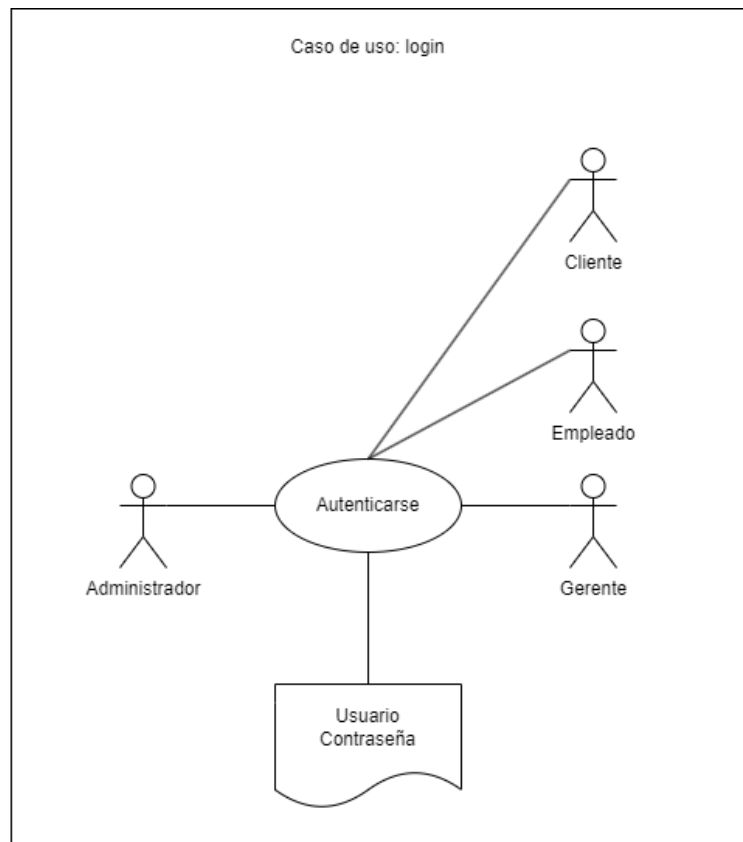
$resultado = mysqli_query($conexion, $consulta_base);
?>
```

Figura 14. Buscador

```
<div class="col-4"></div>
<div class="col-auto">
    <label for="buscar" class="col-form-label">Buscar:</label>
</div>
<div class="col-md-4">
    <input type="text" id="buscar" name="buscar" class="form-control" value="<?php echo isset($_GET['buscar']) ? $_GET['buscar'] : ''; ?>"
</div>
</div>
```

3.4.1 Módulo login

Figura 15. Caso de uso: login



El sistema cuenta con módulos donde cada uno realiza diferentes acciones, el uso y el acceso a estas funcionalidades depende exclusivamente del usuario y el perfil al que está

relacionado, teniendo en cuenta que el administrador del sistema es quien configura el acceso, el empleado y gerente pueden hacer uso de los módulos para la gestión de datos logísticos de quality, por último, el cliente únicamente puede ingresar para revisar su historial de pedidos y ver el proceso de sus envíos.

Figura 16. Validaciones login

```
<?php
// archivo de conexión a la base de datos
include 'bd.php';

// Formulario
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Obtener los datos del formulario
    $usuario = $_POST['usuario'];
    $contrasena = $_POST['contrasena'];

    // Consultar en la tabla de Empleados
    $query_empleado = "SELECT * FROM Empleados WHERE correo_empleado = '$usuario' AND contraseña_empleado = '$contrasena'";
    $result_empleado = mysqli_query($conexion, $query_empleado);

    // Consultar en la tabla de Clientes
    $query_cliente = "SELECT * FROM Clientes WHERE correo_cliente = '$usuario' AND contraseña_cliente = '$contrasena'";
    $result_cliente = mysqli_query($conexion, $query_cliente);

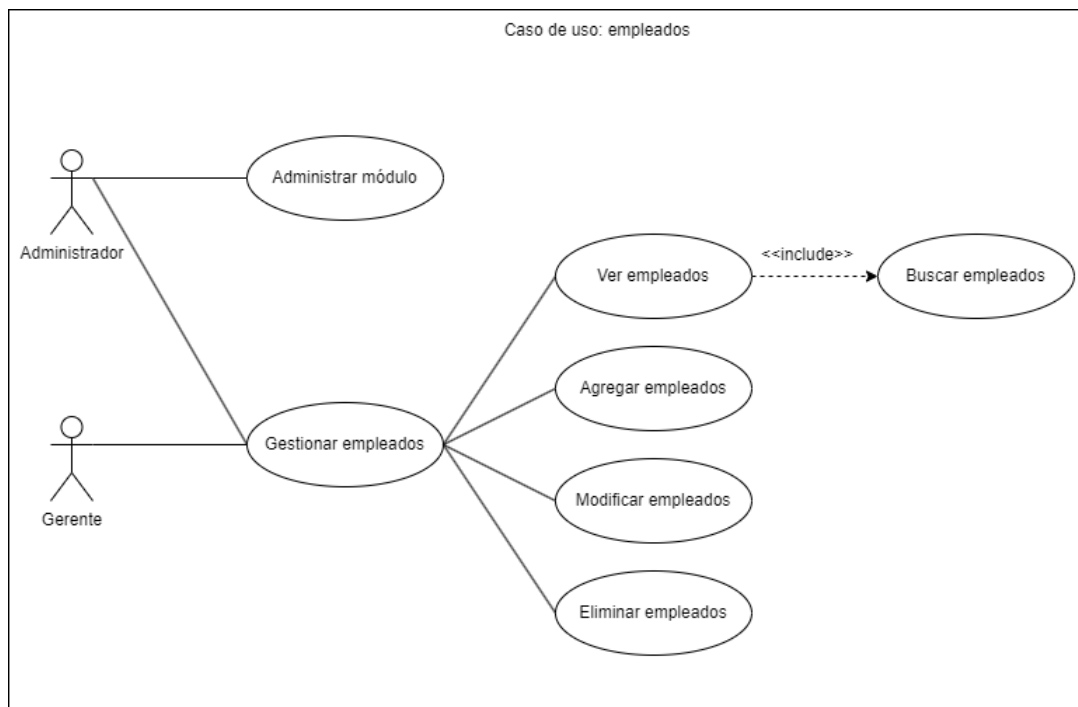
    // Verificar si se encontró el usuario en alguna tabla
    if (mysqli_num_rows($result_empleado) > 0) {
        // Si el usuario que quiere ingresar es un empleado, redirigir a index.php
        header("location: ../paginas/index.php");
        exit();
    } elseif (mysqli_num_rows($result_cliente) > 0) {
        session_start(); // Iniciar la sesión, El usuario es un cliente, establecer la variable de sesión y redirigir a cli_index.php
        $cliente = mysqli_fetch_assoc($result_cliente);
        $_SESSION['id_cliente'] = $cliente['id_cliente'];
        header("location: ../paginas/cli_index.php");
        exit();
    } else {
        // Si los datos son incorrectos, redirigir de vuelta a login.php con un parametro de error
        header("location: ../paginas/login.php?error=true");
        exit();
    }
}
?>
```

Si algún usuario intenta ingresar con datos incorrectos el sistema hace una validación y es redirigido al login con un mensaje de error dentro del frontend indicándole que su “Usuario o contraseña están incorrectos”

Figura 17. Login usuario o contraseña incorrecta

```
<?php
if (isset($_GET['error']) && $_GET['error'] == 'true') {
    echo '<div class="alert alert-danger text-center" role="alert">Usuario o contraseña incorrectos</div>';
}
?>
```

3.4.2 Módulo empleados

Figura 18. Caso de uso: empleados

La creación de la tabla empleados cuenta con un id_empleado que se establece como clave primaria, con opción de “auto_increment” esto con el fin de que el valor de esta columna se incrementará automáticamente para cada nuevo registro. Por otro lado, tenemos una clave foránea “id_cargo” de la tabla “Cargo” para poder asignarle un cargo(rol) al empleado.

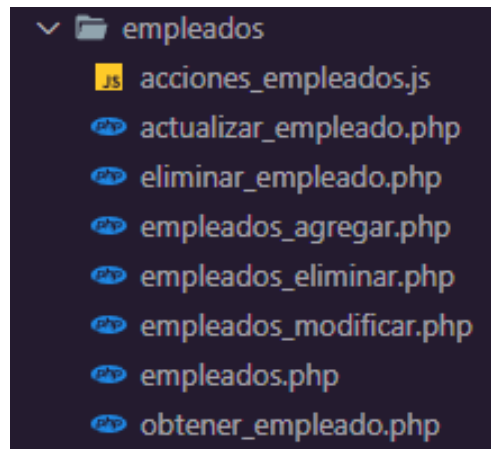
Figura 19. Tabla empleados SQL

```

6 • CREATE TABLE Empleados (
7     id_empleado INT PRIMARY KEY AUTO_INCREMENT,
8     nombre_empleado VARCHAR(50),
9     apellido_empleado VARCHAR(50),
10    usuario_empleado VARCHAR(50),
11    correo_empleado VARCHAR(100),
12    telefono_empleado VARCHAR(20),
13    contraseña_empleado VARCHAR(100),
14    id_cargo INT,
15    FOREIGN KEY (id_cargo) REFERENCES Cargo(id_cargo)
16 );
  
```

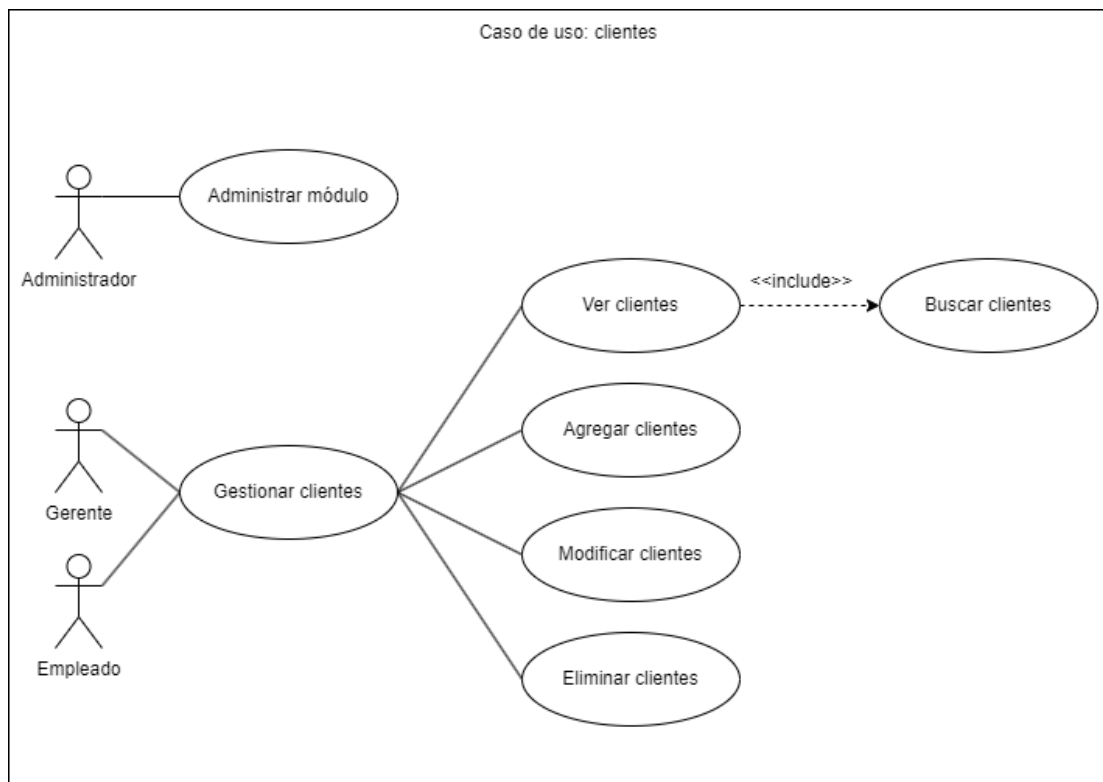
El módulo empleado nos permite gestionar la información de todos los empleados, nos permite visualizar, registrar, editar, actualizar y eliminar empleados.

Figura 20. Módulo empleados



3.4.3 Módulo clientes:

Figura 21. Caso de uso: clientes



La creación de la tabla clientes cuenta con un “id_cliente” que se establece como clave primaria, con opción de “auto_increment” esto con el fin de que el valor de esta columna se incrementará automáticamente para cada nuevo registro. Por otro lado, tenemos una clave foránea “id_cargo” de la tabla “Cargo” para poder identificar que es un usuario de tipo “cliente” y se

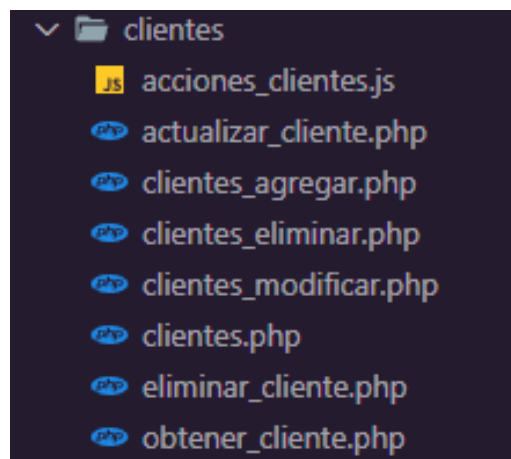
puedan establecer los privilegios que tendrán los clientes en base a la lógica de negocio de Quality Std S.A de C.V.

Figura 22. Tabla clientes SQL

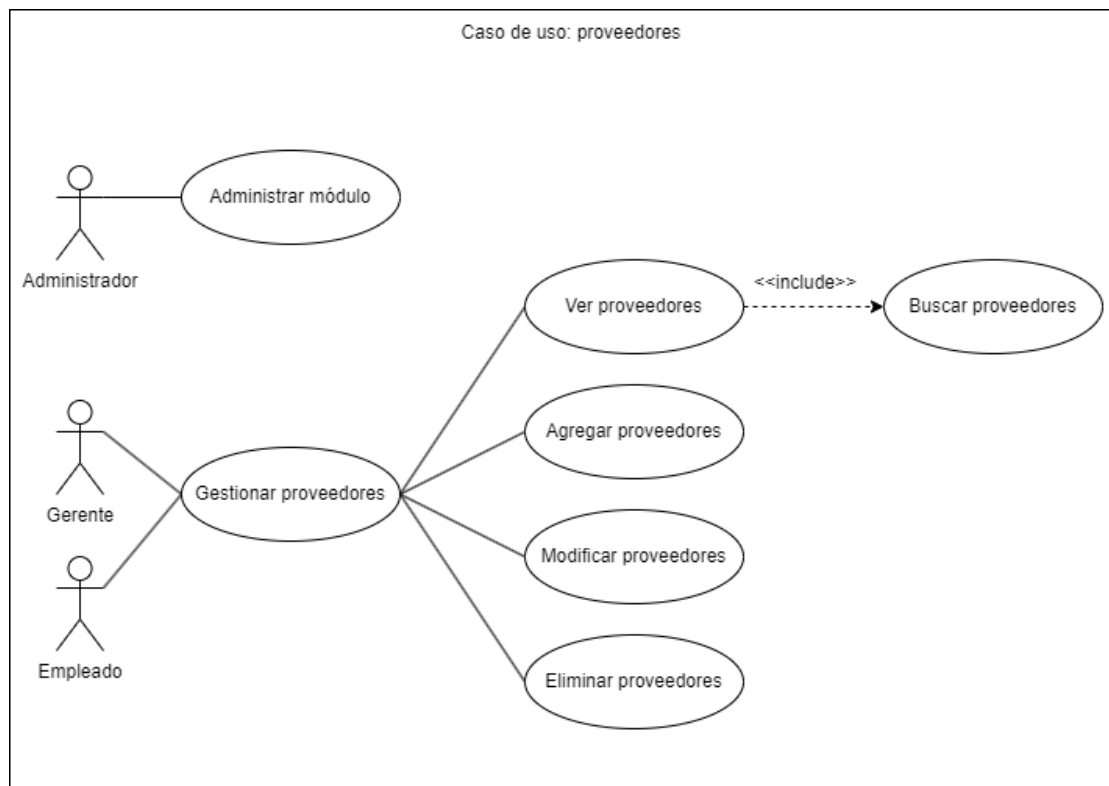
```
18 ● ○ CREATE TABLE Clientes (  
19     id_cliente INT PRIMARY KEY AUTO_INCREMENT,  
20     nombre_cliente VARCHAR(50),  
21     apellido_cliente VARCHAR(50),  
22     usuario_cliente VARCHAR(50),  
23     correo_cliente VARCHAR(100),  
24     telefono_cliente VARCHAR(20),  
25     contraseña_cliente VARCHAR(100),  
26     id_cargo INT,  
27     FOREIGN KEY (id_cargo) REFERENCES Cargo(id_cargo)  
28 );
```

El módulo clientes nos permite gestionar la información de todos los clientes, nos permite visualizar, registrar, editar, actualizar y eliminar clientes.

Figura 23. Módulo clientes



3.4.4 Módulo proveedores:

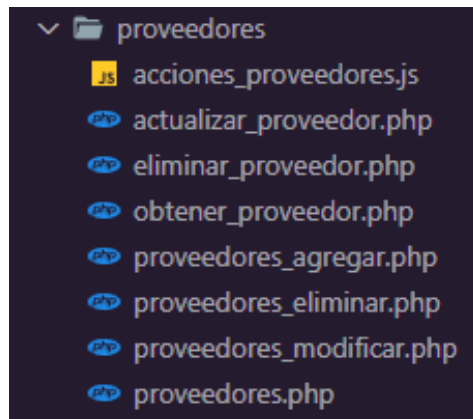
Figura 24. Caso de uso: proveedores

La creación de la tabla proveedores cuenta con un “id_proveedor” que se establece como clave primaria, con opción de “auto_increment” esto con el fin de que el valor de esta columna se incrementará automáticamente para cada nuevo registro. Por otro lado, tenemos una clave foránea “id_cargo” de la tabla “Cargo” para poder identificar que es un usuario de tipo “Proveedor” para hacer consultas con otros módulos con MySQL.

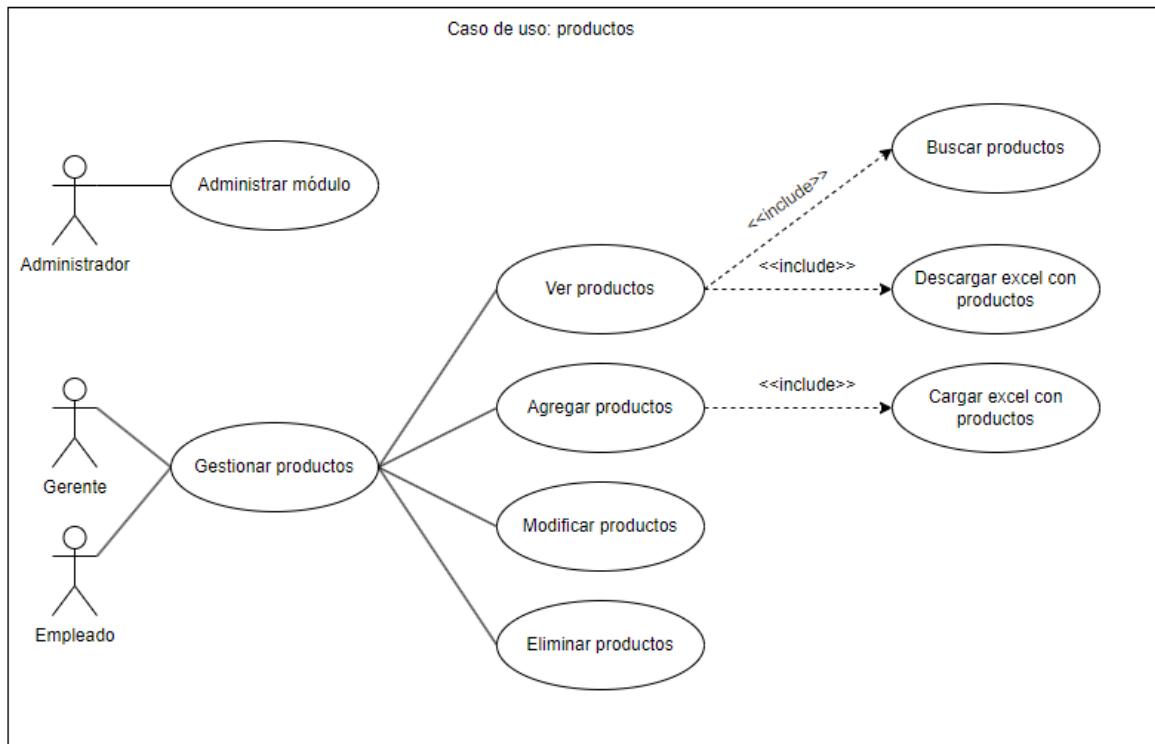
Figura 25. Tabla proveedores SQL

