

UNIVERSIDAD INTERNACIONAL DE LAS AMÉRICAS

FACULTAD DE INGENIERÍA Y ARQUITECTURA

**TRABAJO FINAL DE GRADUACIÓN PARA OPTAR POR EL GRADO DE
BACHILLER EN INGENIERÍA DE SISTEMAS DE INFORMACIÓN**

Título de la Investigación:

Prototipo funcional para el desarrollo de un sistema de gestión contable y de facturación para la
soda Creston, ubicada en San José

Nombre del estudiante:

Alexis Francisco Mora Mora

Tutor:

Lic. Rolando Sequerira Víctor

Sede Central, San José, Costa Rica

Agosto, 2026

Dedicatoria

Este proyecto se lo dedico a mi familia, que me ha acompañado en todo este proceso y ha sido una guía en cada etapa de mi vida, y mi mayor inspiración para seguir adelante.

A mi mamá, quien ha sido un pilar fundamental en este proceso, me ha ayudado en los momentos más difíciles y me ha dado la fuerza y la confianza que necesitaba para seguir adelante, sin importar los desafíos que se presenten.

A mi papá, quien me ha enseñado el valor de la perseverancia, del trabajo duro y que, mediante su ejemplo y su apoyo, me ayudó a ser un mejor estudiante y persona. Me inspira a mejorar continuamente y siempre buscar la excelencia.

A todas las personas que han sido parte de este proceso, que me han ayudado a levantarme cuando me he caído, que con palabras de apoyo y de inspiración me han dado la fuerza para continuar y que con su apoyo incondicional me han recordado que en todo este tiempo han estado a mi lado.

Agradecimientos

Agradezco a todas las personas que me han ayudado y acompañado en este camino, por su apoyo incondicional, por sus consejos y por estar siempre apoyándome para dar mi mejor versión.

Agradezco a cada uno de los profesores que han sido un ejemplo de lo que es un profesional, que en cada curso me han motivado a crecer y que con dedicación se han preocupado por buscar la excelencia.

A mis seres queridos, familia y amigos, quienes han sido una gran motivación para seguir adelante, que han estado en mis momentos más difíciles y me recordaron que, aunque haya tropiezos y dudas, era capaz de levantarme y seguir trabajando, pues todo esfuerzo tiene su recompensa.

CONTENIDO

Dedicatoria	2
Agradecimientos.....	3
CAPÍTULO I: INTRODUCCIÓN	17
Descripción del Problema	17
Registro contable poco estructurado y ordenado	17
Poco control sobre las cuotas por pagar	17
Dificultad de llevar el control de las facturas.....	17
No existe control en el registro de los ingresos de la empresa.....	18
Mal manejo del control de gastos diarios y operativos	18
Dificultad para llevar el control de las cuentas por cobrar a los clientes	18
Poco control y planificación del presupuesto mensual	18
Objetivos	18
Objetivo General	18
Objetivos Específicos.....	18
Justificación.....	19
Viabilidad Técnica.....	19
Viabilidad Operativa	20
Viabilidad Económica	21
Viabilidad Legal	22
Proyecciones.....	23
Alcance Funcional.....	23
Alcance Metodológico	25
Alcance Tecnológico	27
CAPÍTULO II: MARCO REFERENCIAL.....	29
Prototipo de Software.....	29
Desarrollo del Prototipo con C# y Blazor	30
Lenguaje de Programación C# con Blazor.....	30
Blazor Server y Blazor WebAssembly	30
Desarrollo del Prototipo con SQL Server.....	32
Lenguaje de Base de Datos	32

Gestor de base de datos	33
Operaciones CRUD en SQL	34
Contexto Contable del Prototipo	35
Sistemas Contables.....	35
Contexto de Facturación del Prototipo.....	37
Sistemas de Facturación	37
Estructura del Prototipo.....	38
Patrón de diseño	38
Seguridad del Prototipo	40
Seguridad de la Información	40
CAPÍTULO III: MARCO METODOLÓGICO	43
Enfoques de Investigación	43
Enfoque Cuantitativo.....	43
Enfoque Cualitativo.....	43
Enfoque Mixto.....	44
Enfoque de Investigación Seleccionado.....	44
Tipos de Investigación.....	45
Investigación Descriptiva.....	45
Investigación Exploratoria	45
Investigación Explicativa	45
Tipo de Investigación Seleccionado.....	46
Fuente de Información	46
Fuente de Información Primaria.....	46
Fuente de Información Secundaria.....	46
Fuente de Información Terciaria	47
Variables	47
Variables Conceptuales	47
Variables Operacionales	47
Variables Instrumentales	47
Población.....	50
Muestra.....	50

Instrumentos de Recolección de Datos	50
Entrevista.....	51
Encuesta	51
Proceso para Recolección y Análisis de Datos.....	51
CAPÍTULO IV: ANÁLISIS DE RESULTADOS	53
Entrevista.....	53
Preguntas	53
Cuestionario	54
CAPÍTULO V: PROPUESTA.....	62
Análisis detallado del Software por desarrollar	62
Módulo gestionar cuentas contables	62
Módulo de administrar cuentas por pagar	62
Módulo de gestionar facturación.....	62
Módulo de gestionar ingresos	63
Módulo de gestionar gastos.....	63
Módulo de administrar cuentas por cobrar.....	63
Módulo de generar presupuesto mensual	63
Análisis detallado del hardware requerido	64
Análisis detallado de telecomunicaciones.....	65
Tipo de red.....	65
Uso de internet	65
Puertos y protocolos	65
Equipos especializados.....	65
Herramientas técnicas utilizadas para el desarrollo	65
Lenguaje de programación	65
Entorno de desarrollo integrado	65
Motor de base de datos.....	66
Control de versiones.....	66
Conocimiento básico requerido para la operación del sistema	66
Casos de uso	67
Diseño.....	87

Arquitectura del sistema.....	87
Arquitectura del software	88
Diseño de entradas	90
Diseño físico de la base de datos.....	94
Diccionario de datos.....	95
Diseño de procesos.....	106
Diseño de salidas.....	111
Diagramas UML.....	114
Programación	118
Entradas y salidas	118
Procesos.....	120
Validaciones	121
Módulos.....	122
Pruebas	129
CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES	134
Conclusiones	134
Recomendaciones.....	136
Referencias	138
Apéndices	142
Apéndice A: Guía de Entrevista	142
Apéndice B: Cuestionario	143

TABLAS

Tabla 1	Tabla del hardware.....	20
Tabla 2	Tabla del software.....	20
Tabla 3	Costo del desarrollo del sistema	21
Tabla 4	Costo del Software.....	22
Tabla 5	Descripción de los módulos.....	23
Tabla 6	Software por utilizar	28
Tabla 7	Cuadro de variables	48
Tabla 8	Hardware utilizado	64
Tabla 9	CU-01	67
Tabla 10	CU-02	68
Tabla 11	CU-03.....	71
Tabla 12	CU-04	73
Tabla 13	CU-05	75
Tabla 14	CU-06	76
Tabla 15	CU-07	78
Tabla 16	CU-08	80
Tabla 17	CU-09	82
Tabla 18	CU-10	83
Tabla 19	CU-11.....	84
Tabla 20	CU-12	85
Tabla 21	Tabla Persona.....	95
Tabla 22	Tabla Género.....	95
Tabla 23	Tabla Usuario.....	95
Tabla 24	Tabla Rol.....	96
Tabla 25	Tabla TipoRol	96
Tabla 26	Tabla Cliente.....	96
Tabla 27	Tabla Correo	97
Tabla 28	Tabla Teléfono	97
Tabla 29	Tabla Proveedor	97
Tabla 30	Tabla TipoIdentificación.....	98
Tabla 31	Tabla CuentaPorPagar	98
Tabla 32	Tabla Tercero	98
Tabla 33	Tabla TipoCuenta.....	99
Tabla 34	Tabla Cuenta	99
Tabla 35	Tabla CuentaPorCobrar	99
Tabla 36	AsientoContable	100
Tabla 37	Tabla DetalleAsiento	100
Tabla 38	PresupuestoMensual	100
Tabla 39	DetallePresupuesto	101