```
30 ● ○ CREATE TABLE Proveedores (  
31     id_proveedor INT PRIMARY KEY AUTO_INCREMENT,  
32     nombre_proveedor VARCHAR(50),  
33     correo_proveedor VARCHAR(100),  
34     razonsocial_proveedor VARCHAR(100),  
35     rfc_proveedor VARCHAR(13),  
36     telefono_proveedor VARCHAR(20),  
37     direccion_proveedor VARCHAR(255),  
38     id_cargo INT,  
39     FOREIGN KEY (id_cargo) REFERENCES Cargo(id_cargo)  
40 );
```

El módulo proveedores nos permite gestionar la información de todos los clientes, nos permite visualizar, registrar, editar, actualizar y eliminar proveedores.

Figura 26. Módulo proveedores

3.4.5 Módulo productos:

Figura 27. Caso de uso: productos

La creación de la tabla productos cuenta con un “id_producto” que se establece como clave primaria, con opción de “auto_increment” esto con el fin de que el valor de esta columna se incrementará automáticamente para cada nuevo registro. Los requerimientos es que se necesita saber el precio compra, precio venta, stock del producto y su fecha de caducidad para tener un mayor control y evitar que lleguen a caducar los productos.

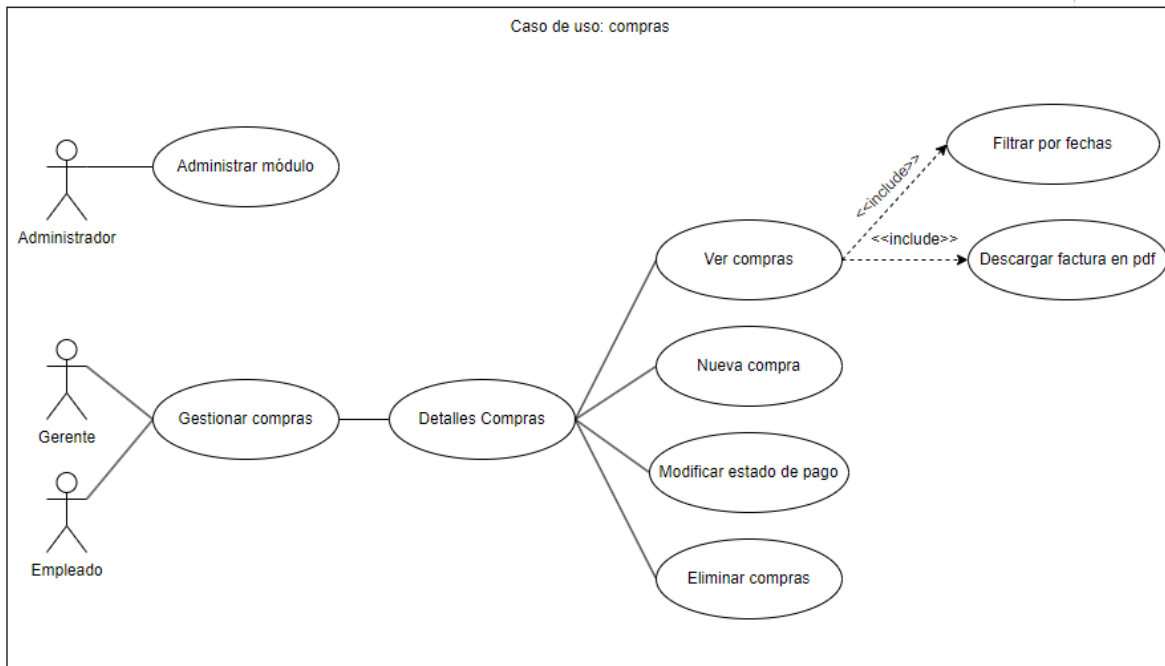
Figura 28. Tabla productos SQL

```

42 ● ○ CREATE TABLE Productos (
43     id_producto INT PRIMARY KEY AUTO_INCREMENT,
44     nombre_producto VARCHAR(100),
45     cantidad_producto INT,
46     precio_compra FLOAT(10, 2),
47     precio_venta FLOAT(10, 2),
48     fecha_caducidad_producto DATE
49 ) ;
  
```

3.4.6 Módulo compras:

Figura 29. Caso de uso: compras



La creación de la tabla compras cuenta con un “id_compra” que se establece como clave primaria, con opción de “auto_increment” esto con el fin de que el valor de esta columna se incrementará automáticamente para cada nuevo registro. Los requerimientos es que compras este vinculado con la tabla “Proveedores” por lo cual tiene llave foránea “id_proveedor” para poder saber a qué proveedor se le compro, la fecha que se realizó la compra, el estado de pago de la compra y el monto total de la compra.

Figura 30. Tabla compras SQL

```

70 • CREATE TABLE Compras (
71     id_compra INT PRIMARY KEY AUTO_INCREMENT,
72     id_proveedor INT,
73     fecha_compra DATE,
74     total_compra FLOAT(10, 2),
75     estado_pago ENUM('Pagado', 'Pendiente', 'Rechazado') DEFAULT 'Pendiente',
76     FOREIGN KEY (id_proveedor) REFERENCES Proveedores(id_proveedor)
77 );
    
```

Se procedió a normalizar la tabla “Compras” mediante la creación de una tabla adicional llamada “Detalles_Compra”. Esta tabla contiene información específica sobre los productos que se compran, descripción del producto, la cantidad comprada y el precio unitario. Esto con el fin de

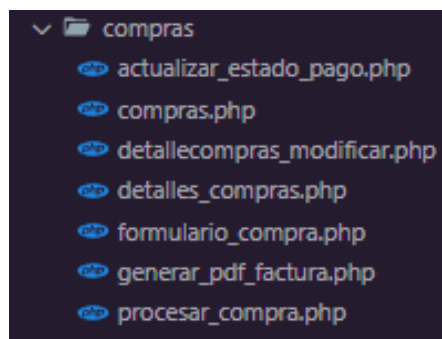
organizar datos, evitar la redundancia y facilitar la manipulación de la información. Cada registro en la tabla “Detalles_Compra” se vincula a su compra correspondiente en la tabla “Compras” a través de una llave foránea, permitiendo así una relación entre los datos.

Figura 31. Tabla detalles compra SQL

```
79 • CREATE TABLE Detalles_Compra (  
80     id_detalle_compra INT PRIMARY KEY AUTO_INCREMENT,  
81     id_compra INT,  
82     descripcion_producto VARCHAR(150),  
83     cantidad_producto INT,  
84     precio_unitario_producto FLOAT(10, 2),  
85     FOREIGN KEY (id_compra) REFERENCES Compras(id_compra)  
86 );
```

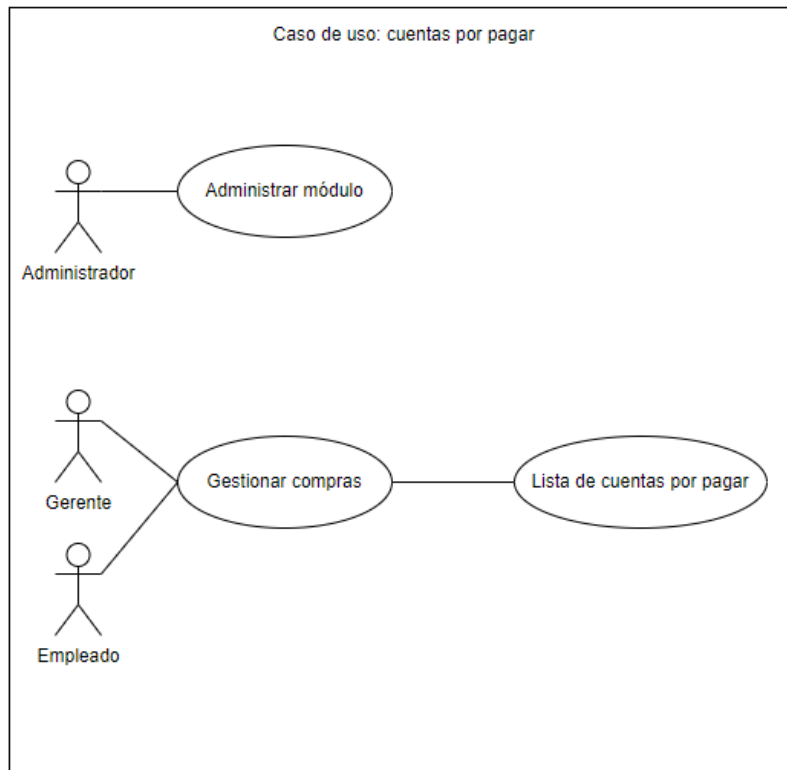
El módulo compras nos permite gestionar la información de todas las compras, nos permite visualizar, registrar, actualizar el estado de pago, eliminar compras y generar en un archivo PDF la factura de la compra.

Figura 32. Módulo compras



3.4.7 Módulo cuentas por pagar:

Figura 33. Caso de uso: cuentas por pagar



La creación de la tabla cuentas por pagar con un “id_cuenta_por_pagar” se establece como llave primaria, con opción de “auto_increment” esto con el fin de que el valor de esta columna se incrementará automáticamente para cada nuevo registro. Los requerimientos es que cuentas por pagar este vinculado con la tabla “Proveedores” y está a “Compras” para poder realizar consultas y saber a qué proveedor se le debe con su nombre, correo, la fecha de compra y el total de la compra.

Figura 34. Tabla cuentas por pagar SQL

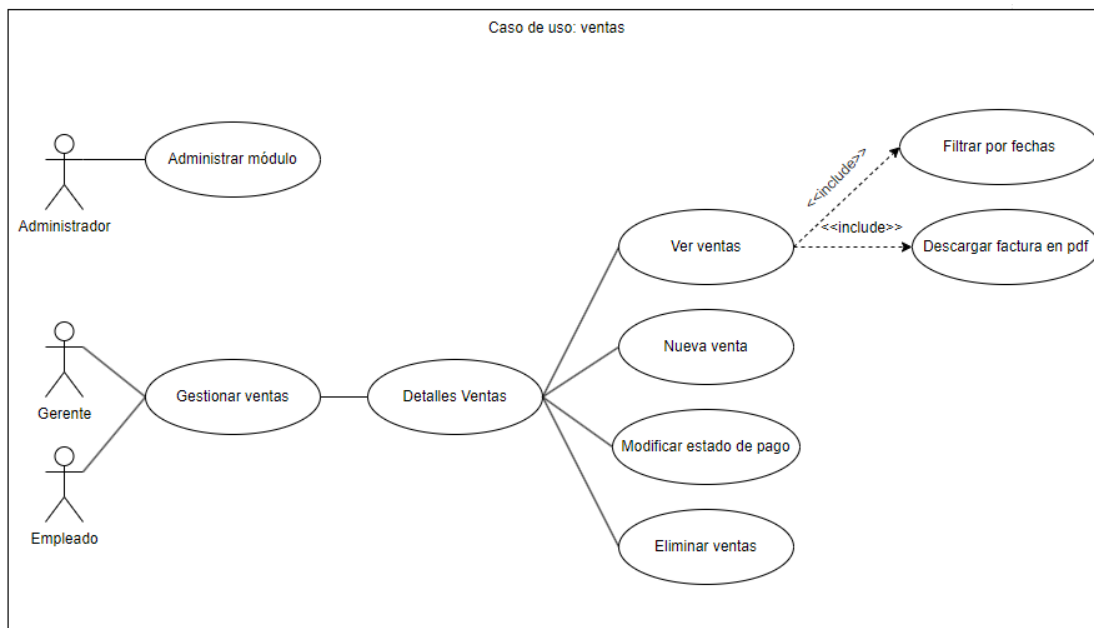
```

88 ● ○ CREATE TABLE Cuentas_por_Pagar (
89     id_cuenta_por_pagar INT PRIMARY KEY AUTO_INCREMENT,
90     id_proveedor INT,
91     monto_cuenta FLOAT(10, 2),
92     fecha_vencimiento_cuenta DATE,
93     pagado BOOLEAN,
94     FOREIGN KEY (id_proveedor) REFERENCES Proveedores(id_proveedor)
95 );

```

Este módulo nos permite visualizar información exacta de todas las cuentas que se deben de pagar, también se hace uso de una consulta sql para el “Inicio” y mostrar globalmente cuánto dinero se debe de pagar.

3.4.8 Módulo ventas:

Figura 35. Caso de uso: ventas

La creación de la tabla ventas cuenta con un “id_venta” que se establece como llave primaria, con opción de “auto_increment” esto con el fin de que el valor de esta columna se incrementará automáticamente para cada nuevo registro. Los requerimientos es que ventas este vinculado con la tabla “Clientes” por lo cual tiene llave foránea “id_cliente” para poder saber a

qué cliente se le realizó la venta, la fecha que se realizó la venta, el estado de pago de la venta y el monto total de la venta.

Figura 36. Tabla ventas SQL

```
51 • CREATE TABLE Ventas (  
52     id_venta INT PRIMARY KEY AUTO_INCREMENT,  
53     id_cliente INT,  
54     fecha_venta DATE,  
55     total_venta FLOAT(10, 2),  
56     estado_pago ENUM('Pagado', 'Pendiente', 'Rechazado') DEFAULT 'Pendiente',  
57     FOREIGN KEY (id_cliente) REFERENCES Clientes(id_cliente)  
58 );
```

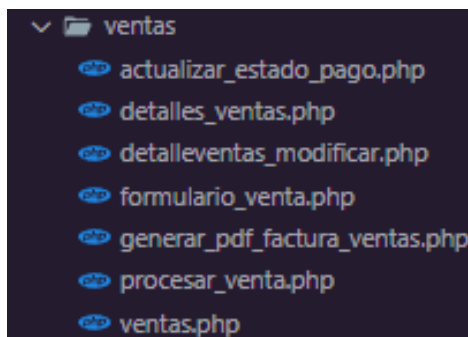
Se procedió a normalizar la tabla “Ventas” mediante la creación de una tabla adicional llamada “Detalles_Venta”. Esta tabla contiene información específica sobre los productos que se compran a través de la tabla “Productos” con una llave foránea “id_producto”, la cantidad comprada y el precio unitario. Esto con el fin de organizar datos, evitar la redundancia y facilitar la manipulación de la información. Cada registro en la tabla “Detalles_Venta” se vincula a su venta correspondiente en la tabla “Ventas” a través de una llave foránea, permitiendo así una relación entre los datos.

Figura 37. Tabla detalles venta SQL