Tabla 40	Tabla Producto	101
Tabla 41	Tabla CategoríaProducto	102
Tabla 42	Tabla TipoDePago	102
Tabla 43	Tabla Impuesto.....	102
Tabla 44	Tabla DetalleFactura.....	102
Tabla 45	Tabla Factura	103
Tabla 46	Tabla Ingreso	103
Tabla 47	Tabla Gasto	104
Tabla 48	Tabla Bitácora.....	104
Tabla 49	Tabla TokenRecuperacion.....	105
Tabla 50	Prueba Iniciar Sesión	129
Tabla 51	Prueba Cuenta por pagar.....	130
Tabla 52	Prueba Ingresar datos vacíos	131
Tabla 53	Prueba cuentas iguales	132
Tabla 54	Prueba formato de correo	133

FIGURAS

Figura 1 Eventos Scrum	25
Figura 2 Ciclo de vida del software	26
Figura 3 Arquitectura de tres capas	28
Figura 4 Representación de la conexión utilizando Blazor Server	31
Figura 5 Representación del modelo de alojamiento WebAssembly	31
Figura 6 Patrón de diseño MVC.....	39
Figura 7 Pregunta 1	54
Figura 8 Pregunta 2	55
Figura 9 Pregunta 3	56
Figura 10 Pregunta 4	57
Figura 11 Pregunta 5	58
Figura 12 Pregunta 6	59
Figura 13 Pregunta 7	59
Figura 14 Pregunta 8	60
Figura 15 Pregunta 9	61
Figura 16 Pregunta 10	61
Figura 17 Arquitectura del sistema	88
Figura 18 Arquitectura del software	89
Figura 19 Pantalla de inicio de sesión.....	90
Figura 20 Gestión de Cuentas Contables	91
Figura 21 Formulario Proveedor.....	91
Figura 22 Vista Reporte Balance General.....	92
Figura 23 Vista Usuario	93
Figura 24 Modelo de entidad-relación	94
Figura 25 Diagrama de flujo inicio de sesión	106
Figura 26 Diagrama de flujo Registro Asiento Contable	107
Figura 27 Diagrama de flujo Crear cliente.....	108
Figura 28 Diagrama de flujo Cuentas por pagar	109
Figura 29 Diagrama de flujo Reportes	110
Figura 30 Vista módulo cuentas contables.....	111
Figura 31 Vista Mantenimiento Proveedor	112
Figura 32 Vista Usuarios.....	112
Figura 33 Vista Cuenta por cobrar	113
Figura 34 Vista Facturación	113
Figura 35 Diagrama de Clases	114
Figura 36 Diagrama de secuencia Inicio de Sesión	115
Figura 37 Diagrama de secuencia Facturación	116
Figura 38 Diagrama de secuencia Cuenta por cobrar	116

Figura 39 Diagrama de secuencia Asiento contable	117
Figura 40 Registro Usuario	118
Figura 41 Salida Productos Stock Mínimo	119
Figura 42 Generación asientos contables	120
Figura 43 Validar credenciales	121
Figura 44 Iniciar Sesión	121
Figura 45 Módulo cuentas contables.....	122
Figura 46 Módulo Cuentas por cobrar	123
Figura 47 Módulo Cuentas por pagar.....	124
Figura 48 Módulo Facturación.....	125
Figura 49 Módulo Gastos.....	126
Figura 50 Módulo Ingresos	127
Figura 51 Módulo Presupuesto Mensual.....	128

CAPÍTULO I: INTRODUCCIÓN

Descripción del Problema

La soda Creston es una empresa fundada en el año 2023 por Oscar Delgado Naranjo, quien hoy en día la dirige. Es una soda que ofrece servicios de desayunos, meriendas y almuerzos en la “Creston School”, escuela ubicada en el cantón de Coronado, provincia San José.

Sus servicios están dirigidos a la población estudiantil, sin embargo, también ofrece sus productos en la modalidad de pedidos, por lo que padres de familia, familiares, profesores y demás personas vinculadas al negocio tienen acceso a dichos servicios. En la actualidad la empresa presenta los siguientes problemas:

Registro contable poco estructurado y ordenado

La soda Creston no cuenta con un catálogo contable estructurado y organizado que permita visualizar la clasificación de los activos, pasivos, patrimonio, ingresos y gastos, esto debido a que la gran mayoría de procesos se realizan de forma manual, y el alto volumen de pedidos realizados por la población estudiantil ha representado una gran dificultad para mantener el ritmo de trabajo y catalogar los movimientos.

Poco control sobre las cuotas por pagar

La soda presenta dificultades para llevar un registro de las cuentas por pagar a los proveedores, ya que no existe un control automatizado fiable, por el contrario, es un control que se realiza de forma manual, y esto conlleva a confusiones sobre lo pagado y lo que continúa pendiente de pago; así como atrasos, debido al alto porcentaje de error humano que se puede presentar.

Dificultad de llevar el control de las facturas

La soda, al operar principalmente durante los espacios de recreo de la escuela, debe atender a una gran cantidad de clientes en un tiempo reducido, esto ha provocado que el proceso de facturación y registro de ventas se realice de forma poco organizada, llegando a presentar un descontrol con respecto a los registros de los productos vendidos a los clientes e inconsistencias en sus registros.

No existe control en el registro de los ingresos de la empresa

Diariamente la soda recibe una cierta cantidad de ingresos, provenientes de las ventas que realiza a sus clientes, además de los ingresos que llegan por parte de las distintas cuentas por cobrar a los clientes. Actualmente, estos ingresos no se están registrando de una manera ordenada y controlada.

Mal manejo del control de gastos diarios y operativos

La Soda realiza gastos básicos, como lo son: agua, luz y gas, además de otros tipos de gastos operativos o administrativos, y al no contar con una herramienta que le permita registrar y clasificar estos gastos seguirá presentando dificultades para manejar y controlar los egresos realizados.

Dificultad para llevar el control de las cuentas por cobrar a los clientes

La Soda, al realizar operaciones diariamente y de manera constante, presenta dificultades para llevar el control de las cuentas por cobrar a los clientes; debido a esta situación, el propietario de la Soda presenta pérdidas económicas ya que no cuenta con herramientas que le permitan dar el seguimiento requerido de los pagos pendientes.

Poco control y planificación del presupuesto mensual

La Soda no cuenta con ninguna herramienta o sistema que le permita llevar el control adecuado del presupuesto mensual, esta situación produce dificultades para dar seguimiento a los ingresos y gastos de la empresa, lo que problematiza la capacidad de planificar y administrar correctamente los recursos con que cuenta, lo que implica una problemática añadida al proceso de toma de decisiones.

Objetivos

Objetivo General

Desarrollar un sistema de gestión contable y facturación para la soda Creston, utilizando las herramientas de C# con el framework Blazor para el frontend y lógica de negocio, y SQL Server como gestor de base de datos.

Objetivos Específicos

Analizar los requerimientos del negocio con el objetivo de realizar una adecuada implementación en el sistema, de acuerdo con las necesidades encontradas.

Diseñar el modelo de la base de datos utilizando la herramienta MySQL Workbench y la lógica de la aplicación, de acuerdo con los requerimientos

Desarrollar los módulos del sistema, de acuerdo con el análisis realizado previamente.

Probar los distintos componentes a través de pruebas unitarias e integrales.

Justificación

La investigación se lleva a cabo para proponer una solución tecnológica que mejore la gestión contable y de facturación, ya que la empresa enfrenta a una serie de dificultades de manera diaria, tales como: llevar el control de ingresos y gastos, ejecución del presupuesto mensual, registro de cuotas por pagar y demás. Esta investigación permitirá a la empresa contar con un sistema que reduzca el error humano, cuente con una mejor organización y control de su gestión contable.