```
60 • CREATE TABLE Detalles_Venta (  
61     id_detalle_venta INT PRIMARY KEY AUTO_INCREMENT,  
62     id_venta INT,  
63     id_producto INT,  
64     cantidad_producto INT,  
65     precio_unitario_producto FLOAT(10, 2),  
66     FOREIGN KEY (id_venta) REFERENCES Ventas(id_venta),  
67     FOREIGN KEY (id_producto) REFERENCES Productos(id_producto)  
68 );
```

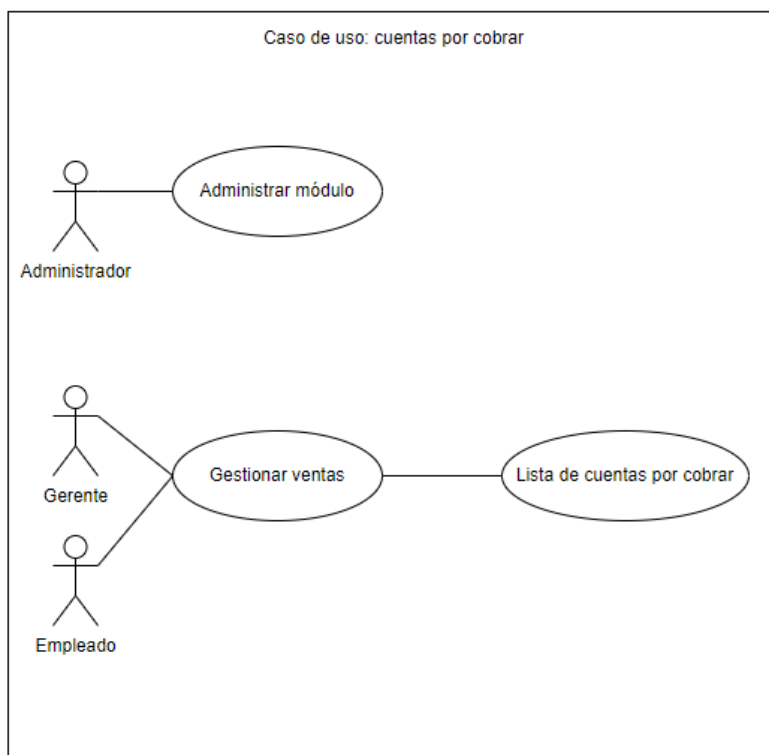
El módulo ventas nos permite gestionar la información de todas las ventas, nos permite visualizar, registrar, actualizar el estado de pago, eliminar ventas y generar en un archivo PDF la factura de la venta.

Figura 38. Módulo ventas



3.4.9 Módulo cuentas por cobrar:

Figura 39. Caso de uso: cuentas por cobrar



La creación de la tabla cuentas por cobrar con un “id_cuenta_por_cobrar” se establece como llave primaria, con opción de “auto_increment” esto con el fin de que el valor de esta columna se incrementará automáticamente para cada nuevo registro. Los requerimientos es que cuentas por cobrar este vinculado con la tabla “Clientes” y está a “Ventas” para poder realizar consultas y saber a qué cliente se le debe con su nombre, apellido, correo, teléfono, la fecha de venta y el total de la venta.

Figura 40. Tabla cuentas por cobrar SQL

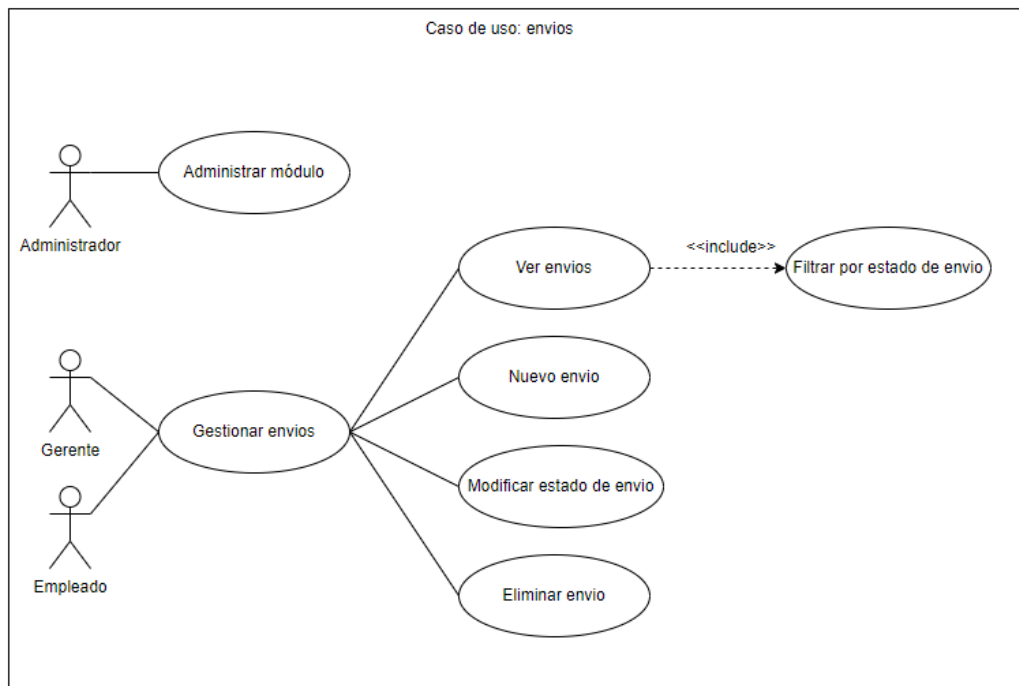
```

97 • CREATE TABLE Cuentas_por_Cobrar (
98     id_cuenta_por_cobrar INT PRIMARY KEY AUTO_INCREMENT,
99     id_cliente INT,
100     monto_cuenta FLOAT(10, 2),
101     fecha_vencimiento_cuenta DATE,
102     pagado BOOLEAN,
103     FOREIGN KEY (id_cliente) REFERENCES Clientes(id_cliente)
104 );
    
```

Este módulo nos permite visualizar la información exacta de todas las cuentas que nos deben de pagar, también se hace uso de una consulta SQL para el “Inicio” y mostrar globalmente cuánto dinero es lo que nos deben.

3.4.10 Módulo envíos:

Figura 41. Caso de uso: envíos



La creación de la tabla envíos cuenta con un “id_envío” que se establece como clave primaria, con opción de “auto_increment” esto con el fin de que el valor de esta columna se incrementará automáticamente para cada nuevo registro. Los requerimientos es que envíos este

vinculado con la tabla “Ventas” y “Clientes” por lo cual tiene llave foránea “id_venta” y “id_cliente” para poder saber que cliente está relacionado con la venta y poder obtener consultas detalladas sobre la información de los productos para así poder asignar una fecha de envío y establecer el estado de envío en el que se encuentra los productos.

Figura 42. Tabla envíos SQL

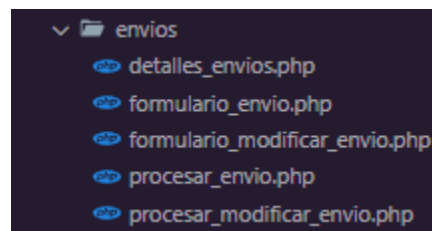
```

135 CREATE TABLE Envios (
136     id_envio INT PRIMARY KEY AUTO_INCREMENT,
137     id_venta INT,
138     id_cliente INT,
139     fecha_envio DATE,
140     estado_envio ENUM('Procesando compra', 'Preparando productos', 'En camino', 'Entregado', 'Intento de entrega fallido', 'Devuelto al remitente'),
141     FOREIGN KEY (id_venta) REFERENCES Ventas(id_venta),
142     FOREIGN KEY (id_cliente) REFERENCES Clientes(id_cliente)
143 );

```

El módulo envíos nos permite gestionar la información de todos los envíos, nos permite obtener las ventas que no se le han asignado envíos para así poder asignarle una fecha de envío, actualizar el estado de envío y filtrar los resultados.

Figura 43. Módulo envíos



3.5 Estructuras de control

En la empresa Quality Std S.A de C.V las estructuras de control juegan un papel esencial en la creación de aplicaciones flexibles. Estas estructuras permiten dirigir la ejecución del código en función de condiciones lógicas, tales como la evaluación de valores de variables o el resultado de expresiones booleanas. Esto incluye la ejecución de bloques de código de manera condicional, la iteración sobre conjunto de datos y la gestión de excepciones que puedan surgir durante la ejecución del programa. Este sistema de información web contiene demasiadas estructuras de control, pero para generalizarlo mostraremos solamente algunas:

3.5.1 Sentencia WHILE

Este código con sentencia WHILE se encarga de generar dinámicamente el HTML necesario para mostrar los resultados de nuestros clientes a través de una consulta a la base de datos y traerlos en forma de tabla.

Figura 44. Sentencia WHILE

```
<?php
while ($fila = mysqli_fetch_assoc($resultado)) {
    echo "<tr>";
    echo "<th scope='row'>" . $fila['id_cliente'] . "</th>";
    echo "<td>" . $fila['nombre_cliente'] . "</td>";
    echo "<td>" . $fila['apellido_cliente'] . "</td>";
    echo "<td>" . $fila['usuario_cliente'] . "</td>";
    echo "<td>" . $fila['correo_cliente'] . "</td>";
    echo "<td>" . ($fila['telefono_cliente'] ? $fila['telefono_cliente'] : 'N/A') . "</td>";
    echo "<td>" . ($fila['descripcion_cargo'] ? $fila['descripcion_cargo'] : 'N/A') . "</td>";
    echo "<td>";
    <button type='button' class='btn btn-warning btn-sm btn-modificar' data-id='\" . $fila['id_cliente'] . "\" data-bs-toggle='modal' data-bs-target='#clientes_modificar_modal'>Modificar</button>
    <button type='button' class='btn btn-danger btn-sm btn-eliminar' data-id='\" . $fila['id_cliente'] . "\" data-bs-toggle='modal' data-bs-target='#clientes_eliminar_modal'>Eliminar</button>
    </td>";
    echo "</tr>";
}
?>
```

3.5.2 Sentencia IF ELSE

Este código con sentencia IF ELSE se encarga de ejecutar una consulta preparada en la base de datos y manejar los resultados y posibles errores asociados con esta operación, en este caso se obtiene el parámetro del id del cliente para poder eliminarlo cuando se presiona el botón correspondiente.

Figura 45. Sentencia IF ELSE

```
if ($stmt) {
    mysqli_stmt_bind_param($stmt, "i", $idCliente);
    mysqli_stmt_execute($stmt);

    if (mysqli_stmt_affected_rows($stmt) > 0) {
        $response = array("success" => true);
    } else {
        $response = array("success" => false, "error" => "No se pudo encontrar el cliente con el ID especificado");
    }

    // Cerrar la declaración
    mysqli_stmt_close($stmt);
} else {
    // Error al preparar la consulta
    $response = array("success" => false, "error" => "Error al preparar la consulta");
}
```

3.5.3 Sentencia FOR

Este código FOR se utiliza para generar una barra de paginación HTML dinámica que permite al usuario navegar a través de múltiples páginas de resultados, esto con el fin de dividir

grandes conjuntos de datos en páginas más pequeñas y manejables para poder mejorar el rendimiento, reducir la carga del servidor y optimizar.

Figura 46. Sentencia FOR

```
<?php for ($i = 1; $i <= $total_paginas; $i++) : ?>
    <li class="page-item <?php echo $i == $pagina_actual ? 'active' : ''; ?>">
        <a class="page-link" href="?pagina=<?php echo $i; ?>"><?php echo $i; ?></a>
    </li>
<?php endfor; ?>
```

3.5.3 Sentencia SWITCH

Este código con sentencia SWITCH nos permite asignar dinámicamente un estilo al texto según el estado de pago que tenga asignado nuestros detalles en las compras y esto nos permite poder establecer un color verde, amarillo y rojo dependiendo el caso, esto con el fin de que para el usuario identifique los procesos que tiene Quality Std S.A de C.V.

Figura 47. Sentencia SWITCH

```
$estado_pago_class = '';
switch ($fila['estado_pago']) {
    case 'Pagado':
        $estado_pago_class = 'text-success';
        break;
    case 'Rechazado':
        $estado_pago_class = 'text-danger';
        break;
    case 'Pendiente':
        $estado_pago_class = 'text-warning';
        break;
    default:
        $estado_pago_class = '';
}
```

3.6 Trigger (disparador)

En la lógica de negocio de Quality Std S.A de C.V se ocupó triggers que básicamente es un procedimiento almacenado en la base de datos que se activa cuando ocurre un evento específico,

es decir, nos permite ejecutar acciones predefinidas en respuesta a cambios en los datos de la base de datos. A continuación, se explican los triggers implementados:

El trigger “actualizar_total_venta” se activa después de insertar nuevos detalles de venta en la tabla "Detalles_Venta". Su propósito es calcular automáticamente el total de una venta y actualizar el campo correspondiente "total_venta" en la tabla "Ventas". Esto asegura que el total de la venta se mantenga actualizado en todo momento, reflejando con precisión la suma de los precios unitarios de los productos vendidos.

Figura 48. Trigger: actualizar total venta

```
154 DELIMITER //
155 • CREATE TRIGGER actualizar_total_venta
156 AFTER INSERT ON Detalles_Venta
157 FOR EACH ROW
158 BEGIN
159     DECLARE total FLOAT(10, 2);
160
161     SELECT SUM(cantidad_producto * precio_unitario_producto) INTO total
162     FROM Detalles_Venta
163     WHERE id_venta = NEW.id_venta;
164
165
166     UPDATE Ventas SET total_venta = total WHERE id_venta = NEW.id_venta;
167 END //
168 DELIMITER ;
```

El trigger “agregar_cuenta_por_cobrar” se dispara después de insertar nuevas ventas en la tabla "Ventas". Su función es agregar automáticamente una nueva entrada en la tabla "Cuentas_por_Cobrar" si la venta no ha sido pagada completamente en el momento de su registro. Calcula el monto total de la venta y establece una fecha de vencimiento para la cuenta por cobrar, asegurando así un seguimiento efectivo de los pagos pendientes por parte de los clientes.

Figura 49. Trigger: agregar cuenta por cobrar

```

178 DELIMITER //
179 • CREATE TRIGGER agregar_cuenta_por_cobrar
180 AFTER INSERT ON Ventas
181 FOR EACH ROW
182 BEGIN
183     DECLARE total_venta FLOAT(10, 2);
184
185
186     SELECT SUM(precio_unitario_producto * cantidad_producto) INTO total_venta
187     FROM Detalles_Venta
188     WHERE id_venta = NEW.id_venta;
189
190
191     IF NEW.estado_pago <> 'Pagado' THEN
192         INSERT INTO Cuentas_por_Cobrar (id_cliente, monto_cuenta, fecha_vencimiento_cuenta, pagado)
193         VALUES (NEW.id_cliente, total_venta, fecha_venta);
194     END IF;
195 END;
196 //
197 DELIMITER ;

```

El trigger “eliminar_cuenta_por_cobrar” se activa después de actualizar el estado de pago de una venta en la tabla "Ventas". Si la venta se marca como "Pagado", el trigger elimina automáticamente la entrada correspondiente en la tabla "Cuentas_por_Cobrar". Esto garantiza que las cuentas por cobrar se actualicen y eliminen correctamente una vez que se haya completado el pago de la venta.

Figura 50. Trigger: eliminar cuenta por cobrar

```

206 DELIMITER //
207 • CREATE TRIGGER eliminar_cuenta_por_cobrar
208 AFTER UPDATE ON Ventas
209 FOR EACH ROW
210 BEGIN
211     IF NEW.estado_pago = 'Pagado' THEN
212         DELETE FROM Cuentas_por_Cobrar WHERE id_venta = NEW.id_venta;
213     END IF;
214 END //
215 DELIMITER ;

```

El trigger “agregar_cuenta_por_pagar” se dispara después de insertar nuevas compras en la tabla "Compras". Su objetivo es crear automáticamente una nueva entrada en la tabla "Cuentas_por_Pagar" si la compra no ha sido pagada completamente en el momento de su registro. Calcula el monto total de la compra y establece una fecha de vencimiento para la cuenta por pagar, asegurando así un seguimiento efectivo de los pagos pendientes a los proveedores.

Figura 51. Trigger: agregar cuenta por pagar

```

218 DELIMITER //
219 • CREATE TRIGGER agregar_cuenta_por_pagar
220 AFTER INSERT ON Compras
221 FOR EACH ROW
222 BEGIN
223
224 IF NEW.estado_pago <> 'Pagado' THEN
225     INSERT INTO Cuentas_por_Pagar (id_proveedor, monto_cuenta, fecha_vencimiento_cuenta, pagado)
226     VALUES (NEW.id_proveedor, NEW.total_compra, fecha_compra);
227 END IF;
228 END;
229 //
230 DELIMITER ;

```

El trigger “eliminar_cuenta_por_pagar” se activa después de actualizar el estado de pago de una compra en la tabla "Compras". Si la compra se marca como "Pagado", el trigger elimina automáticamente la entrada correspondiente en la tabla "Cuentas_por_Pagar". Esto garantiza que las cuentas por pagar se actualicen y eliminen correctamente una vez que se haya completado el pago a los proveedores.

Figura 52. Trigger: eliminar cuenta por pagar