A continuación, se detallan diferentes puntos sobre la viabilidad del proyecto, tomando en cuenta aspectos técnicos, operativos, económicos y legales.

Viabilidad Técnica

En cuanto a la viabilidad técnica, la empresa ya cuenta con un espacio físico y equipo de cómputo el cual será necesario para el correcto uso del sistema de gestión contable y de facturación. Se utilizará el lenguaje de programación C#, junto con el framework Blazor Server, de esta manera se permitirá integrar la interfaz gráfica del usuario con la lógica de negocio en una misma arquitectura.

Como entorno de desarrollo integrado (IDE) se utilizará Microsoft Visual Studio, ya que es una herramienta que permite el desarrollo de tecnologías como Blazor y cuenta con funcionalidades para depuración y conexión a base de datos. En este caso, se utilizará la versión de Visual Studio Community, que Microsoft ofrece como un servicio gratuito, sin embargo, existen suscripciones mensuales que proporcionan el uso del IDE acompañado de un plan básico de Azure en \$45 por usuario.

Tabla 1*Tabla del hardware*

Recurso	Especificaciones
Laptop	HP 15 – Fd0006ia
Pantalla	15.6 pulgadas – 1920 x 1080 resolución
Memoria	12 GB DDR4
Disco SSD	512 GB
Sistema Operativo	Windows 11

Fuente: Elaboración propia

La tabla 1 es una representación del equipo con que cuentan los usuarios, para efectos del proyecto. El propietario del negocio no deberá incurrir en gastos de equipo de cómputo, puesto que ya cuentan con el propio.

Tabla 2*Tabla del software*

Producto	Característica
Visual Studio Community	Versión gratuita
C#	Versión gratuita
Blazor	Código abierto
SQL Server Express	Versión gratuita

Fuente: Elaboración propia

Viabilidad Operativa

El sistema será operado por un usuario administrador y otros usuarios regulares; contará con una interfaz gráfica intuitiva y sencilla, en donde cualquier tipo de operador pueda utilizar el sistema sin mucha complicación. Adicionalmente, se contará con la guía de distintas ayudas visuales, para facilitar su uso. El personal no necesitará de un conocimiento técnico avanzado para operar el sistema, sin embargo, posterior al desarrollo del proyecto se impartirá una inducción a los propietarios del negocio.

La implementación del sistema no mostrará afectación al personal de la empresa, se tiene como meta facilitar y automatizar tareas que se realizan de manera manual ya que cambiará la forma en que realizan sus tareas, pues podrán utilizar el sistema como herramienta. Con el fin de que el uso del sistema no represente un problema añadido, se hará la debida inducción a las personas usuarias.

Viabilidad Económica

Para realizar el análisis de la viabilidad económica se utilizará el método de cascada con el fin de resaltar las distintas fases o etapas y desglosar su costo, de esta forma se podrá visualizar tanto el costo por etapa, como el costo total. Para ello, se detalla en la Tabla 4 los costos del desarrollo del sistema.

Tabla 3

Costo del desarrollo del sistema

Fase	Días	Costo total
Análisis de requerimientos	5	¢79 919.8
Diseño del sistema	7	¢111 887.72
Desarrollo	35	¢559 438.6
Pruebas	5	¢79 919.8
Total	52	¢831 165,92

Fuente: Elaboración propia

Los costos descritos en la tabla #3 son a manera de representación y la empresa no deberá recurrir en el costo del desarrollo del sistema, esto por la naturaleza del proyecto. Se toma en cuenta que, de acuerdo con la tabla de salarios mínimos del Ministerio de Trabajo y seguridad social, el salario por día de un programador de computación sin título de bachillerato es de ¢15 983,96, este monto multiplicado por la cantidad de días da como resultado el costo total de cada fase En cuanto al hardware, la empresa dispone de su equipo de cómputo propio, mismo que cumple con los requerimientos mínimos para la ejecución del sistema. En la Tabla 4 se detalla la tabla de costos del software:

Tabla 4
Costo del Software

Producto	Característica	Costo mensual
Visual Studio Community	Se utilizará la versión gratuita, sin embargo, existe una suscripción mensual	\$45
SQL Server Express	Motor de base de datos	Gratuito
C#	Lenguaje de programación de código abierto	Código abierto
Blazor	Framework de código abierto	Código abierto
Total		\$45

Fuente: Elaboración propia

Viabilidad Legal

A continuación, se presentan las leyes por las cuales el desarrollo del prototipo se regirá: Ley 8148 Adición de los artículos 196 BIS, 217 BIS y 229 BIS al Código Penal. Esta ley menciona los artículos 196 bis Violación de comunicaciones electrónicas, con una pena de prisión de seis meses a dos años, el artículo 217 bis Fraude informático con una pena de prisión de uno a diez años y el artículo 229 bis Alteración de datos y sabotaje informático, con una pena de prisión de uno a cuatro años.

El alcance de este prototipo cuenta con el consentimiento del dueño de la empresa, el cual aceptó ser parte del proyecto, se le explicó desde el primer momento que la finalidad es académica y que al ser un proyecto de graduación, no incurrirá en ningún costo por lo que no se busca ningún beneficio económico que se pueda considerar como fraude informático. Estos artículos son adiciones a la Ley N° 4573 para reprimir y sancionar los delitos informáticos.

La ley 6683 de derechos de autor y derechos conexos protege los programas de cómputo como producción intelectual original. El prototipo desde su creación está protegido por los derechos de autor y mediante un acuerdo formal se pueden detallar los términos y condiciones respectivos para su uso, en este caso, con la empresa beneficiaria. Del mismo modo, el prototipo, al requerir los datos personales de los usuarios se apoyará en la ley 8968, sobre la protección de la persona frente al tratamiento de sus datos personales.

Proyecciones

Al realizar el correspondiente análisis sobre la situación actual de la empresa se encontraron varios problemas, tales como: que la empresa muestra dificultad para llevar las cuentas por pagar, o controlar adecuadamente sus ingresos y gastos y demás situaciones, por lo que, se pretende que el prototipo sea una herramienta que solvete estos problemas y la empresa se pueda beneficiar del sistema, al automatizar procesos que actualmente realiza de forma manual, lo que da paso a errores humanos. Se proyecta que con ayuda del prototipo la empresa tendrá mayor control de sus cuentas contables y podrá gestionar sus facturas, así como sus ingresos y gastos. Asimismo, podrá llevar el adecuado control de las cuentas por pagar y cuentas por cobrar y visualizar su presupuesto mensual.

Alcance Funcional

En cuanto al alcance funcional del proyecto, se busca abordar los problemas que la empresa enfrenta diariamente. En la Tabla 5 se muestra el nombre de cada módulo y su debida descripción, detallando sus características, que se plantean como solución para cada problema detectado.

Tabla 5

Descripción de los módulos

Nombre del módulo	Descripción del módulo
Gestionar cuentas contables	Este módulo se encargará de definir y estructurar el catálogo contable de la soda, clasificando los activos, pasivos, patrimonio, los ingresos y los gastos, además permitirá la generación de asientos contables provenientes de los módulos de facturación, ingresos y gastos, y ajustes de cierre mensual. Permitirá la creación y modificación de cuentas, así como la posibilidad de desactivar la cuenta.
Administrar cuentas por pagar	Este módulo controla las cuentas por pagar con los proveedores, de igual manera, incluye la totalidad por pagar, así como la fecha de emisión. Mostrará un registro de las facturas con fechas de vencimiento, las condiciones de pago, si es a contado o a crédito, así como alertas de vencimiento. Permitirá registrar pagos parciales o totales.
Gestionar facturación	Este módulo toma en cuenta los pedidos realizados, procesa las compras realizadas por parte de los clientes y crea una factura electrónica, especificando el producto adquirido, así como el detalle. En caso de que la compra sea a crédito,

Nombre del módulo	Descripción del módulo
	creará una cuenta por cobrar en el módulo correspondiente y registrará los ingresos en el módulo de Ingresos
Gestionar ingresos	Este módulo se encargará del registro de los ingresos de la soda, tomando en cuenta los ingresos por facturación, así como los ingresos de las cuentas por cobrar y cualquier otro ingreso extraordinario. Este módulo se integrará con el módulo de cuentas contables en dónde clasificará de manera automática los ingresos, de acuerdo con su origen.
Gestionar gastos	Este módulo se encargará de registrar y controlar los gastos directos de la soda, tanto los gastos diarios como los eventuales. Permitirá clasificar los gastos según su tipo, ya sean operativos, administrativos o de mantenimiento. Estará relacionado con el módulo de cuentas contables, mostrando los gastos reales, y con el módulo del presupuesto mensual, para realizar la comparación del gasto planificado contra el real.
Administrar cuentas por cobrar	Este módulo administra y da seguimiento de los pagos pendientes por parte de los clientes con la empresa, también tomará en cuenta los abonos realizados y al registrarlos recalculará el saldo pendiente. Facilitará y controlará el registro de las facturas emitidas a clientes. Enviará recordatorios automáticos de pagos. Contará con una integración con el módulo de Ingresos, el cual se actualizará al momento de recibir el pago.
Generar presupuesto mensual	Este módulo tendrá la funcionalidad de crear, controlar y ajustar el presupuesto mensual de la soda, el usuario podrá ingresar el presupuesto mensual y el módulo, al integrar los datos extraídos del módulo de gestión de cuentas contables, permitirá comparar los datos planificados inicialmente contra los datos reales; además, contará con avisos al usuario en caso de que se superen los límites presupuestados y mostrará gráficamente el desempeño financiero de la soda, esto para una mejor toma de decisiones en el aspecto financiero
MANTENIMIENTOS	Este módulo se encargará de realizar el borrado, inserción, modificación y actualización de datos

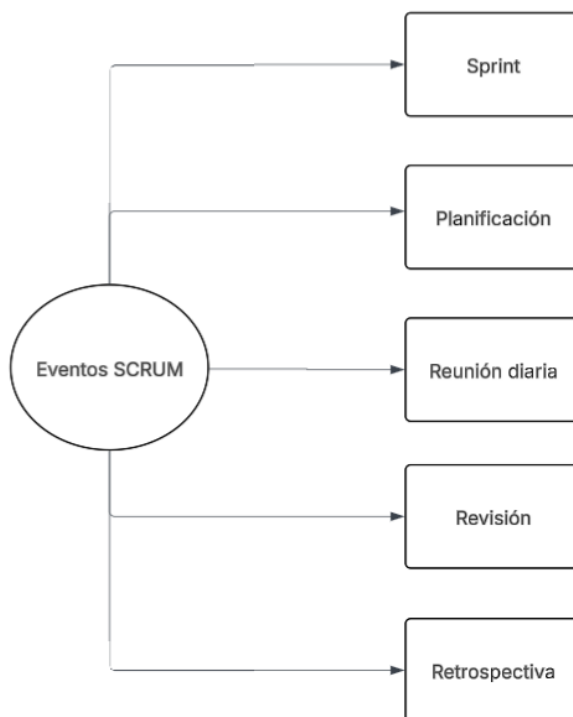
Nombre del módulo	Descripción del módulo
CONSULTAS	Este módulo se encargará de generar información proporcionada de las diferentes tablas
REPORTES	Este módulo se encargará de generar información proporcionada de las diferentes tablas y procesos, pero con un formato específico, según lo solicite el usuario. Podrá ser impreso o por pantalla.
SEGURIDAD	Este módulo se encargará de realizar la autenticación de contraseñas y definición de perfiles

Fuente: Elaboración propia

Alcance Metodológico

Con respecto al alcance metodológico del proyecto, se implementará una metodología ágil, en este caso, la metodología de Scrum, la cual cuenta con 5 eventos que se detallan en la Figura 1:

Figura 1
Eventos Scrum



Fuente: Elaboración propia

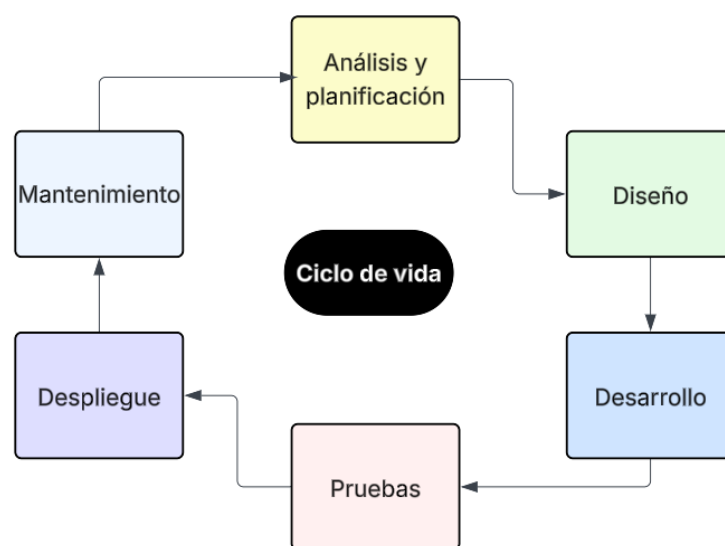
Como se muestra en la figura 1, Scrum es una metodología ágil que se centra en 5 eventos. Primeramente, se realizará un cronograma del “Sprint” que tomará en cuenta las semanas

necesarias para completar las tareas por realizar. Se llevará a cabo un “sprint planning” en donde se definirán dichas tareas , clasificándolas en orden de prioridad y tomando en cuenta el tiempo que tomará completarlas

Cada “sprint” necesitará de un seguimiento específico diario, con el objetivo de agilizar las tareas y priorizar el plan de trabajo. De forma semanal, se hará un “sprint review”, con el objetivo de revisar el trabajo realizado en la semana, compararlo con el cronograma y verificar que el el proyecto se desarrolle dentro del tiempo presupuestado; así mismo, se hará la última ceremonia del “sprint”, que será la retrospectiva, para la cual se tomarán en cuenta observaciones, ajustes o cambios que hayan surgido a ese punto, para dar por finalizado el “sprint” y dar inicio al siguiente.

Una vez detallada la metodología por utilizar, es importante entender el ciclo de vida de un “software”, el cual es fundamental para el desarrollo del prototipo, ya que aplicar el ciclo de vida al proyecto permitirá tener una estructura clara, tomando en cuenta las etapas que lo componen. En la siguiente figura se puede visualizar lo indicado:

Figura 2
Ciclo de vida del software



Fuente: Elaboración propia

Tal como se demuestra en la figura 2, el ciclo de vida del software cuenta con varias fases que lo componen para su correcto desarrollo. Primeramente, se encuentra la etapa de planificación y análisis de requerimientos, en esta fase se definen aspectos como viabilidad del proyecto, análisis

de riesgo, asignación de recursos, y se deben analizar los requerimientos, para entender qué es lo que realmente necesita el usuario.

Una vez se define la planificación y el análisis de requerimientos, se pasa a la fase de diseño, en donde se analizan los requisitos y se identifican las mejores soluciones para crear el software, también se elige la tecnología a desarrollar y las herramientas a utilizar; y se decide la estructura general del software.

En la fase de desarrollo o implementación se comienza la codificación del producto, a partir del análisis de las herramientas por utilizar y se procede a desarrollar la solución. Seguidamente, en la fase de pruebas, el equipo desarrollador realiza distintos tipos de ensayos, ya sean funcionales, y no funcionales, para corroborar si el software presenta errores y poder corregirlos antes de entregar el producto.