```

232 DELIMITER //
233 • CREATE TRIGGER eliminar_cuenta_por_pagar
234 AFTER UPDATE ON Compras
235 FOR EACH ROW
236 BEGIN
237 IF NEW.estado_pago = 'Pagado' THEN
238     DELETE FROM Cuentas_por_Pagar WHERE id_compra = NEW.id_compra;
239 END IF;
240 END //
241 DELIMITER ;

```

3.7 Pruebas

Se realizarán pruebas exhaustivas para validar la funcionalidad, seguridad y rendimiento del sistema antes de su implementación en el entorno de producción.

Se utilizarán herramientas de prueba para identificar posibles vulnerabilidades y garantizar la fiabilidad del sistema.

3.8 Documentación

Se elaborarán manuales de usuario detallados para facilitar la comprensión y el uso del sistema por parte del personal de Quality Std S.A de C.V.

Se documentará el código fuente y la arquitectura del sistema para futuras referencias y mantenimiento.

CAPÍTULO IV

RESULTADOS

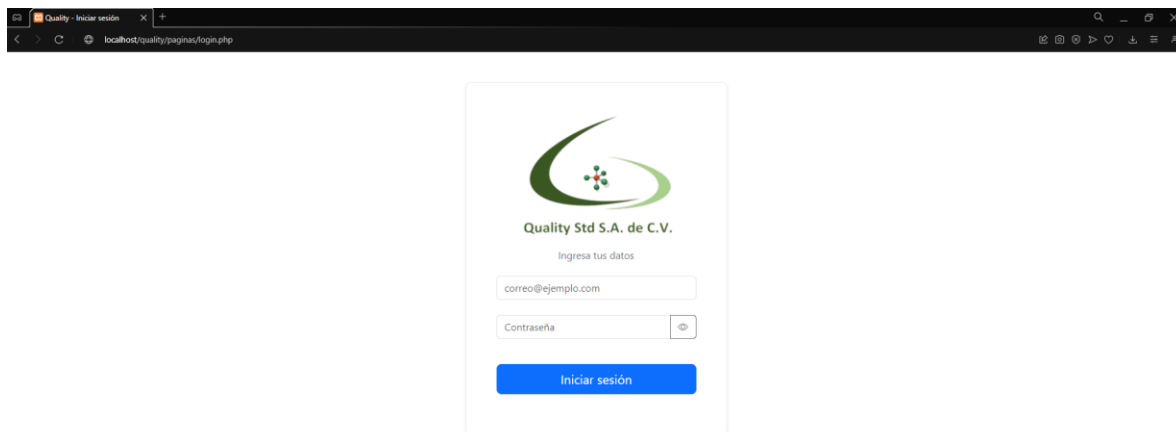
4.1 Login

Este sistema de login proporciona acceso a través de un proceso de autenticación que incluye la introducción de un correo y una contraseña. La interfaz de login se ha diseñado de manera minimalista para cumplir con los requisitos establecidos por la empresa.

El sistema web implementa una estructura de acceso basada en roles, que incluye perfiles para distintos tipos de usuarios: cliente, gerente, empleados, administradores. Esta estructura garantiza que cada usuario tenga las herramientas necesarias para realizar sus funciones de manera eficiente y segura, esto con el fin de proteger la integridad y confidencialidad de los datos del sistema.

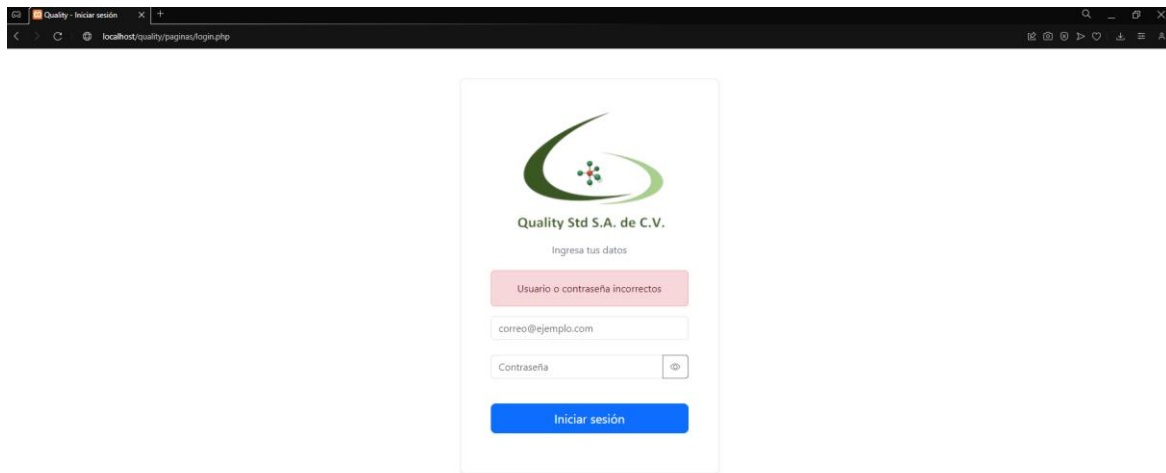
Se implementó que solo los administradores y gerente podían crear las cuentas definiendo el rol que tendrán los usuarios, posteriormente, los usuarios podrían cambiar su contraseña ya que era requerimiento de la empresa.

Figura 53. Interfaz login



En dado caso que el usuario ingrese datos incorrectos en el correo o contraseña, las validaciones indicaran el error y por ende no podrá iniciar sesión hasta que digite sus datos correctamente.

Figura 54. Interfaz login: usuario o contraseña incorrectos

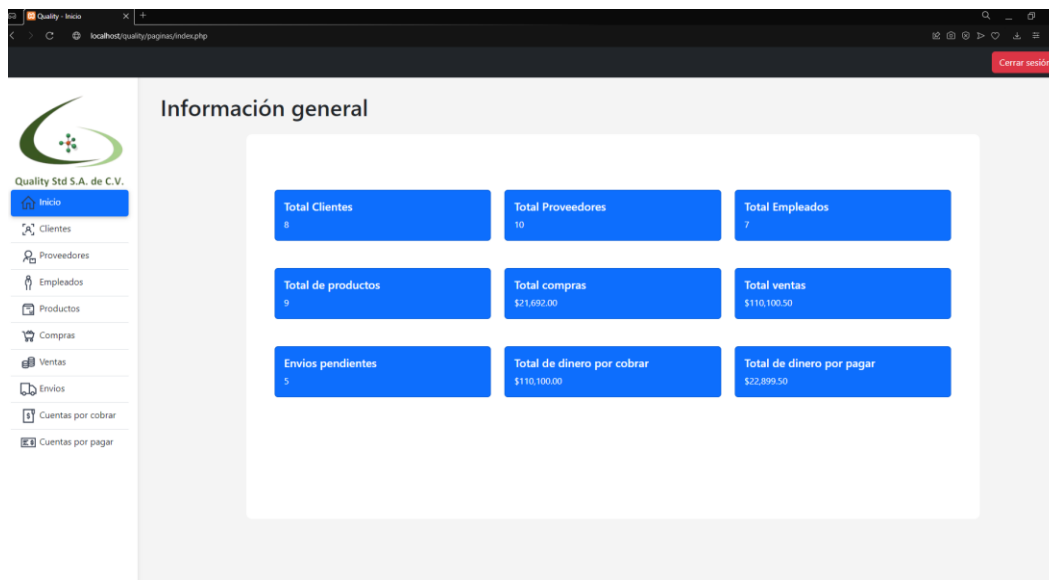


Antes de comenzar a mostrar los módulos es necesario saber que los datos ingresados son aplicados en un ambiente de prueba, por lo que los datos mostrados no son reales.

4.2 Interfaz inicio

En nuestro inicio tenemos información global o general de todos los registros que existen actualmente en la base de datos y van cambiando constantemente conforme se agreguen, modifiquen o eliminen datos.

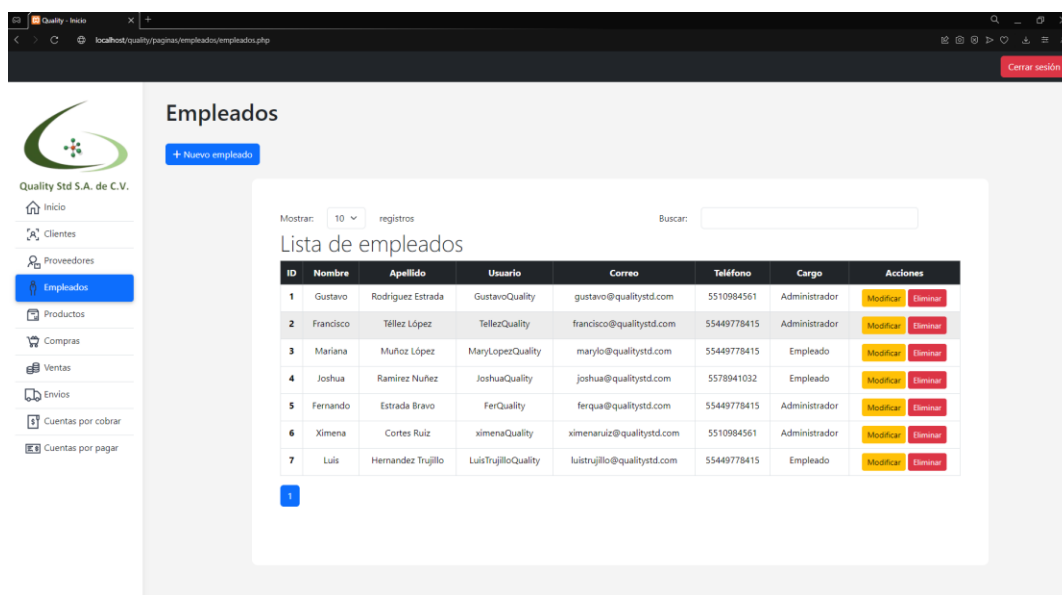
Figura 55. Interfaz inicio



4.3 Empleados

En el módulo empleados es donde solo tiene acceso los usuarios con el cargo de “administrador” o “gerente” y pueden gestionar a los empleados. También la lógica de negocio es que no muestre la “contraseña” de los empleados para tener confidencialidad.

Figura 56. Interfaz empleados



Para agregar nuevo empleado, se abre un “modal” el cual es una ventana emergente que se superpone al contenido principal de una página web, esto con el fin de que no se redireccione a

otra “página” o a otro “enlace”. Los cargo (rol) ya están definidos para que el administrador o gerente pueda seleccionarlo ya que esa información la trae desde la base de datos.

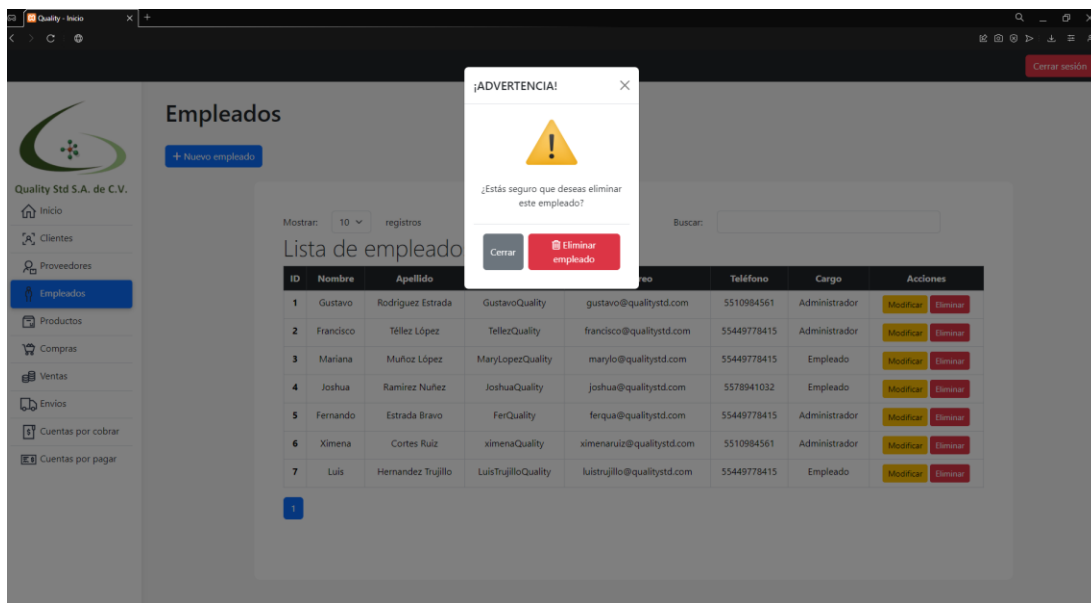
Figura 57. Modal agregar empleado

Para poder modificar o actualizar los empleados a través del id del empleado el botón obtiene la información que tiene en la base de datos para así saber que empleado está siendo modificado y poder guardar cambios si es necesario.

Figura 58. Modal modificar empleado

Para eliminar algún empleado se hizo uso de advertencias para tratar que se borren empleados por accidente, el cual consta de una ventana emergente con un botón para confirmar su eliminación.

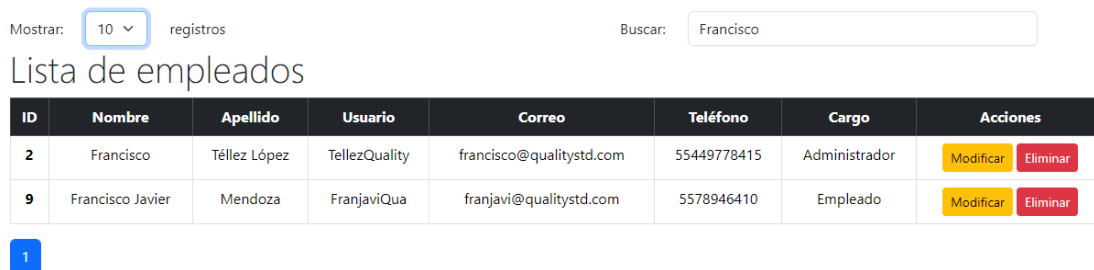
Figura 59. Modal eliminar empleado



Podemos encontrar registros conforme a datos esenciales como lo es: nombre, usuario, correo, cargo. Aquí se puede apreciar que se buscó el empleado “Francisco”, mostrando los resultados (porque podría haber varios empleados con el mismo nombre).

También el sistema cuenta con “paginación” esto nos permite poder ver cierta cantidad de registros en este caso son 10, 15 o 20 registros y es útil cuando se tiene una gran cantidad de datos en la base de datos y se desea mostrarlos de manera eficiente en una página web, dividiéndolos en páginas más pequeñas para una mejor experiencia de usuario

Figura 60. Buscador empleados



Aquí podemos ver los registros insertados desde phpmyadmin que es una herramienta para poder administrar base de datos MySQL basada en páginas web, que son los mismos que tenemos en nuestro sistema.

Figura 61. Base de datos empleados

The screenshot shows the phpMyAdmin interface for the 'empleados' table. The table structure is as follows:

id_empleado	nombre_empleado	apellido_empleado	usuario_empleado	correo_empleado	telefono_empleado	contraseña_empleado	id_cargo
1	Gustavo	Rodríguez Estrada	GustavoQuality	gustavo@qualtystd.com	5510984561	PruebaContraseña123	1
2	Francisco	Téllez López	TéllezQuality	francisco@qualtystd.com	55449778415	PruebaContraseña123	1
3	Mariana	Muñoz López	MaryLopezQuality	marylo@qualtystd.com	55449778415	ContraseñaPrueba123	2
4	Joshua	Ramírez Nuñez	JoshuaQuality	joshua@qualtystd.com	5578941032	ContraseñaPrueba123	2
5	Fernando	Estrada Bravo	FerQuality	ferqua@qualtystd.com	55449778415	ContraseñaPrueba123	1
6	Ximena	Cortés Ruiz	ximenaQuality	ximenaruiz@qualtystd.com	5510984561	ContraseñaPrueba123	1
7	Luis	Hernández Trujillo	LuisTrujilloQuality	lulstrujillo@qualtystd.com	55449778415	ContraseñaPrueba123	2
9	Francisco Javier	Mendoza	FranjaviQua	franjavi@qualtystd.com	5578946410	123123asd	2

4.4 Clientes

En el módulo clientes tenemos todos los usuarios con el cargo de “clientes” para que sea mejor identificado, mostrando de igual forma información necesaria y sin mostrar su contraseña para mantener la confidencialidad.

Figura 62. Interfaz clientes

The screenshot shows the 'Clientes' interface. The list of clients is as follows:

ID	Nombre	Apellido	Usuario	Correo	Teléfono	Cargo	Acciones
1	Laura	Hernandez	Laura48904	laura@gmail.com	5510984561	Cliente	Modificar Eliminar
2	Alejandro	González	alegonz	alejandro@gmail.com	5554325678	Cliente	Modificar Eliminar
3	Sara	Luna	saraluna	sara@gmail.com	5558765432	Cliente	Modificar Eliminar
4	Roberto	Díaz	Roberto2H	roberto@gmail.com	5559876543	Cliente	Modificar Eliminar
5	Martha	Cruz	martha	martha@gmail.com	5551234567	Cliente	Modificar Eliminar
6	Javier	Martínez	javi1111	javier@gmail.com	5558765432	Cliente	Modificar Eliminar
9	Elena	Gutiérrez	elen123	elena@gmail.com	5559876543	Cliente	Modificar Eliminar
12	Mariano	Hernandez	MarianoCliente123	mariano33@gmail.com	5510984561	Cliente	Modificar Eliminar

Para agregar clientes a la base de dato en el formulario del modal en automático esta solamente la opción “cliente” para que no se le asigne otro cargo, por otra parte, en el campo contraseña tenemos un icono para mostrar/ocultar contraseña en dado caso que sea necesario.

Figura 63. Modal agregar cliente

The screenshot shows a web application interface for 'Quality Std S.A. de C.V.'. The main menu on the left includes 'Inicio', 'Clientes', 'Proveedores', 'Empleados', 'Productos', 'Compras', 'Ventas', 'Envíos', 'Cuentas por cobrar', and 'Cuentas por pagar'. The 'Clientes' page is active, displaying a table of clients. A modal titled 'Agregar cliente' is open, allowing the user to add a new client. The modal contains the following fields:

- Nombre:
- Apellido:
- Usuario:
- Contraseña:
- Teléfono:
- Correo:
- Cargo:

At the bottom right of the modal are two buttons: 'Cerrar' and 'Guardar cliente'.

Para poder modificar o actualizar los clientes a través del id del cliente el botón obtiene la información que tiene en la base de datos para así saber que cliente está siendo modificado y poder guardar cambios si es necesario.

Figura 64. Modal modificar cliente

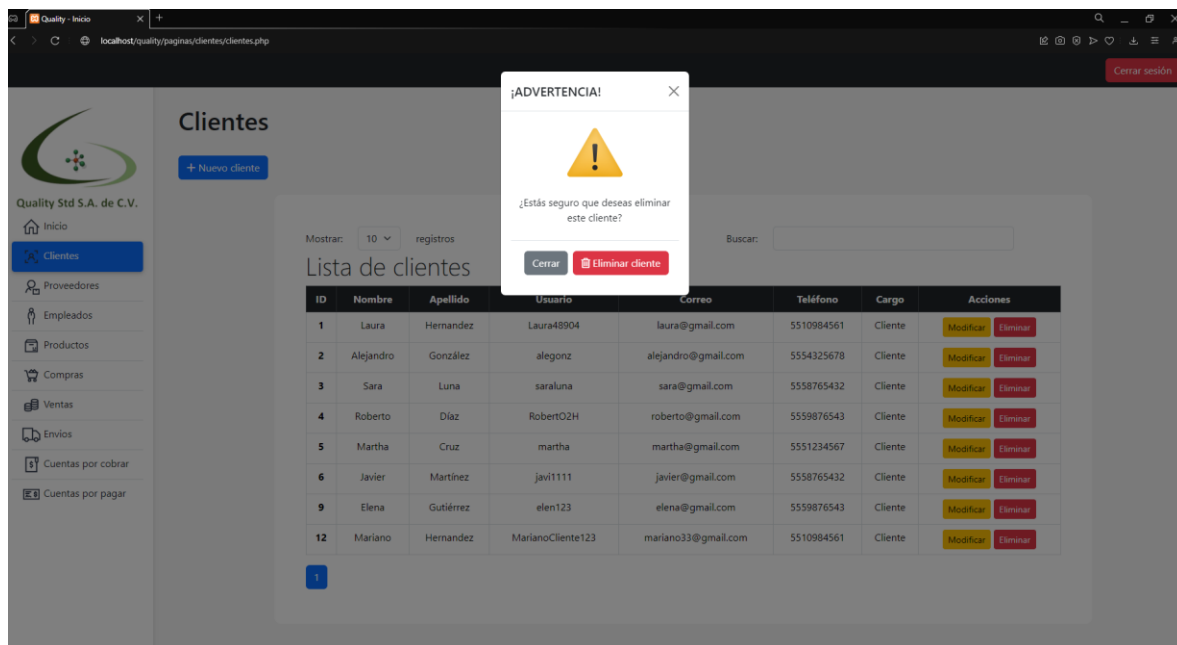
The screenshot shows the same web application interface. The 'Modificar cliente' modal is open, allowing the user to edit an existing client. The modal contains the following fields, pre-filled with data:

- Nombre:
- Apellido:
- Usuario:
- Contraseña:
- Teléfono:
- Correo:
- Cargo:

At the bottom right of the modal are two buttons: 'Cerrar' and 'Guardar cambios'.

Para eliminar algún cliente se hizo uso de advertencias para tratar que se borren clientes por accidente, el cual consta de una ventana emergente con un botón para confirmar su eliminación.

Figura 65. Modal eliminar cliente



Podemos encontrar registros conforme a datos esenciales como lo es: nombre, apellido, usuario, cargo. Aquí se puede apreciar que se buscó el cliente con apellido “Hernandez”, mostrando los resultados (porque podría haber varios clientes con el mismo apellido).

Figura 66. Buscador clientes

Mostrar: 10 registros Buscar: Hernandez

Lista de clientes

ID	Nombre	Apellido	Usuario	Correo	Teléfono	Cargo	Acciones
1	Laura	Hernandez	Laura48904	laura@gmail.com	5510984561	Cliente	Modificar Eliminar
12	Mariano	Hernandez	MarianoCliente123	mariano33@gmail.com	5510984561	Cliente	Modificar Eliminar

1

Desde phpmyadmin podemos ver los datos de los clientes y confirmar que los registros o cambios se han reflejado correctamente en la base de datos.

Figura 67. Base de datos clientes

	id_cliente	nombre_cliente	apellido_cliente	usuario_cliente	correo_cliente	telefono_cliente	contraseña_cliente	id_cargo
☐	1	Laura	Hernandez	Laura48904	laura@gmail.com	5510984561	ASD	3
☐	2	Alejandro	González	alegonz	alejandrog@gmail.com	5554325678	123123	3
☐	3	Sara	Luna	saraluna	sara@gmail.com	5558765432	tesco000	3
☐	4	Roberto	Díaz	Roberto02H	roberto@gmail.com	5559876543	probando123	3
☐	5	Martha	Cruz	martha	martha@gmail.com	5551234567	flotao123	3
☐	6	Javier	Martínez	javi1111	javier@gmail.com	5558765432	francisco030	3
☐	9	Elena	Gutiérrez	elen123	elena@gmail.com	5559876543	123456789	3
☐	12	Mariano	Hernandez	MarianoCliente123	mariano33@gmail.com	5510984561	Mariano123	3

4.5 Proveedores

En el módulo proveedores tenemos todos los usuarios con el cargo “proveedores” para que sea mejor identificado, mostrando de igual forma información necesaria y sin mostrar su contraseña para mantener la confidencialidad.

Figura 68. Interfaz proveedores

Proveedores

+ Nuevo proveedor

Mostrar: 5 registros

Buscar:

Lista de proveedores

ID	Nombre	Correo	Razón Social	RFC	Teléfono	Dirección	Cargo	Acciones
1	Proveedor A	proveedorA@gmail.com	Proveedor A S.A. de C.V.	RFCA123456789	5551234567	Calle 123, Ciudad	Proveedor	Modificar Eliminar
2	Proveedor B	proveedorB@hotmail.com	Proveedor B S.A. de C.V.	RFCB123456789	5559876543	Avenida 456, Ciudad	Proveedor	Modificar Eliminar
3	Proveedor C	proveedorC@gmail.com	Proveedor C S.A. de C.V.	RFCC123456789	5553456789	Calle Principal 789, Ciudad	Proveedor	Modificar Eliminar
4	Proveedor D	proveedorD@outlook.es	Proveedor D S.A. de C.V.	RFCD123456789	5558765432	Calle Secundaria 456, Ciudad	Proveedor	Modificar Eliminar
5	Proveedor E	proveedorE@hotmail.com	Proveedor E S.A. de C.V.	RFCE123456789	5555678901	Avenida Central 123, Ciudad	Proveedor	Modificar Eliminar

Para agregar proveedores a la base de dato en el formulario del modal en automático esta solamente la opción “proveedores” para que no se le asigne otro cargo.

Figura 69. Modal agregar proveedor

Modal form for adding a new provider. Fields include: Nombre, Correo, Razón Social, RFC, Teléfono, Dirección, and Cargo (set to Proveedor). Buttons: Cancelar, Guardar proveedor.

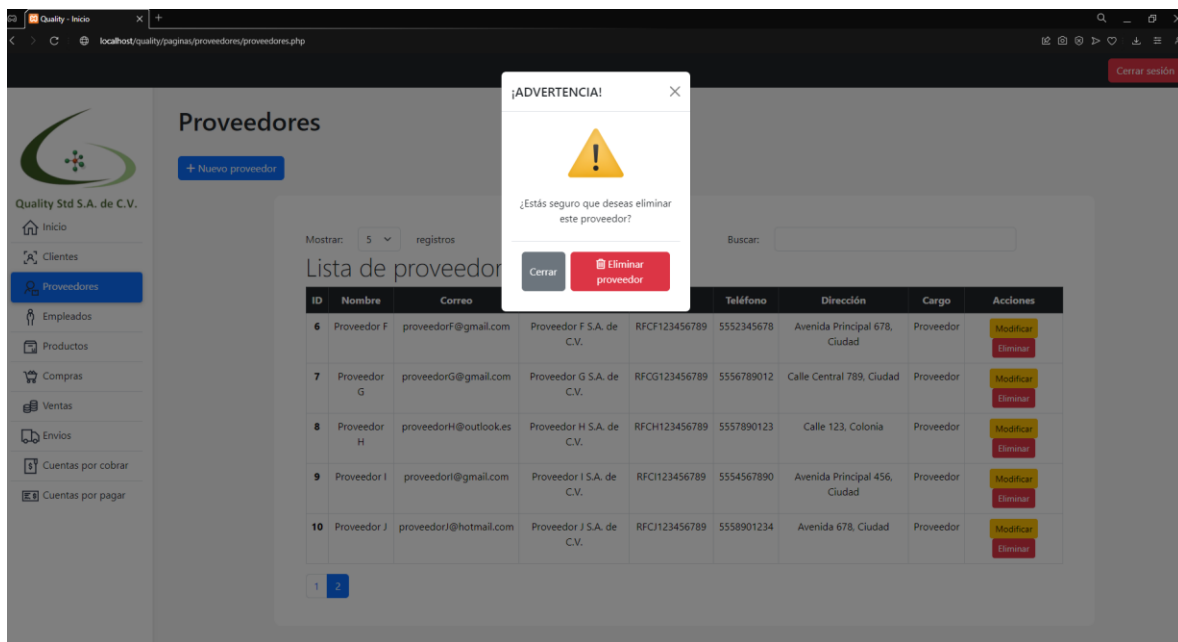
Para poder modificar o actualizar los proveedores a través del id del proveedor el botón obtiene la información que tiene en la base de datos para así saber que proveedor está siendo modificado y poder guardar cambios si es necesario.

Figura 70. Modal modificar proveedor

Modal form for modifying an existing provider. Fields are pre-filled with: Nombre: Proveedor J, Correo: proveedorj@hotmail.com, Razón Social: Proveedor J S.A. de C.V., RFC: RFC123456789, Teléfono: 5558901234, Dirección: Avenida 678, Ciudad. Buttons: Cancelar, Guardar cambios.

Para eliminar algún proveedor se hizo uso de advertencias para tratar que se borren proveedores por accidente, el cual consta de una ventana emergente con un botón para confirmar su eliminación.

Figura 71. Modal eliminar proveedor



Podemos encontrar registros de proveedores conforme a datos esenciales como lo es: nombre, correo, razón social, rfc. Aquí se puede apreciar que se buscó el proveedor con razón social “Proveedor C S.A. de C.V.” mostrando el resultado.

Figura 72. Buscador proveedores



Desde phpmyadmin podemos ver los datos de los proveedores y confirmar que los registros o cambios se han reflejado correctamente en la base de datos.

Figura 73. Base de datos proveedores

Mostrando filas 0 - 9 (total de 10, La consulta tardó 0.0003 segundos)