La siguiente fase corresponde al despliegue, que consiste en desplegar el software a producción para que el usuario pueda utilizarlo. En este punto, el software puede encontrarse en producción y en una copia, en un entorno de pruebas, al cual el usuario no tendrá acceso, ya que es para uso exclusivo del equipo de desarrollo.

Por último, se encuentra la fase de mantenimiento, durante la cual el equipo se encarga de corregir errores o resolver problemas del usuario. De igual forma, puede haber mantenimiento que añada nuevas funcionalidades o se adapte a nuevas necesidades, tomando en cuenta los requerimientos del cliente.

Alcance Tecnológico

Para el alcance tecnológico se implementará una arquitectura de tres capas, tomando en cuenta la capa de interfaz de usuario de presentación, la capa lógica y la capa de datos, tal y como se demuestra en la siguiente figura:

Figura 3
Arquitectura de tres capas



Fuente: Elaboración propia

En la figura 3 se muestran los tres niveles de capas que componen la arquitectura. En un primer nivel se encuentra la capa de presentación, que sería la interfaz que el usuario utiliza y con la que interactúa; en un segundo nivel se encuentra la capa lógica o nivel intermedio, en esta capa se encuentra la lógica de negocio, actúa como intermediario entre la capa de presentación y los datos, y por último, se encuentra la capa de datos, en donde se guardan los datos procesados en la base de datos. A continuación, se detallan las herramientas por utilizar, en la siguiente tabla:

Tabla 6
Software por utilizar

Software	Característica	Versión
Lenguaje de programación	C#	C# 14
Framework	Blazor	.NET 10
Base de datos	SQL Server Express	SQL Server 2022 Express
IDE	Microsoft Visual Studio	Visual Studio 2022
Versionado	Github	
Hospedaje	Blazor Server	.NET 10

Fuente: Elaboración propia

En la Tabla 6 se muestra el software por utilizar, tomando en cuenta el lenguaje de programación C#, utilizando el framework de Blazor y, para el hosting, el Blazor Server. En cuanto al gestor de base de datos, se utilizará la versión *express* de SQL Server, la cual es gratuita y la plataforma de Github, para el versionado del proyecto.

CAPÍTULO II: MARCO REFERENCIAL

El presente capítulo tiene como objetivo exponer conceptos, antecedentes, teorías y conformar una secuencia lógica para apoyar con fundamento teórico el desarrollo del prototipo propuesto, tomando en cuenta temas como las tecnologías a aplicar, arquitectura, patrón de diseño y demás, con el propósito de facilitar una base conceptual de comprensión clara de la solución planteada.

Prototipo de Software

El término de prototipo se puede interpretar de distintas maneras, en este contexto, como se está planteando una solución programable, la interpretación por desarrollar sería la de un prototipo de software. Para ello, Viera (2024) afirma: “un prototipo se entiende como una versión preliminar o una simulación de un producto o sistema que permite probar ideas antes de un lanzamiento oficial” (párr. 3).

Es importante el desarrollo de un prototipo, ya que permite al desarrollador identificar posibles errores, las funcionalidades planteadas, verificar que actúen cómo se planificó desde el primer momento, encontrar mejoras en cuanto a la experiencia del usuario, ya sea por temas de diseño, o propiamente del manejo de la aplicación, y también, si se puede simplificar u optimizar para evitar que al usuario se le presente como un obstáculo, en lugar de una solución.

El prototipo planteado se basa en una aplicación web, Chica (2026) plantea que: “una aplicación web es un software que se ejecuta en un servidor y al que se accede a través de un navegador web.” (párr. 2); ya que se busca, en un primer momento, que los usuarios al interactuar con el prototipo cuenten con accesibilidad total al mismo, y no depender de factores externos como que esté instalado en un computador específico. Dado que la empresa cuenta con acceso a internet y computadores, se puede asegurar que la accesibilidad total.

El prototipo tiene como objetivo poder solventar situaciones que representan atrasos, o poco control en sus operaciones diarias, para ello se propone el desarrollo de un software que cuente con los módulos necesarios, de manera tal que propongan una solución a las distintas problemáticas.

Desarrollo del Prototipo con C# y Blazor

Lenguaje de Programación C# con Blazor

Para desarrollar el prototipo, se utilizará el lenguaje de programación conocido como C#, que, de acuerdo con Barriuso (2024), “es un lenguaje de programación multiparadigma desarrollado por Microsoft. Se caracteriza principalmente por ser un lenguaje de programación orientado a objetos polivalente y fácil de aprender.” (párr. 5). Este lenguaje cuenta con gran versatilidad y sencillez para el desarrollo de aplicaciones, compatible con los navegadores web más utilizados, y permite el uso de elementos modernos como “*frameworks*” especializados en crear interfaces de usuarios interactivas, tales como *framework* Blazor.

Para entender el concepto de Blazor, primero es importante entender qué es un “*framework*”: se puede definir como una estructura o conjunto de bibliotecas, herramientas, buenas prácticas de programación ya determinadas, que funcionan como base para los programadores que desarrollan software. Tiene como propósito agilizar el desarrollo, ya que es posible reutilizar el código siempre que sea necesario, puesto que ya existe una estructura propiamente definida.

En el caso de C# existe un “*framework*” llamado Blazor que permite el desarrollo de aplicaciones web de una manera rápida y sencilla. Menciona Montesdeoca del Puerto (s. f.) que: “Blazor es un framework open-source y multiplataforma para crear aplicaciones webs.” (párr. 3). Permite trabajar tanto el “front-end”, ya que, está basado en HTML, CSS y C#, como el “back-end”, permitiendo manejar la lógica de negocios y los datos.

Blazor Server y Blazor WebAssembly

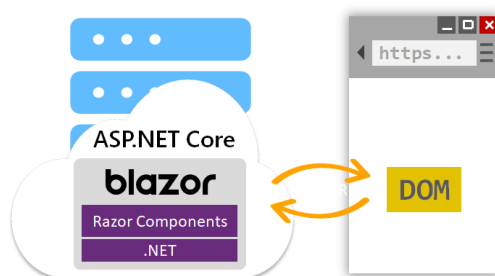
Al permitir el desarrollo de aplicaciones web, Blazor tiene opciones para alojar dichas aplicaciones, para ello cuenta con modelos de alojamiento populares como Blazor Server y WebAssembly. Ambos modelos poseen beneficios a la hora de alojar la aplicación, pero también con diferencias que pueden ser determinantes, dependiendo de la preferencia del cliente o de la naturaleza del proyecto.

En primer lugar, Blazor Server requiere de una conexión llamada SignalR, que actúa como un túnel al conectar los componentes de Blazor con el servidor, por lo que cada vez que se realice una acción, esta se recibe a través de la conexión. Este alojamiento está del lado del servidor, lo que permite que al ejecutar el programa exista un tiempo de respuesta inferior a si se ejecutara con

un alojamiento desde el cliente. De igual forma, en caso de que se interrumpa la conexión, mediante el conector SignalR, es posible tratar de reconectar con la opción que ofrece el servidor de recargar la página.

Figura 4

Representación de la conexión utilizando Blazor Server

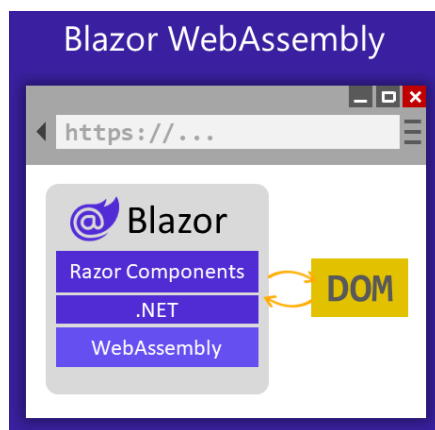


Fuente: Microsoft Learn

Asimismo, Blazor WebAssembly es un modelo de alojamiento que está del lado del navegador, es decir, una vez se accede a la aplicación, se descarga el código junto a sus dependencias, de esta manera no es necesario contar con un servidor conectado a través de un túnel como es el caso de Blazor Server. Dependiendo del tipo de sistema y necesidad que se tenga puede verse como un punto decisivo o en contra, ya que, si bien es cierto, es posible ejecutarse sin un servidor, la descarga es mucho mayor, y en casos de aplicaciones de gran tamaño puede afectar significativamente el rendimiento. (Montesdeoca, E. s. f).