```
SELECT * FROM `proveedores`
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Opciones extra

	id_proveedor	nombre_proveedor	correo_proveedor	razonsocial_proveedor	rfc_proveedor	telefono_proveedor	direccion_proveedor	id_cargo
☐ Editar ☐ Copiar ☐ Borrar	1	Proveedor A	proveedorA@gmail.com	Proveedor A S.A. de C.V.	RFC A123456789	5551234567	Calle 123, Ciudad	4
☐ Editar ☐ Copiar ☐ Borrar	2	Proveedor B	proveedorB@hotmail.com	Proveedor B S.A. de C.V.	RFC B123456789	5559876543	Avenida 456, Ciudad	4
☐ Editar ☐ Copiar ☐ Borrar	3	Proveedor C	proveedorC@gmail.com	Proveedor C S.A. de C.V.	RFC C123456789	5553456789	Calle Principal 789, Ciudad	4
☐ Editar ☐ Copiar ☐ Borrar	4	Proveedor D	proveedorD@outlook.es	Proveedor D S.A. de C.V.	RFC D123456789	5558765432	Calle Secundaria 456, Ciudad	4
☐ Editar ☐ Copiar ☐ Borrar	5	Proveedor E	proveedorE@hotmail.com	Proveedor E S.A. de C.V.	RFC E123456789	5555678901	Avenida Central 123, Ciudad	4
☐ Editar ☐ Copiar ☐ Borrar	6	Proveedor F	proveedorF@gmail.com	Proveedor F S.A. de C.V.	RFC F123456789	5552345678	Avenida Principal 678, Ciudad	4
☐ Editar ☐ Copiar ☐ Borrar	7	Proveedor G	proveedorG@gmail.com	Proveedor G S.A. de C.V.	RFC G123456789	5556789012	Calle Central 789, Ciudad	4
☐ Editar ☐ Copiar ☐ Borrar	8	Proveedor H	proveedorH@outlook.es	Proveedor H S.A. de C.V.	RFC H123456789	5557890123	Calle 123, Colonia	4
☐ Editar ☐ Copiar ☐ Borrar	9	Proveedor I	proveedorI@gmail.com	Proveedor I S.A. de C.V.	RFC I123456789	5554567890	Avenida Principal 456, Ciudad	4
☐ Editar ☐ Copiar ☐ Borrar	10	Proveedor J	proveedorJ@hotmail.com	Proveedor J S.A. de C.V.	RFC J123456789	5558901234	Avenida 678, Ciudad	4

Seleccionar todo | Para los elementos que están marcados: Editar Copiar Borrar Exportar

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Operaciones sobre los resultados de la consulta

4.6 Productos

En el módulo productos contamos con varias acciones que se pueden realizar, además de poder agregar un nuevo producto tenemos las opciones de cargar una cantidad grande de información para almacenarla en la base de datos o simplemente descargar la lista de productos que tenemos en formato Excel.

Figura 74. Interfaz productos

Productos

+ Nuevo producto Cargar Excel Descargar Excel

Mostrar: 10 registros Buscar:

Lista de productos

ID	Nombre	Cantidad	Precio Compra	Precio Venta	Fecha Caducidad	Acciones
1	Producto 1	100	50.00	100.00	2024-03-01	Modificar Eliminar
2	Producto 2	150	75.00	120.00	2024-03-15	Modificar Eliminar
3	Producto 3	200	80.00	130.00	2024-03-20	Modificar Eliminar
4	Producto 4	120	60.00	110.00	2024-03-10	Modificar Eliminar
5	Producto 5	180	70.00	115.00	2024-03-25	Modificar Eliminar
6	Producto 6	90	45.00	95.00	2024-03-05	Modificar Eliminar
7	Producto 7	160	65.00	105.00	2024-03-17	Modificar Eliminar
8	Producto 8	220	85.00	140.00	2024-03-30	Modificar Eliminar
9	Producto 9	130	55.00	108.00	2024-03-12	Modificar Eliminar

1

Para agregar un producto es necesario dar clic en “Nuevo producto”, se abrirá un modal donde podemos continuar con el registro del producto donde es necesario colocar el nombre, cantidad, precio compra, precio venta y colocar su fecha de caducidad.

Figura 75. Modal agregar producto

The screenshot shows a web application interface for 'Quality Std S.A. de C.V.'. A modal titled 'Agregar producto' is displayed in the center. The modal has the following fields: 'Nombre' (text input), 'Cantidad' (text input), 'Precio de compra' (text input), 'Precio de venta' (text input), and 'Fecha de caducidad' (text input with a placeholder 'dd/mm/aaaa'). At the bottom of the modal are two buttons: 'Cancelar' and 'Guardar producto'. In the background, a table of products is visible with columns for ID, Nombre, Cantidad, Precio de compra, Precio de venta, Fecha de caducidad, and Acciones. The table lists products 1 through 9.

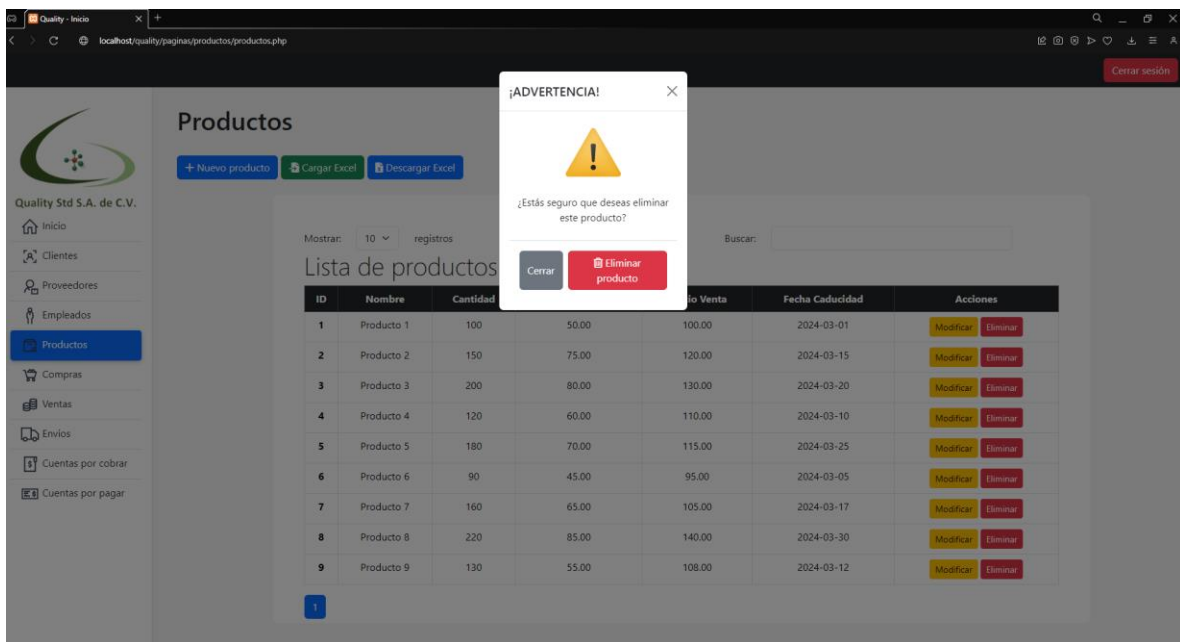
Para poder modificar o actualizar los productos a través del id del producto el botón obtiene la información que tiene en la base de datos para así saber que producto está siendo modificado y poder guardar cambios si es necesario.

Figura 76. Modal modificar producto

The screenshot shows the same web application interface. A modal titled 'Modificar producto' is displayed. The fields are pre-filled with data for 'Producto 6': 'Nombre' (Producto 6), 'Cantidad' (90), 'Precio de compra' (45.00), and 'Precio de venta' (95.00). The 'Fecha de caducidad' field has a calendar widget open, showing the date '01/01/2024'. At the bottom of the modal are two buttons: 'Cancelar' and 'Guardar cambios'. The background table shows the same product list as Figure 75.

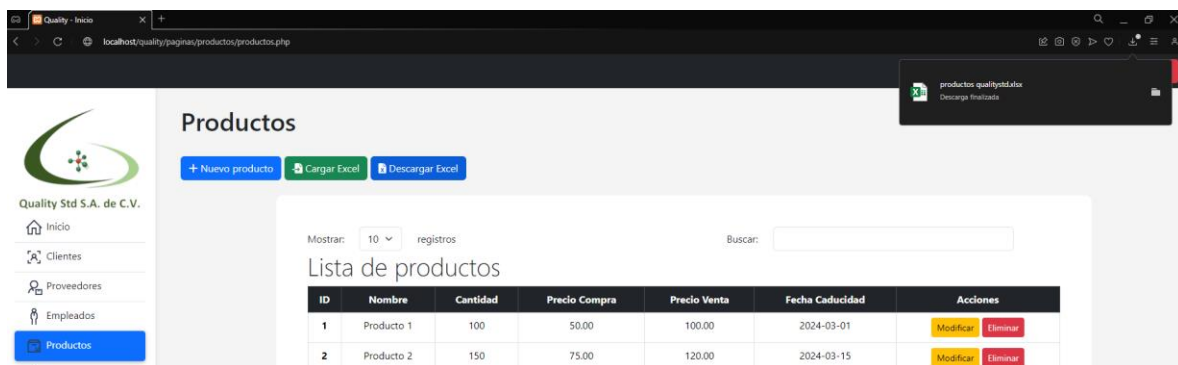
Para eliminar algún producto se hizo uso de advertencias para tratar que se borren proveedores por accidente, el cual consta de una ventana emergente con un botón para confirmar su eliminación.

Figura 77. Modal eliminar producto



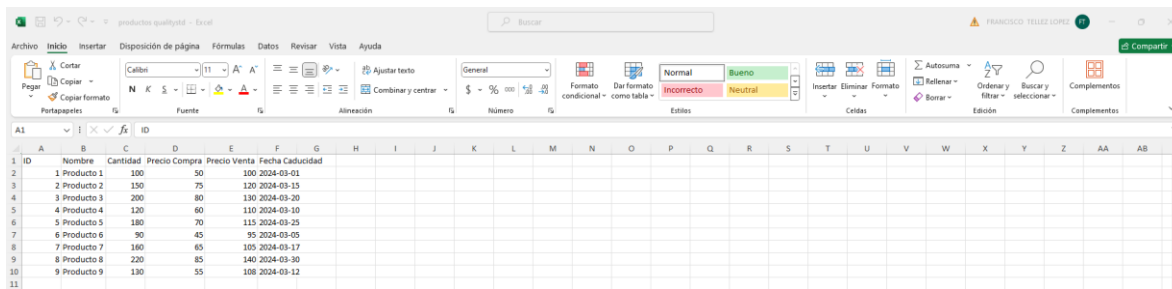
Podemos exportar los productos que tenemos en la base de datos a un Excel en dado caso que se requería para uso administrativo.

Figura 78. Descargar excel con los productos de la base de datos



Permitir descargar un archivo Excel es una forma común y familiar para los usuarios, ya que están familiarizados con este formato y saben cómo utilizarlo para organizar y manipular datos.

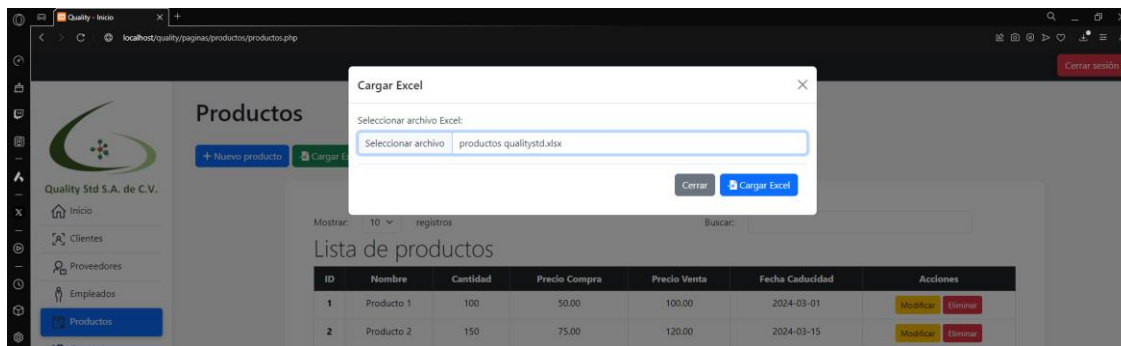
Figura 79. Excel con los productos de la base de datos



ID	Nombre	Cantidad	Precio Compra	Precio Venta	Fecha Caducidad
1	Producto 1	100	50	100	2024-03-01
2	Producto 2	150	75	120	2024-03-15
3	Producto 3	200	80	130	2024-03-30
4	Producto 4	120	60	110	2024-03-10
5	Producto 5	180	70	115	2024-03-25
6	Producto 6	90	45	95	2024-03-05
7	Producto 7	140	65	105	2024-03-17
8	Producto 8	220	85	140	2024-03-30
9	Producto 9	130	55	108	2024-03-12

También podemos cargar un Excel con una cantidad grande de datos, esto nos proporciona una eficiencia y rapidez para no ingresar datos de productos uno por uno, es de suma importancia para Quality Std S.A de C.V ya que tiene grandes cantidades de datos.

Figura 80. Cargar excel con productos



Podemos encontrar registros de los productos conforme a datos esenciales como lo es: nombre del producto, cantidad, precios o fecha de caducidad. Aquí se puede apreciar que se buscó el producto 9, mostrando los resultados.

Figura 81. Buscador productos

Mostrar: 10 registros

Buscar: Producto 9

Lista de productos

ID	Nombre	Cantidad	Precio Compra	Precio Venta	Fecha Caducidad	Acciones
9	Producto 9	130	55.00	108.00	2024-03-12	Modificar Eliminar

1

Podemos ver los datos de los productos y confirmar que los registros o cambios se han reflejado correctamente en la base de datos.

Figura 82. Base de datos productos

id_producto	nombre_producto	cantidad_producto	precio_compra	precio_venta	fecha_caducidad_producto
1	Producto 1	100	50.00	100.00	2024-03-01
2	Producto 2	150	75.00	120.00	2024-03-15
3	Producto 3	200	80.00	130.00	2024-03-20
4	Producto 4	120	60.00	110.00	2024-03-10
5	Producto 5	180	70.00	115.00	2024-03-25
6	Producto 6	90	45.00	95.00	2024-03-05
7	Producto 7	160	65.00	105.00	2024-03-17
8	Producto 8	220	85.00	140.00	2024-03-30
9	Producto 9	130	55.00	108.00	2024-03-12

4.7 Compras

En el módulo compras contamos con varias acciones que se pueden realizar, además de poder agregar un nuevo producto tenemos la opción de factura, donde podemos descargar en un archivo PDF la factura de la compra.

Figura 83. Interfaz compras

ID Compra	Fecha de Compra	ID Proveedor	Estado Pago	Descripción Producto	Cantidad Producto	Precio Unitario Producto	Total Compra	Acciones	Factura
1	2024-01-10	10	Pagado	Acetilcisteína 100g	50	\$128.00	\$8300.00	Modificar estado pago	PDF
1	2024-01-10	10	Pagado	Dimeticona 350 (Silicona Alim.) 250g	20	\$95.00	\$8300.00	Modificar estado pago	PDF
2	2024-01-11	9	Pendiente	Hidroxietilcelulosa	40	\$220.00	\$8800.00	Modificar estado pago	PDF
3	2024-01-12	5	Pendiente	Magnesio Estearato (Vegetal) 5kg	10	\$300.00	\$3000.00	Modificar estado pago	PDF
4	2024-05-13	2	Rechazado	Sodio Acetato Trihidrato Cristal 250g	8	\$199.00	\$1592.00	Modificar estado pago	PDF

Para agregar una compra es necesario dar clic en “Nueva compra”, se abrirá un modal donde podemos continuar con el registro de la compra donde es necesario colocar la fecha de

compra, que proveedor es, la descripción del producto, cantidad, precio unitario. Podemos agregar varios productos y una vez que este todo correcto se procede a dar clic en registrar compra.

Figura 84. Modal agregar nueva compra

The screenshot shows a web application interface for 'Quality Std S.A. de C.V.'. A modal titled 'Nueva Compra' is displayed in the center. The modal has a close button (X) in the top right corner. Inside the modal, there is a date picker for 'Fecha de Compra' with the format 'dd/mm/aaaa'. Below this is a text input field for 'ID Proveedor'. Then, there are three rows of input fields for product details: 'Descripción del Producto', 'Cantidad', and 'Precio Unitario'. At the bottom of the modal, there are two buttons: 'Agregar Producto' (blue) and 'Registrar Compra' (grey). In the background, a table of purchases is visible, showing columns for ID, Fecha de Compra, ID Proveedor, Estado, Descripción Producto, Cantidad Producto, Precio Unitario, Total Compra, Acciones, and Factura.

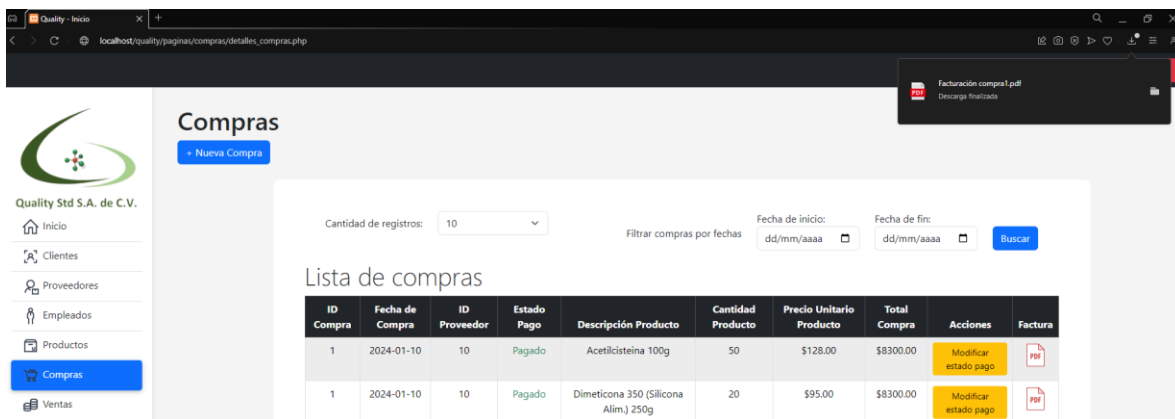
Para poder modificar o actualizar el estado de pago a través del id de la compra, el botón obtiene la información que tiene en la base de datos para así saber que compra está siendo modificado y poder guardar cambios si es necesario.

Figura 85. Modal modificar estado de pago compra

The screenshot shows the same web application interface. A modal titled 'Modificar Estado de Pago' is displayed in the center. The modal has a close button (X) in the top right corner. Inside the modal, there is a dropdown menu for 'Estado de Pago' with the following options: 'Seleccionar estado de pago...', 'Pagado', 'Pendiente', and 'Rechazado'. Below the dropdown, there are two buttons: 'Filtrar compras por fechas' and 'Buscar'. In the background, a table of purchases is visible, showing columns for ID, Fecha de Compra, ID Proveedor, Estado, Descripción Producto, Cantidad Producto, Precio Unitario, Total Compra, Acciones, and Factura.

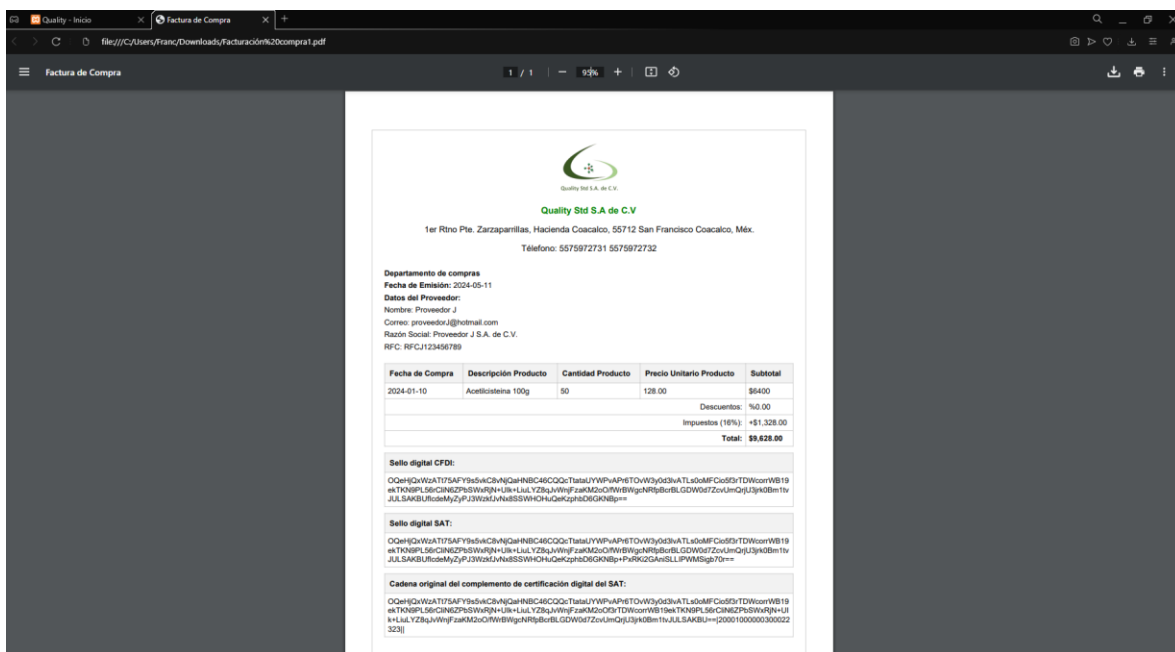
Para generar la factura presionamos el botón de la imagen de PDF en automático se nos descarga el archivo PDF con el id de la compra en este caso “Facturación compra1” ya que fue seleccionado el id compra 1

Figura 86. Descargar factura compra PDF



En la facturación podemos ver datos esenciales de la empresa y del proveedor al cual se le realizó la compra, detalles de la compra como la fecha, descripción del producto, cantidad, precio unitario el subtotal, descuentos, impuestos y total. Además de que como es factura cuenta con algunos requisitos que pide el SAT para poder ser valido.

Figura 87. Factura PDF compra



Podemos encontrar registros de las compras filtrando por días para que sea más fácil identificar las compras que se han realizado en algún periodo del tiempo.

Figura 88. Filtrar compras

Cantidad de registros: 10

Filtrar compras por fechas

Fecha de inicio: 01/01/2024

Fecha de fin: 11/01/2024

Buscar

Lista de compras

ID Compra	Fecha de Compra	ID Proveedor	Estado Pago	Descripción Producto	Cantidad Producto	Precio Unitario Producto	Total Compra	Acciones	Factura
1	2024-01-10	10	Pagado	Acetilcisteína 100g	50	\$128.00	\$8300.00	Modificar estado pago	
1	2024-01-10	10	Pagado	Dimeticona 350 (Silicona Alim.) 250g	20	\$95.00	\$8300.00	Modificar estado pago	
2	2024-01-11	9	Pendiente	Hidroxietilcelulosa	40	\$220.00	\$8800.00	Modificar estado pago	

1

Podemos observar los datos de compras y confirmar que los registros o cambios se han reflejado correctamente en la base de datos.

Figura 89. Base de datos compras

phpMyAdmin

Reciente Favoritas

Nueva

information_schema

mysql

performance_schema

phpmyadmin

qualitybd

Nueva

cargo

clientes

compras

cuentas_por_cobrar

cuentas_por_pagar

detalles_compra

detalles_de_orden_de_compra

detalles_venta

empleados

envios

facturas

ordenes_de_compra

productos

proveedores

seguimiento_de_envios

ventas

Servidor: 127.0.0.1 » Base de datos: qualitybd » Tabla: compras

Examinar Estructura SQL Buscar Insertar Exportar Importar Pr

Mostrando filas 0 - 3 (total de 4, La consulta tardó 0,0006 segundos.)

SELECT * FROM `compras`

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave

Opciones extra

id_compra id_proveedor fecha_compra total_compra estado_pago

☐ Editar Copiar Borrar

1 10 2024-01-10 8300.00 Pagado

☐ Editar Copiar Borrar

2 9 2024-01-11 8800.00 Pendiente

☐ Editar Copiar Borrar

3 5 2024-01-12 3000.00 Pendiente

☐ Editar Copiar Borrar

4 2 2024-05-13 1592.00 Rechazado

Seleccionar todo Para los elementos que están marcados: Editar Copiar Borrar

Mostrar todo Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave

Operaciones sobre los resultados de la consulta

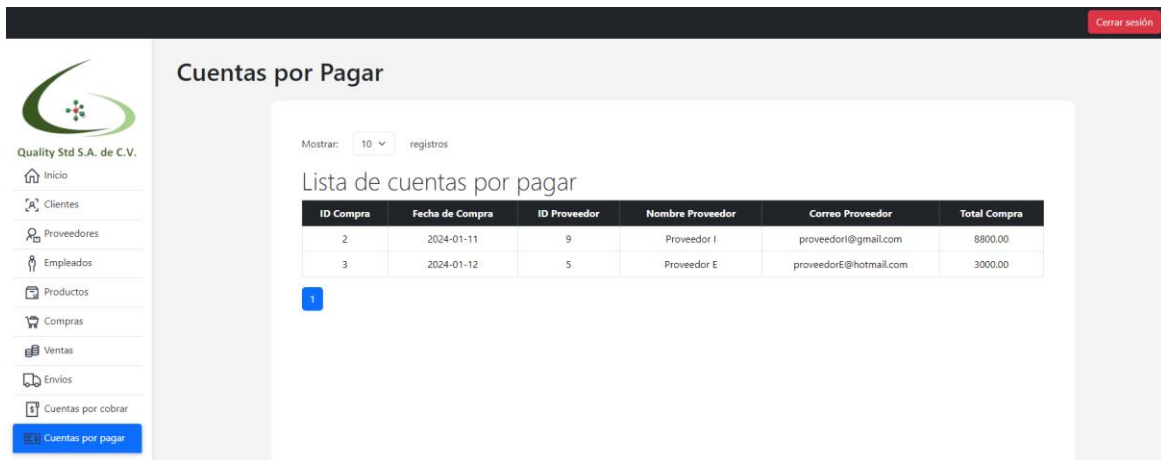
Imprimir Copiar al portapapeles Exportar Mostrar gráfico Crear vista

Guardar esta consulta en favoritos

4.8 Cuentas por pagar

En cuentas por pagar obtendremos la lista de cuentas por pagar, donde son consultas relacionadas de varias tablas y saber qué día realizamos la compra, el nombre del proveedor, su correo y el total que tenemos que pagar. En dado caso de que la compra sea pagada desaparecerá de esa lista en automático ya que solo se muestra lo que se debe.

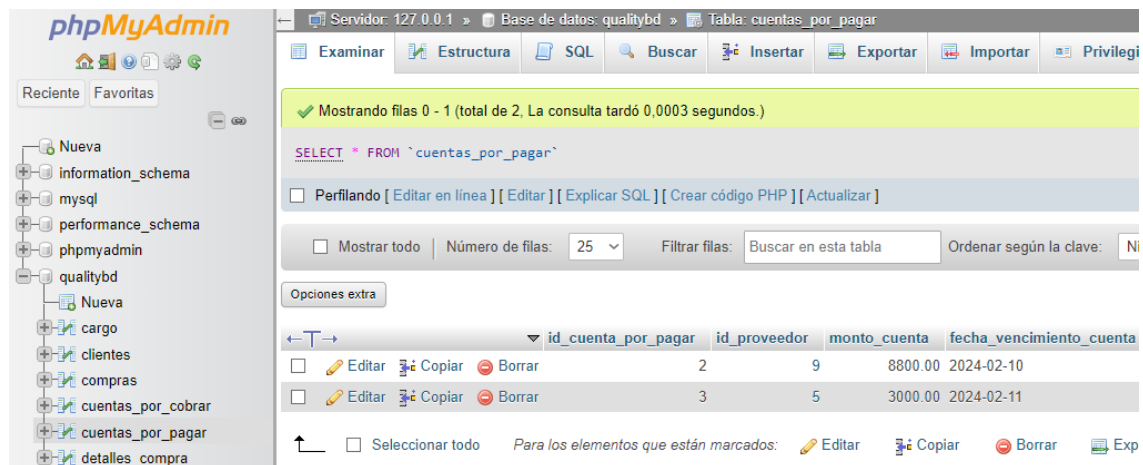
Figura 90. Interfaz cuentas por pagar



ID Compra	Fecha de Compra	ID Proveedor	Nombre Proveedor	Correo Proveedor	Total Compra
2	2024-01-11	9	Proveedor I	proveedorI@gmail.com	8800.00
3	2024-01-12	5	Proveedor E	proveedorE@hotmail.com	3000.00

Podemos observar los datos de cuentas por pagar y confirmar que los registros o cambios se han reflejado correctamente en la base de datos.

Figura 91. Base de datos cuentas por pagar



	id_cuenta_por_pagar	id_proveedor	monto_cuenta	fecha_vencimiento_cuenta
☐ Editar ☐ Copiar ☐ Borrar	2	9	8800.00	2024-02-10
☐ Editar ☐ Copiar ☐ Borrar	3	5	3000.00	2024-02-11

4.9 Ventas

En el módulo ventas contamos con varias acciones que se pueden realizar, además de poder agregar un nuevo producto tenemos la opción de factura, donde podemos descargar en un archivo PDF la factura de la venta, además podemos filtrar la búsqueda a través de las fechas.

Figura 92. Interfaz ventas

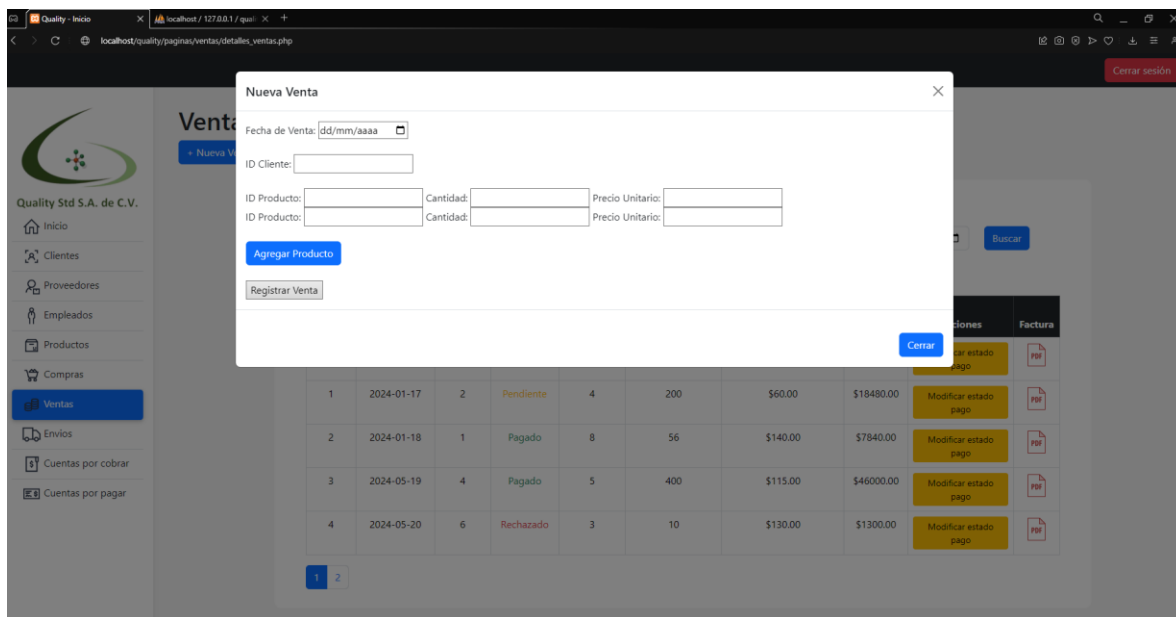
The screenshot displays the 'Ventas' module interface. On the left is a sidebar with the company logo 'Quality Std S.A. de C.V.' and navigation links: Inicio, Clientes, Proveedores, Empleados, Productos, Compras, Ventas (highlighted), Envíos, Cuentas por cobrar, and Cuentas por pagar. The main header shows 'Ventas' with a '+ Nueva Venta' button and a 'Cerrar sesión' button. The central area features a search filter with 'Cantidad de registros' set to 5, 'Filtrar ventas por fechas' with date pickers for 'Fecha de inicio' and 'Fecha de fin', and a 'Buscar' button. Below the filter is a table titled 'Lista de ventas' with the following data:

ID Venta	Fecha de Venta	ID Cliente	Estado Pago	ID Producto	Cantidad Producto	Precio Unitario Producto	Total Venta	Acciones	Factura
1	2024-01-17	2	Pendiente	9	60	\$108.00	\$18480.00	Modificar estado pago	PDF
1	2024-01-17	2	Pendiente	4	200	\$60.00	\$18480.00	Modificar estado pago	PDF
2	2024-01-18	1	Pagado	8	56	\$140.00	\$7840.00	Modificar estado pago	PDF
3	2024-05-19	4	Pagado	5	400	\$115.00	\$46000.00	Modificar estado pago	PDF
4	2024-05-20	6	Rechazado	3	10	\$130.00	\$1300.00	Modificar estado pago	PDF

At the bottom of the table, there are pagination controls showing '1' and '2'.