Figura 5

Representación del modelo de alojamiento WebAssembly



Fuente: Blazorise.com

Para efectos del prototipo, se utilizará el modelo de alojamiento de Blazor Server, aprovechando la herramienta de conexión de SignalR, que Wadepickett. (s. f) define como: “SignalR es una biblioteca de código abierto que simplifica la incorporación de funciones web en tiempo real a las aplicaciones. La funcionalidad web en tiempo real permite que el código del servidor envíe contenidos a los clientes al instante.” (párr. 1).

La naturaleza del proyecto permite utilizar SignalR y, por lo tanto, Blazor Server como modelo de alojamiento, como una buena opción, debido a que el usuario requiere una aplicación con respuesta inmediata. Al contar con buena conexión a internet, el tema de la latencia no representa un inconveniente y debido a la prontitud con que el usuario requiere actuar dado el estilo de negocio, la aplicación sugerida es un buen candidato.

Desarrollo del Prototipo con SQL Server

Lenguaje de Base de Datos

Un aspecto igual de importante a tomar en cuenta en un sistema de este estilo es el lenguaje por utilizar para manejar los datos. Para efectos de este proyecto se estará utilizando el lenguaje de consulta estructurada (SQL), que Amazon Web Services (s. f) define como:

Un lenguaje de programación para almacenar y procesar información en una base de datos relacional. Una base de datos relacional almacena información en forma de tabla, con filas y columnas que representan diferentes atributos de datos y las diversas relaciones entre los valores de datos. Puede usar las instrucciones SQL para almacenar, actualizar, eliminar, buscar y recuperar información de la base de datos. También puede usar SQL para mantener y optimizar el rendimiento de la base de datos. (párr. 1).

Se utilizará este lenguaje de base de datos debido a su buena integración con el lenguaje de programación por utilizar, en este caso, con C#, y por sus componentes como las tablas, consultas y procedimientos almacenados De acuerdo con Amazon Web Services (s. f): “una tabla SQL es un elemento básico de una base de datos relacional” (párr. 5), al estar compuesta por filas y columnas permite mostrar de manera sencilla la relación de los componentes del sistema.

Las consultas SQL son instrucciones creadas por el desarrollador que el sistema de administración de base de datos entiende y ejecuta. Al ser un lenguaje de consultas estructuradas es un componente muy utilizado; ya que dentro de una consulta se pueden encontrar componentes como variables, condiciones lógicas e identificadores que permiten realizar la consulta necesaria.

Un procedimiento almacenado es un componente de SQL que se compone por una o varias instrucciones SQL almacenadas en la base de datos, son utilizados mayoritariamente para efectos de optimizar y mejorar el rendimiento, ya que, en lugar de consultar la misma instrucción SQL en distintas aplicaciones, se puede crear un procedimiento almacenado, para mejorar la eficiencia.

Gestor de base de datos

Una vez definido el lenguaje de bases de datos, es importante definir el gestor que se va a utilizar, pero, primeramente, explicar lo que es un sistema de gestor de base de datos. Darías (s. f) define un gestor de base de datos como: “un conjunto de programas invisibles para el usuario final con el que se administra y gestiona la información que incluye una base de datos” (párr. 2).

El gestor de base de datos representa una interfaz gráfica entre el usuario y la base de datos por utilizar, en él se permiten operaciones como crear, administrar y manipular grandes cantidades de datos. El gestor de base de datos controla cualquier operación que realice el desarrollador contra la base de datos.

Existen distintos tipos de gestor, que se pueden clasificar con base en el modelo de datos y en la cantidad de usuarios. Con base en el modelo de datos, existen tipos como: sistema jerárquico, sistema en red, sistema relacional, sistema multidimensional y sistema de objetos. Para efectos del proyecto y al requerir almacenamiento de datos en filas y columnas conformando relaciones entre tablas por medio de llaves, el tipo indicado es el sistema relacional.

En cuanto al tipo de gestor basado en su distribución, se considera un sistema de gestión de datos centralizados, ya que, para efectos del proyecto el sistema de gestión y la base de datos estarían almacenados en un único lugar. Teniendo claro tanto el lenguaje de base de datos, como la definición de gestor de bases de datos, se decide utilizar Microsoft SQL Server como gestor de base de datos. Darías (s. f) define a Microsoft SQL Server como:

Uno de los principales sistemas de gestión de bases de datos relacional del mercado que presta servicio a un amplio abanico de aplicaciones de software destinadas a la inteligencia

empresarial y análisis sobre entornos corporativos. Basada en el lenguaje Transact-SQL, incorpora un conjunto de extensiones de programación propias de lenguaje estándar y su aplicación está disponible para usarse tanto a nivel *on premise* o bajo una modalidad *cloud*. (párr. 3).

SQL Server cuenta con diversidad de herramientas destinadas a la gestión y análisis de datos, permite integración de manera sencilla con la aplicación desarrollada en Visual Studio. Asimismo, es de código abierto, por lo que es accesible y sus funcionalidades de memoria integrada permiten incrementar la flexibilidad y facilidad de uso.

Existen distintas ediciones disponibles para su uso, en primer lugar, se encuentra la edición *Enterprise*, destinada para empresas que manejen altos niveles de datos y requieran de mayores opciones disponibles y de una edición más completa. La edición estándar, está orientada a pequeñas empresas ya que ofrece herramientas básicas de administración de datos.

La edición *Express* es gratuita, ideal para desarrolladores independientes y para compilar pequeñas aplicaciones, y la edición *Developer* que muestra todas las funcionalidades como la *Enterprise*, pero en un entorno de pruebas únicamente, pues no está disponible para un entorno de producción. Para efectos de este proyecto, la edición seleccionada para gestionar la base de datos será la edición de SQL Server Express.

Operaciones CRUD en SQL

Se conoce como CRUD a un conjunto de operaciones básicas que se realizan para la gestión de datos, Ostrowska (2024) menciona que: “CRUD es un acrónimo que significa Crear, Leer, Actualizar y Eliminar. Estas son las cuatro operaciones básicas que se pueden realizar sobre los datos almacenados en una base de datos u otros sistemas de almacenamiento de datos.” (párr. 1).

En el caso del prototipo, se estarán utilizando estas operaciones para el manejo de datos de manera estandarizada, con el propósito de que el usuario pueda visualizar la información en reportes o consultas, para ello es importante describir las operaciones y entender su funcionamiento.

La operación de crear, como su nombre lo indica, crea nuevos registros o datos al sistema. En SQL se utiliza la palabra reservada *Insert* para agregar datos. La operación leer, o *Read*, es utilizada para consultar datos de la base de datos y al realizar esta operación, el usuario podrá

visualizar sin posibilidad de editar la información. La operación de actualización o *update*, es utilizada para modificar un registro previamente insertado, y la operación borrar, o *delete* se emplea para eliminar un registro de la base de datos.

Contexto Contable del Prototipo

Sistemas Contables

En este punto de la investigación se han detallado aspectos técnicos como las herramientas por utilizar para el desarrollo del prototipo, que, si bien es cierto, ha sido importante definir estos términos, también lo es definir el tipo de sistema que se espera desarrollar, que corresponde a un software de contabilidad y facturación. Oracle (s. f) define al software de contabilidad como:

Un software que gestiona y registra las transacciones financieras diarias de una organización, incluido el activo fijo, los gastos, los ingresos, las cuentas a cobrar, las cuentas por pagar, la contabilidad de libros auxiliares y la presentación de información y la realización de análisis. Un sistema de contabilidad completo debe realizar el seguimiento de los activos, los pasivos, los ingresos y los gastos de una organización. Estas transacciones se registran en el libro mayor de manera automáticamente, proporcionando a los CFO, los tesoreros y los controllers acceso inmediato a datos financieros precisos en tiempo real. También permite a los responsables de la cuenta de resultados ver su desempeño a nivel operativo. (párr. 1).

La empresa cuenta con muy poco control de sus transacciones financieras y al no contar con un sistema que automatice o que permita la visualización de sus transacciones, la gestión financiera ha representado un gran problema. Debido a esta situación, se decidió proponer como solución un software que cuente con los módulos necesarios para gestionar los ingresos, gastos, cuentas por cobrar, pagar y demás funcionalidades. Algunas de las ventajas que presenta contar con un software de contabilidad en la empresa se describen a continuación:

- Gestionar todo el ciclo financiero de los activos, incluidas la adquisición, la capitalización, la amortización y la baja en cuentas. Contar con esta visibilidad completa ayuda a las empresas a ofrecer estados financieros precisos en términos de revalorización y amortización.

- Un sistema de contabilidad completo no solo registra transacciones financieras, sino que también incluye capacidades de presentación de información y realización de análisis.
- Ayuda a las organizaciones a reducir la redundancia de registros, ofrecer mejores presupuestos y previsiones y llevar a cabo una gestión de gastos exhaustiva y adecuadamente categorizada.

El desarrollo del prototipo busca solventar las necesidades y desafíos que enfrenta la empresa Soda Creston diariamente, desarrollar un sistema contable y de facturación, proveer soluciones, automatizar procesos, y facilitar la labor diaria de la empresa. Para ello, es importante definir los procesos contables con los que contará el prototipo, mediante los distintos módulos, para atender los requerimientos del negocio.

El sistema tendrá un módulo encargado de gestionar las cuentas contables, de esta forma, la empresa podrá clasificar los activos, pasivos, el patrimonio, los ingresos y los gastos. Las cuentas contables, según Sage (2025), se definen como: “registros estructurados que reflejan todas las transacciones económicas de una empresa. Por lo tanto, se organizan de manera cronológica y categorizada, permitiendo un mejor control financiero.” (párr. 3).

La empresa deberá administrar las cuentas por pagar y por cobrar, para ello, el prototipo contará con módulos específicos para realizar dicha administración. De acuerdo con Úrzua (2026), las cuentas por pagar se definen como: “deudas u obligaciones que tiene tu negocio a sus proveedores o a sus acreedores, según sea el caso porque contrataste un servicio o producto.” (párr. 2).

Por otro lado, las cuentas por cobrar se administrarán en un módulo que permita dar seguimiento a los pagos pendientes por parte de los clientes, así como también visualizar saldos pendientes, en caso de que se hayan realizado abonos. Moonflow (2023) define una cuenta por cobrar como: “el dinero que los clientes deben a una empresa por bienes o servicios proporcionados a crédito.” (párr. 2).

El módulo del presupuesto mensual contará con la información proveniente de los módulos de gestión de cuentas contables, tomando en cuenta los ingresos, gastos y demás gestiones contables. Contar con un presupuesto mensual representa una ventaja, Miro (s. f.) detalla que el

presupuesto mensual: “promueve un enfoque disciplinado de la gestión financiera, optimiza la asignación de recursos y actúa como un sistema de detección temprana de posibles problemas de liquidez.” (párr. 3)

Contexto de Facturación del Prototipo

Sistemas de Facturación

Al igual que con el sistema de contabilidad, el prototipo contará con un módulo de facturación, por lo que es importante entender qué es un sistema de facturación, así como los distintos tipos de facturación que existen. Para ello Stripe (2024) menciona que: “los sistemas de facturación permiten a las empresas crear y gestionar facturas. Estos sistemas suelen incluir funciones como plantillas de facturas personalizables, recordatorios automáticos para pagos vencidos, integración con software de contabilidad e informes detallados” (párr. 1).

El prototipo contará con un módulo para gestionar la facturación, debido al poco control que tiene la empresa con respecto al manejo de las facturas. El módulo contará con integración con los módulos de contabilidad, para ayudar a automatizar procesos que se realizan de forma manual o de forma desordenada.

El módulo de facturación se encargará de crear una factura electrónica la cuál GTI (s. f) define como: “Es el documento que se envía por medios informáticos tanto a Hacienda, como al correo electrónico del cliente.” (párr. 1). El módulo al hacer efectiva la factura, estará completando el proceso de una transacción, que según Square (s. f) se entiende como transacción: “un acuerdo entre dos partes: un comprador y un vendedor. En una transacción, el vendedor proporciona productos, servicios u otros activos financieros a cambio de fondos.” (párr. 1).

Existen diversos tipos de sistemas de facturación, aplicables de acuerdo con la naturaleza de la empresa, ya que cada negocio presenta sus propias necesidades y requerirá de un sistema específico para atender sus requerimientos particulares. Para efectos del prototipo, el módulo de facturación contará con un funcionamiento similar al tipo de software básico de facturación, ya que, Stripe (2024) define este tipo de software como:

El software de facturación básico ofrece las funciones más importantes para generar, enviar y hacer seguimiento de las facturas. Estos sistemas pueden gestionar facturas recurrentes, recordatorios de pagos simples y algunas automatizaciones fundamentales. Suelen ser

fáciles de usar y accesibles y ofrecen más automatización que los métodos manuales. (párr. 6).

Si bien es cierto, existen diversos tipos de software de facturación, para efectos del proyecto se estará desarrollando un módulo específico, ya que la empresa requiere de un software que gestione la facturación, por lo que el módulo de facturación se integrará con los demás módulos de gestión contable del prototipo.

Una de las grandes fortalezas de un sistema de facturación es su integración con otros sistemas, Stripe (2024) indica: “Para muchas empresas, el verdadero poder de un sistema de facturación es su capacidad para integrarse con otras herramientas, como las plataformas de contabilidad, los sistemas de gestión de relaciones con los clientes (CRM) y las soluciones ERP” (párr. 20).

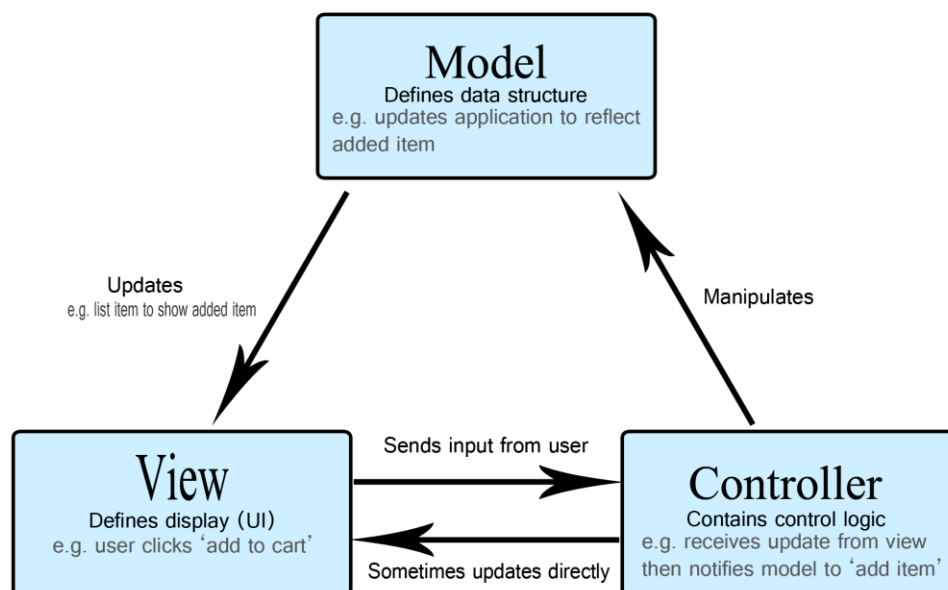
Estructura del Prototipo

Patrón de diseño

Una vez mencionados los tipos de sistemas contables y de facturación y los módulos que requiere el prototipo, es importante entender cómo estará estructurado el software y el patrón de diseño que seguirá. En este caso, se utilizará el patrón conocido como MVC, o el patrón modelo, vista, controlador.