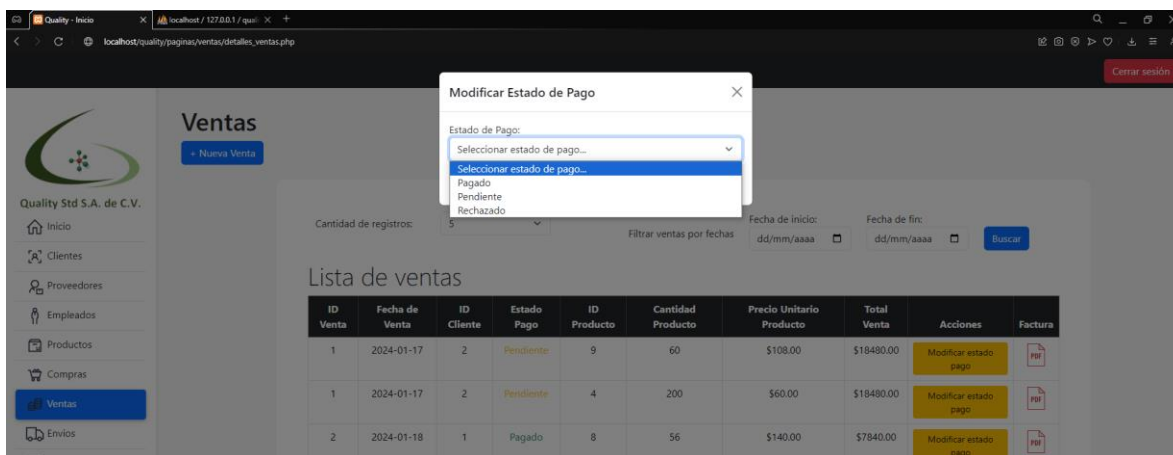
Para agregar una venta es necesario dar clic en “Nueva venta”, se abrirá un modal donde podemos continuar con el registro de la compra donde es necesario colocar la fecha de compra, que cliente es, el id del producto de la base de datos, cantidad, precio unitario. Podemos agregar varios productos y una vez que este todo correcto se procede a dar clic en registrar venta.

Figura 93. Modal agregar nueva venta



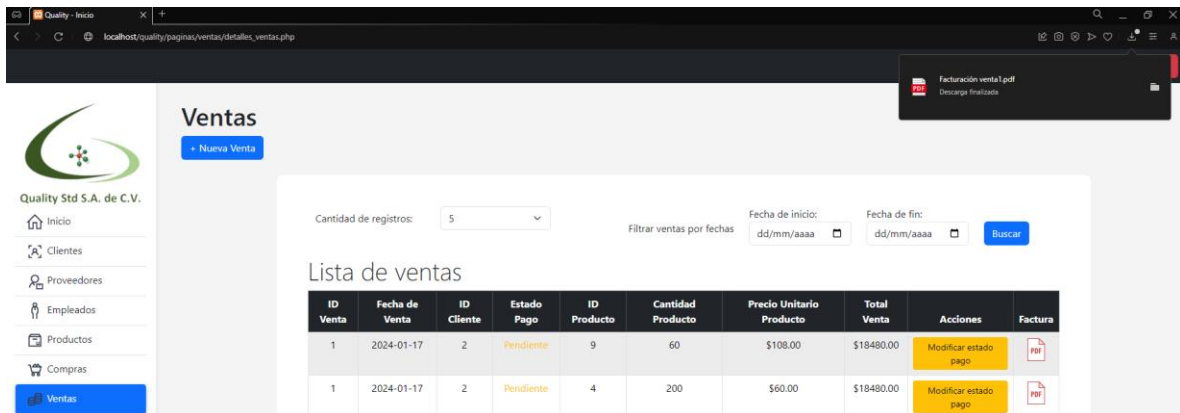
Para poder modificar o actualizar el estado de pago a través del id de la venta, el botón obtiene la información que tiene en la base de datos para así saber que venta está siendo modificada y poder guardar cambios si es necesario.

Figura 94. Modal modificar estado de pago venta



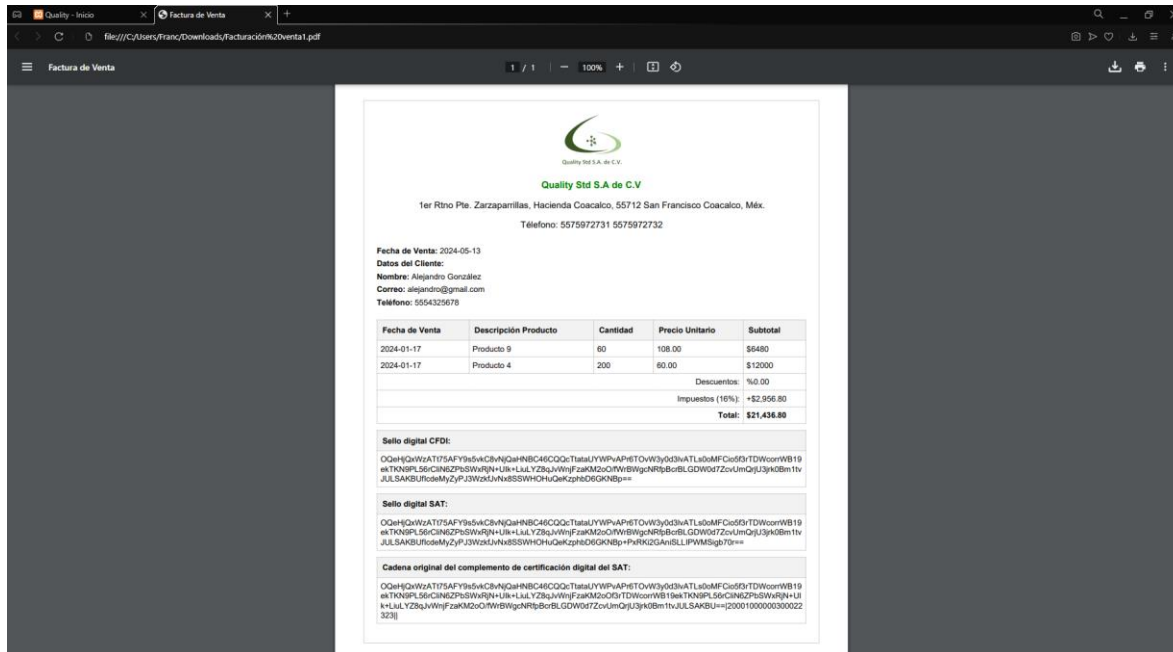
Para generar la factura presionamos el botón de la imagen de PDF en automático se nos descarga el archivo PDF con el id de la compra en este caso “Facturación venta1” ya que fue seleccionado el id venta 1

Figura 95. Descargar factura venta PDF



En la facturación podemos ver datos esenciales de la empresa y del proveedor al cual se le realizó la compra, detalles de la venta como la fecha, descripción del producto, cantidad, precio unitario el subtotal, descuentos, impuestos y total. Además de que como es factura cuenta con algunos requisitos que pide el SAT para poder ser valido.

Figura 96. Factura PDF venta



Podemos observar los datos de ventas y confirmar que los registros o cambios se han reflejado correctamente en la base de datos.

Figura 97. Base de datos ventas

Mostrando filas 0 - 5 (total de 6, La consulta tardó 0,0005 segundos.)

```
SELECT * FROM `ventas`
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Opciones extra

	id_venta	id_cliente	fecha_venta	total_venta	estado_pago
☐ Editar ☐ Copiar ☐ Borrar	1	2	2024-01-17	18480.00	Pendiente
☐ Editar ☐ Copiar ☐ Borrar	2	1	2024-01-18	7840.00	Pagado
☐ Editar ☐ Copiar ☐ Borrar	3	4	2024-05-19	46000.00	Pagado
☐ Editar ☐ Copiar ☐ Borrar	4	6	2024-05-20	1300.00	Rechazado
☐ Editar ☐ Copiar ☐ Borrar	5	12	2024-05-20	35805.00	Pendiente
☐ Editar ☐ Copiar ☐ Borrar	6	1	2024-05-24	675.50	Pendiente

☐ Seleccionar todo | Para los elementos que están marcados: ☐ Editar ☐ Copiar ☐ Borrar ☐ Exportar

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Operaciones sobre los resultados de la consulta

☐ Imprimir ☐ Copiar al portapapeles ☐ Exportar ☐ Mostrar gráfico ☐ Crear vista

4.10 Cuentas por cobrar

En cuentas por cobrar obtendremos la lista de cuentas por cobrar, donde son consultas relacionadas de varias tablas y saber qué día se realizó la venta, el nombre del cliente, el apellido, su correo y el total que tenemos que nos tiene que pagar. En dado caso de que la venta sea pagada desaparecerá de esa lista en automático ya que solo se muestra los datos de los clientes que nos deben dinero.

Figura 98. Interfaz cuentas por cobrar

Cuentas por Cobrar

Mostrar: 10 registros

Lista de cuentas por cobrar

ID Venta	Fecha de Venta	ID Cliente	Nombre Cliente	Apellido Cliente	Teléfono Cliente	Correo Cliente	Total Venta
1	2024-01-17	2	Alejandro	González	5554323678	alejandro@gmail.com	18480.00
5	2024-05-20	12	Mariano	Hernandez	5510984561	mariano33@gmail.com	35805.00
6	2024-05-24	1	Laura	Hernandez	5510984561	laura@gmail.com	675.50

Podemos observar los datos de cuentas por cobrar y confirmar que los registros o cambios se han reflejado correctamente en la base de datos.

Figura 99. Base de datos cuentas por cobrar

Mostrando filas 0 - 3 (total de 4, La consulta tardó 0,0004 segundos.)

`SELECT * FROM `cuentas_por_cobrar``

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Opciones extra

	id_cuenta_por_cobrar	id_cliente	monto_cuenta	fecha_vencimiento_cuenta	pagado
☐ Editar ☐ Copiar ☐ Borrar	2	2	18480.00	2024-02-16	0
☐ Editar ☐ Copiar ☐ Borrar	4	1	7840.00	2024-02-17	0
☐ Editar ☐ Copiar ☐ Borrar	10	12	35805.00	2024-06-19	0
☐ Editar ☐ Copiar ☐ Borrar	12	1	675.00	2024-06-23	0

☐ Seleccionar todo | Para los elementos que están marcados: ☐ Editar ☐ Copiar ☐ Borrar ☐ Exportar

☐ Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Ordenar según la clave: Ninguna

Operaciones sobre los resultados de la consulta

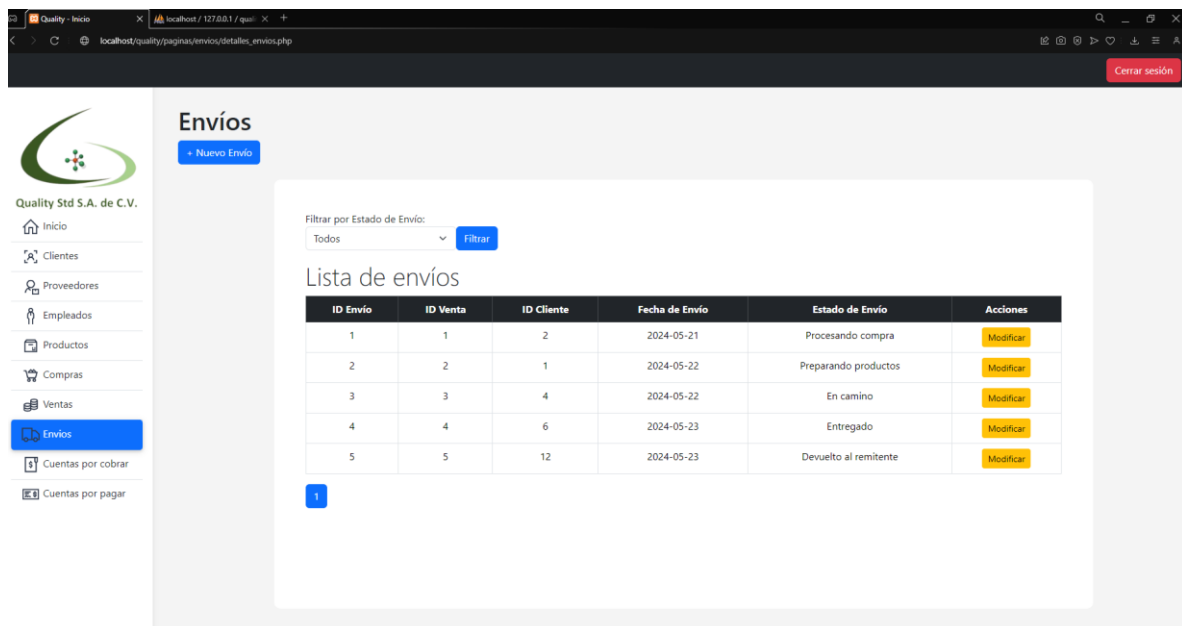
☐ Imprimir ☐ Copiar al portapapeles ☐ Exportar ☐ Mostrar gráfico ☐ Crear vista

☐ Guardar esta consulta en favoritos

4.11 Envíos

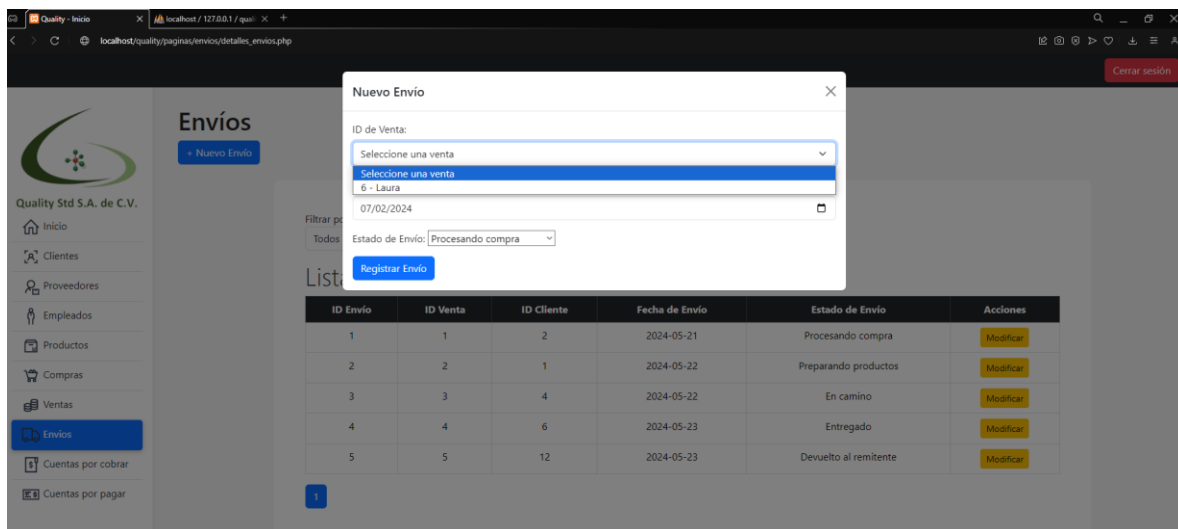
En el módulo envíos podemos ver la lista de los envíos, realizar varias acciones como el agregar un nuevo envío, modificar y se puede filtrar por el estado de envío para saber el proceso de los envíos.

Figura 100. Interfaz envíos



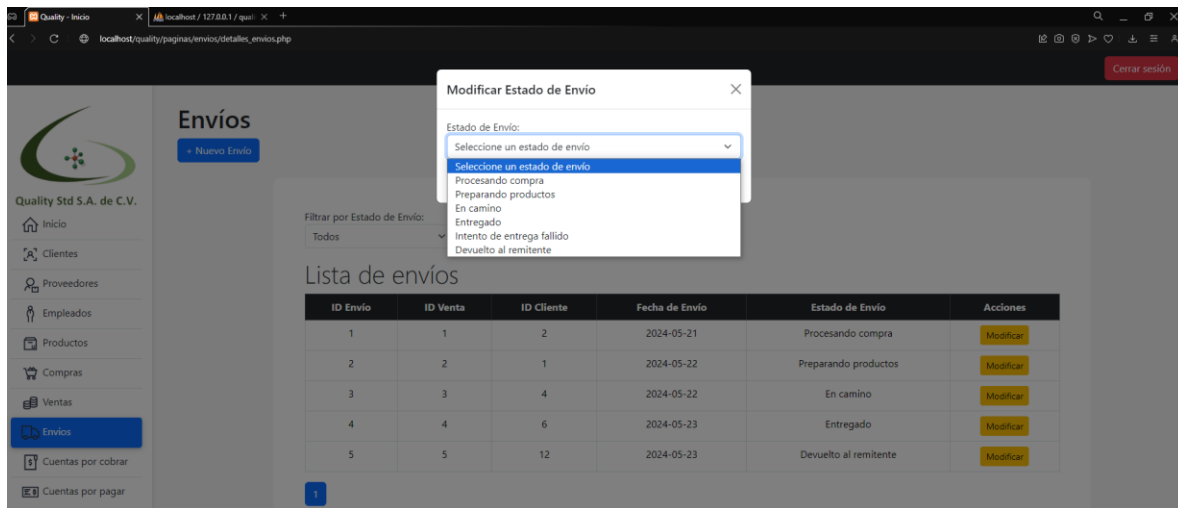
Para agregar un nuevo envío es necesario dar clic en “Nuevo envío” y se nos abrirá un modal donde en ID de venta nos aparecerá una lista de las ventas que no se les ha asignado un envío, se selecciona la fecha de cuándo será enviado y el estado de envío.

Figura 101. Modal agregar nuevo envío



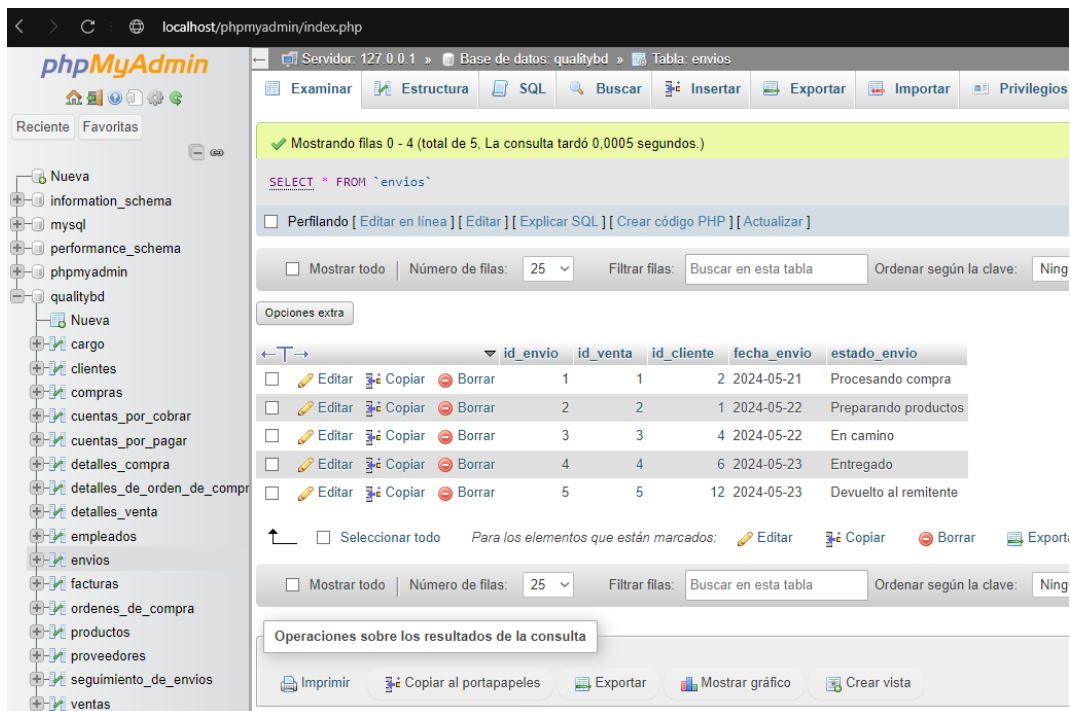
Para poder modificar o actualizar el estado del envío a través del id del envío, el botón obtiene la información que tiene en la base de datos para así saber que envío está siendo modificado y poder guardar cambios si es necesario.

Figura 102. Modal modificar estado de envío



Podemos observar los datos de envíos y confirmar que los registros o cambios se han reflejado correctamente en la base de datos.

Figura 103. Base de datos envíos

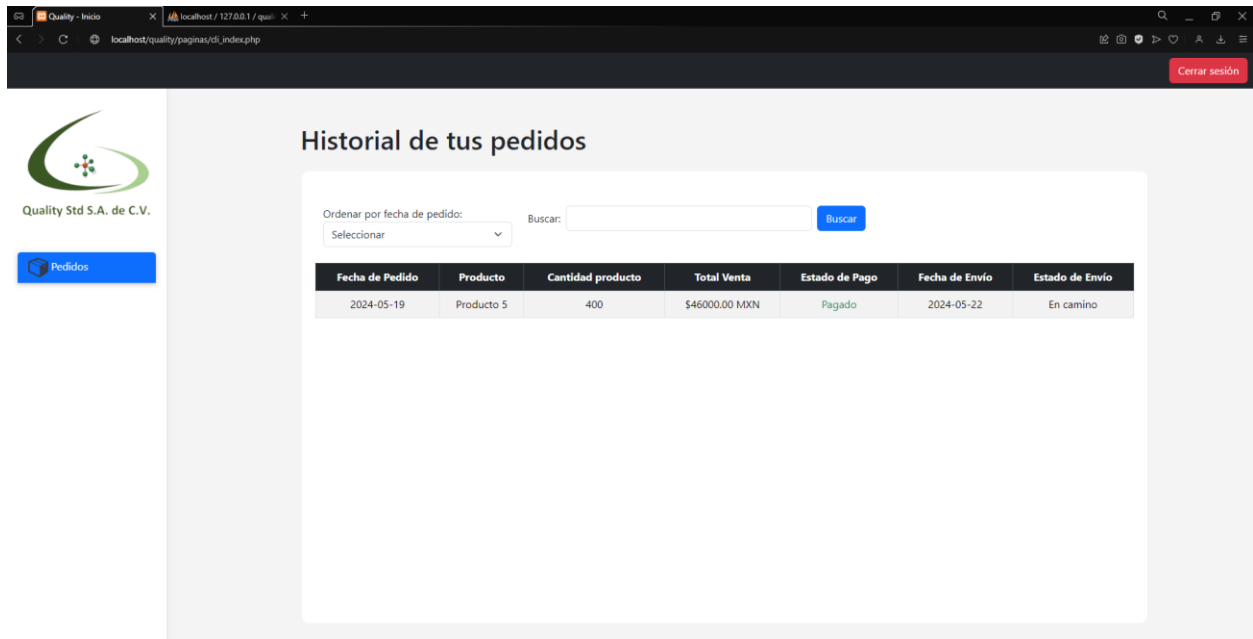


4.12 Pedidos

Cuando los clientes ingresan con sus datos al sistema pasa por validaciones y así poder proporcionarle su interfaz con los datos de sus compras, donde puede consultar los detalles de su

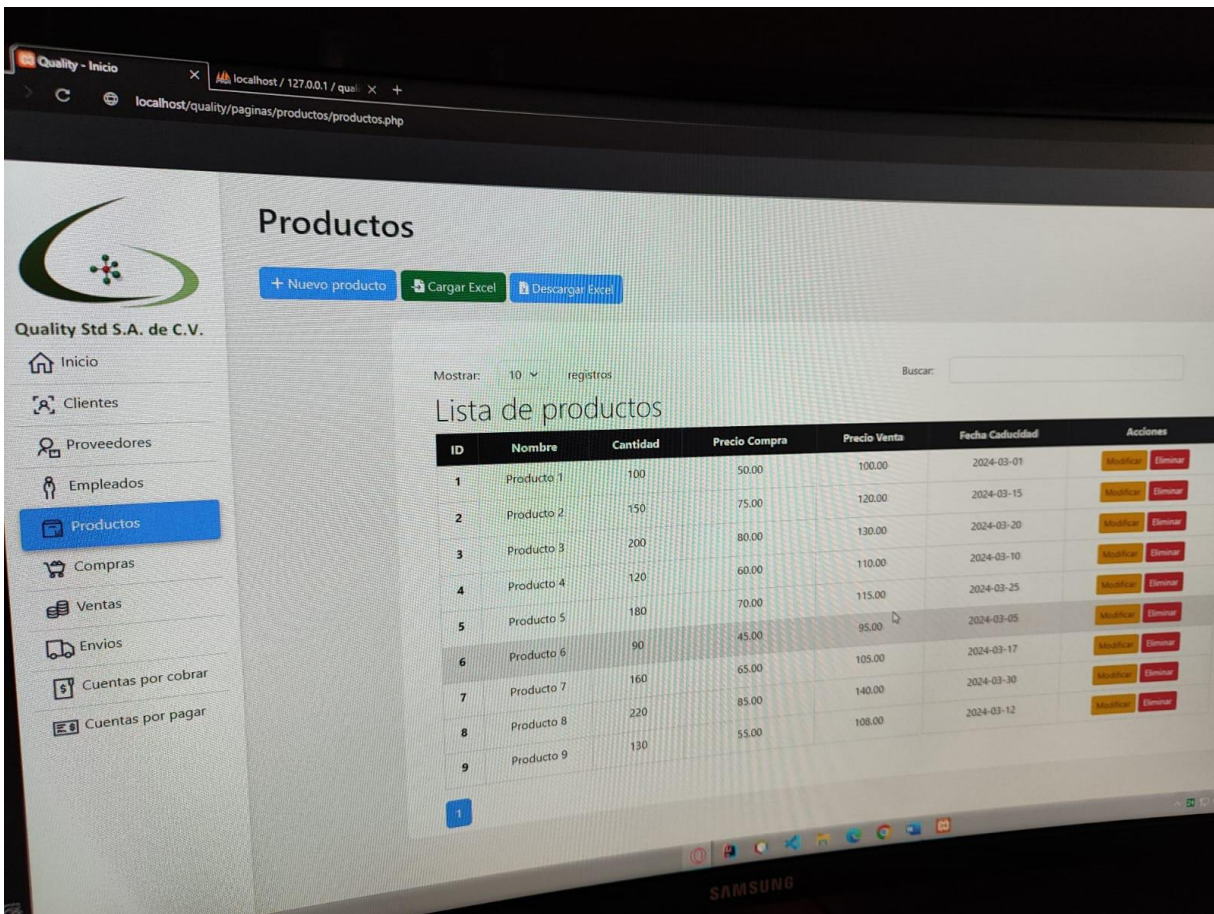
pedido y saber el estado de envío, también puede ordenar sus pedidos por antiguos o más recientes y buscar por nombre del producto. Cuando el cliente ya recibe sus productos su historial de pedidos aun permanecerá para que pueda consultar sus pedidos en cualquier momento.

Figura 104. Interfaz del cliente: pedidos



4.13 Prueba en cuanto a funcionalidad en uso

Figura 105. Prueba de funcionalidad



4.14 Implementación del proyecto en ambiente productivo:

Durante la fase de implementación del sistema de información web para la gestión logística en Quality Std S.A de C.V., se lograron avances significativos que han mejorado notablemente la eficiencia operativa. Sin embargo, es importante reconocer que aún debemos abordar un aspecto fundamental: la infraestructura subyacente que respaldará completamente todas las funcionalidades del sistema.

Hasta ahora, se ha invertido tiempo en el diseño, desarrollo y lanzamiento de la interfaz de usuario, así como en las características principales del sistema. Esto ha conducido a mejoras notables en la gestión de datos y la automatización de procesos, proporcionando una visión alentadora de cómo este sistema puede mejorar los procesos de la empresa.

Sin embargo, es esencial comprender que la infraestructura tecnológica subyacente, que abarca servidores, redes, almacenamiento y seguridad, aún requiere ser establecida y optimizada para respaldar el sistema de manera sólida y confiable. Esta infraestructura es fundamental para garantizar la disponibilidad, escalabilidad y seguridad del sistema.

La implementación de la infraestructura es un proceso que demanda una planificación meticulosa y una inversión cuidadosa en términos de recursos financieros y técnicos. Reconocemos que esta fase es fundamental para completar la implementación exitosa del sistema web y para asegurar que esté preparado para satisfacer las necesidades presentes y futuras de nuestra empresa.

Por lo tanto, como líderes del proyecto y basándonos en los recursos disponibles en la empresa, hemos decidido considerar esta tarea pendiente como una extensión natural del proyecto actual. Esto nos permitirá enfocarnos en garantizar que la infraestructura cumpla con estándares de calidad, confiabilidad y seguridad. Además, esta decisión nos brinda la oportunidad de colaborar estrechamente con expertos en tecnología.

CONCLUSIONES

En el transcurso de este proyecto, se ha abordado el diseño, desarrollo e implementación de un Sistema de Información Web con PHP y MySQL para la gestión de datos logísticos en la empresa Quality Std S.A de C.V. Esta iniciativa surge como respuesta a los desafíos identificados en la gestión de la información dentro de la empresa, específicamente en el ámbito logístico.

Desde la etapa inicial de diagnóstico y análisis de requerimientos hasta la fase de implementación y pruebas, se ha trabajado de manera exhaustiva para asegurar que el sistema desarrollado satisfaga las necesidades específicas de Quality Std S.A de C.V. En este proceso, se ha tomado en cuenta la retroalimentación y las expectativas del personal involucrado en las operaciones logísticas.

El diseño del sistema se centró en mejorar la eficiencia operativa, la precisión en la gestión de datos, la seguridad de la información y la accesibilidad a través de una interfaz intuitiva. Se implementaron funcionalidades clave como el registro y seguimiento de envíos, gestión de inventario, generación de facturas, entre otras, con el objetivo de optimizar los procesos logísticos y mejorar la experiencia tanto para el personal interno como para los clientes de la empresa.

La utilización de tecnologías como PHP y MySQL proporcionó una base sólida para el desarrollo del sistema, permitiendo una integración fluida y eficiente de la lógica de negocio y la gestión de datos. Además, la aplicación de prácticas de seguridad y pruebas garantizó la confiabilidad del sistema, protegiendo la información sensible de la empresa.

En resumen, el sistema de información web desarrollado representa una buena herramienta para Quality Std S.A de C.V., ofreciendo una solución integral para la gestión de datos logísticos que contribuirá significativamente a mejorar la eficiencia operativa, la calidad del servicio y la competitividad en el mercado. Este proyecto no solo cumple con los objetivos planteados inicialmente, sino que también sienta las bases para futuras mejoras y expansiones, adaptándose dinámicamente a las necesidades cambiantes de la empresa y su entorno empresarial.

COMPETENCIAS DESARROLLADAS

Durante el desarrollo de este proyecto para la empresa Quality Std S.A de C.V., he puesto en práctica diversas competencias que han sido fundamentales para su implementación.

Las competencias que más cómodo me sentí y destaque fueron las siguientes:

1. Implementa aplicaciones computacionales para solucionar problemas de diversos contextos, integrando diferentes tecnologías, plataformas o dispositivos.
2. Coordina y participa en equipos multidisciplinarios para la aplicación de soluciones innovadoras en diferentes contextos.
3. Crea y aplica esquemas de bases de datos para garantizar la confiabilidad de los datos en aplicaciones para el tratamiento de información.
4. Diseñar, implementar bases de datos optimizando recursos disponibles, conforme a las normas vigentes de manejo y seguridad de información.
5. Desarrolla y administra software para apoyar la productividad y competitividad de las organizaciones cumpliendo con estándares de calidad

Estas competencias fueron aplicadas de manera efectiva durante mi contribución al proyecto de la empresa Quality Std S.A de C.V., donde trabajé en la optimización, automatización y gestión de datos mediante un sistema web adaptada a las necesidades específicas de la empresa.

Tabla 9. Competencias desarrolladas.

ASIGNATURA	COMPETENCIA DESARROLLADA
Programación orientada a objetos.	• Comprende y aplica la estructura de clases para la creación de objetos y utiliza clases predefinidas para facilitar el desarrollo de aplicaciones • Aplica la programación orientada a objetos para resolver problemas reales y de ingeniería
Taller de base de datos.	• Crea y aplica esquemas de bases de datos para garantizar la confiabilidad de los datos en aplicaciones para el tratamiento de información.
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.
Administración de base de datos.	• Diseñar, implementar bases de datos optimizando recursos disponibles, conforme a las normas vigentes de manejo y seguridad de información.
Ingeniería de software	• Implementa aplicaciones computacionales para solucionar problemas de diversos contextos, integrando diferentes tecnologías, plataformas o dispositivos. • Desarrolla y administra software para apoyar la productividad y competitividad de las organizaciones cumpliendo con estándares de calidad
Taller de investigación II	• Consolida el protocolo para ejecutar la investigación y obtener productos para su exposición, defensa y gestión de su transcendencia.
Gestión de proyectos de software	• Aplica metodologías e instrumentos, para garantizar la gestión adecuada de un proyecto de software.
Ingeniería de requerimientos	• Comprende el modelado de negocios, la importancia de la creación de modelos para entender, comunicar y analizar la complejidad de las organizaciones • Aplica los criterios para determinar la naturaleza de los requerimientos.
Arquitectura y diseño de software	• Fortalecer la capacidad para comprender, documentar, definir, evaluar y evolucionar una arquitectura de software.

BIBLIOGRAFÍA

- AWS. (s.f.). Obtenido de <https://aws.amazon.com/es/what-is/javascript/>
- CreaSystem. (22 de Septiembre de 2022). Obtenido de <https://www.creasystem.net/posts/que-es-un-sistema-web#:~:text=El%20sistema%20web%20o%20tambien,una%20intranet%20desde%20un%20navegador.>
- García, I. J. (30 de Marzo de 2021). *servnet*. Obtenido de <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programacion-de-una-aplicacion-web>
- Gómez, M. C. (27 de Noviembre de 2023). *HubSpot*. Obtenido de <https://blog.hubspot.es/marketing/sistemas-de-informacion-empresas>
- Google Cloud. (s.f.). Obtenido de <https://cloud.google.com/mysql?hl=es>
- HOSTINGER. (11 de Enero de 2023). Obtenido de <https://www.hostinger.mx/tutoriales/que-es-bootstrap>
- HOSTINGER. (27 de Abril de 2023). Obtenido de <https://www.hostinger.mx/tutoriales/que-es-un-servidor-web>
- Ken, A. (3 de Octubre de 2023). *gluo*. Obtenido de <https://www.gluo.mx/blog/backend-que-es-y-para-que-sirve>
- Manz. (s.f.). *Lenguaje CSS*. Obtenido de <https://lenguajecss.com/css/introduccion/que-es-css/>
- MDN Web Docs. (s.f.). Obtenido de <https://developer.mozilla.org/es/docs/Web/HTML>
- ORACLE. (s.f.). Obtenido de [https://www.oracle.com/mx/database/what-is-database/#:~:text=Una%20base%20de%20datos%20es,bases%20de%20datos%20\(DBMS\).](https://www.oracle.com/mx/database/what-is-database/#:~:text=Una%20base%20de%20datos%20es,bases%20de%20datos%20(DBMS).)
- PHP. (s.f.). Obtenido de <https://www.php.net/manual/es/intro-what-is.php>
- Robledano, A. (24 de Septiembre de 2019). *OpenWebinars*. Obtenido de <https://openwebinars.net/blog/que-es-mysql/>
- Santander. (21 de Diciembre de 2020). *Santander Open Academy*. Obtenido de <https://www.santanderopenacademy.com/es/blog/metodologias-desarrollo-software.html>
- UNAM. (2017). Obtenido de https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod_resource/content/1/contenido/index.html
- Unir. (22 de Septiembre de 2022). Obtenido de <https://unirfp.unir.net/revista/ingenieria-y-tecnologia/framework/#:~:text=Un%20framework%20es%20un%20esquema,organización%20y%20desarrollo%20de%20software.>
- Valtx. (s.f.). Obtenido de <https://www.valtx.pe/blog/metodologias-para-el-desarrollo-de-software-que-son-y-para-que-sirven>