Para definir el patrón, MDN (2023) indica: “MVC es un patrón en el diseño de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control” (párr. 1). Este patrón separa en 3 capas el código del proyecto, teniendo una capa específica de manejo de datos, una capa de diseño y una capa intermediaria, como se detalla en la figura 6.

Figura 6
Patrón de diseño MVC



Fuente: MDN. developer.mozilla.org

La figura 6 es una representación del modelo MVC, la primera capa corresponde a la capa del modelo, Smith (s. f) define las responsabilidades del modelo como:

El modelo en una aplicación de MVC representa el estado de la aplicación y cualquier lógica de negocios u operaciones que esta deba realizar. La lógica de negocios debe encapsularse en el modelo, junto con cualquier lógica de implementación para conservar el estado de la aplicación. Las vistas fuertemente tipadas normalmente usan tipos ViewModel diseñados para que contengan los datos para mostrar en esa vista. El controlador crea y rellena estas instancias de ViewModel desde el modelo. (párr. 4).

El modelo es la capa que define qué datos debe contener la aplicación, pero esos datos no se visualizan en la aplicación desde el modelo, el usuario logra interactuar con la aplicación gracias a la segunda capa, que es la capa Vista, ya que, según Smith (s. f): “Las vistas se encargan de

presentar el contenido a través de la interfaz de usuario” (párr. 5). La capa de la vista define cómo se deben mostrar los datos de la aplicación.

En cuanto a la capa del controlador, ésta contiene la lógica necesaria para actualizar el modelo y la vista, de acuerdo con las entradas de los usuarios en la aplicación. Para definir las responsabilidades del controlador, Smith (s. f) explica:

Los controladores son los componentes que controlan la interacción del usuario, trabajan con el modelo y, en última instancia, seleccionan una vista para representarla. En una aplicación de MVC, la vista solo muestra información; el controlador controla y responde a la interacción y los datos que introducen los usuarios. En el patrón de MVC, el controlador es el punto de entrada inicial que se encarga de seleccionar con qué tipos de modelo trabajar y qué vistas representar (de ahí su nombre, ya que controla el modo en que la aplicación responde a una determinada solicitud). (párr. 6).

El modelo MVC, al aplicarse en los proyectos ofrece grandes y numerosas ventajas, ya que cuenta con una estructura modular y una separación de responsabilidades que permite crear aplicaciones robustas. Entre estas ventajas, la separación de responsabilidades permite mantener una buena estructura y código fácil de entender. Otra ventaja corresponde a la posibilidad de reutilizar el código, ya que el mismo modelo permite crear diferentes vistas.

Gracias a su estructura modular, admite realizar cambios de manera sencilla, ya que, en caso de actualizar la lógica de negocio, se requiere actualizar la capa del modelo, sin necesidad de modificar la capa de vista o controlador. Asimismo, el patrón MVC es soportado por gran cantidad de “frameworks”, esto logra facilitar la implementación del patrón en distintas tecnologías.

Seguridad del Prototipo

Seguridad de la Información

Cuando se desarrolla un sistema, un aspecto fundamental es la seguridad de la información, especialmente si el sistema contiene datos sensibles como de facturas o cuentas contables, por lo que este es un aspecto que se debe considerar de suma importancia y por ende, seguir los principios de seguridad de la información.

En primer lugar, es importante definir qué es la seguridad de la información, abreviado como *InfoSec*, para ello Holdsworth y Kosinski (2025) detallan:

La seguridad de la información (InfoSec) es la protección de la información importante contra el acceso, la divulgación, el uso, la alteración o la interrupción no autorizados. Ayuda a garantizar que los datos confidenciales de la organización estén disponibles para los usuarios autorizados, permanezcan confidenciales y mantengan su integridad. (párr. 1).

Las grandes organizaciones se encargan de proteger los datos, especialmente si son datos sensibles como datos financieros, porque las organizaciones están expuestas a recibir ataques cibernéticos, ya que los ciberdelincuentes conocen el valor que tienen los datos o la información confidencial y exigen grandes cantidades de dinero para recuperar esa información. Esto se conoce como “*ransomware*”, por lo tanto, es importante conocer prácticas de seguridad para aplicar en sistemas que manejen datos sensibles.

Las buenas prácticas de seguridad se basan en principios, estos principios son: la tríada CID, aseguramiento de la información y no repudio. De acuerdo con Holdsworth y Kosinski (2025): “la tríada CID pretende guiar a las organizaciones en la elección de tecnologías, políticas y prácticas para proteger sus sistemas de información. Los elementos de la tríada de la CID incluyen: confidencialidad, integridad y disponibilidad” (párr. 19).

La confidencialidad de la información garantiza que los datos sensibles se mantengan en un estatus privado y evita el acceso sin autorización. En el caso de que un usuario sin acceso obtenga de manera deshonesta o sin autorización información, estaría violentando la confidencialidad, y particularmente, en caso de un sistema contable y de facturación, como el prototipo propuesto, estaría adquiriendo información sensible.

La integridad de la información se refiere al esfuerzo que existe para que la información esté completa y precisa, es decir, evitar manipulación, alteración o eliminación de datos no autorizados.

Al contrario de la confidencialidad, se encuentra la disponibilidad, que consiste en asegurar el correcto acceso de los usuarios a la información que tienen autorizada, para ello se debe garantizar la solidez del producto, para que el usuario tenga la posibilidad operativa de disponer de la información cuando la requiera.

El principio de aseguramiento de la información corresponde, según Holdsworth y Kosinski (2025) a: “El proceso continuo de lograr la confidencialidad, integridad y disponibilidad de los datos dentro de un sistema de información”. Y el principio del no repudio garantiza que únicamente los usuarios autorizados trabajen con los datos y puedan utilizarlos o modificarlos de manera acreditada.

Estos principios son prácticas de seguridad que se deben aplicar para proteger la información, ya que existen leyes que penalizan con tiempo de prisión los sabotajes de información, o alteración de datos, por fallas de protección de información personal. Dado lo anterior, es importante tomar en cuenta herramientas o técnicas que protejan al usuario de ataques cibernéticos, como la criptografía, o contar con un “firewall” que impida tráfico sospechoso, realizar evaluaciones de riesgos e identificar vulnerabilidades, como buenas prácticas para proteger la información.

CAPÍTULO III: MARCO METODOLÓGICO

Enfoques de Investigación

Existen diferentes enfoques de investigación que ayudan a determinar la naturaleza de un estudio, definen la ruta metodológica y permiten la recolección de información para su debido análisis. El enfoque cuantitativo se centra en la recolección de datos sobre variables, el enfoque cualitativo utiliza la recolección de información para desarrollar hipótesis, y el enfoque mixto integra ambos en un solo método.

Enfoque Cuantitativo

El enfoque cuantitativo es un método de investigación estructurado que se basa en la recolección y análisis de datos cuantitativos sobre variables. Sampieri et al. (2014) definen el enfoque cuantitativo como: “Utiliza la recolección de datos para probar hipótesis con base en la medición numérica y el análisis estadístico, con el fin establecer pautas de comportamiento y probar teorías.” (p. 4).

El enfoque cuantitativo cuenta con una serie de pasos que se deben seguir para su correcta implementación, estos pasos deben respetar un orden establecido y, a diferencia de otros enfoques de investigación, no es posible regresar a pasos anteriores, o salir del orden que se tiene previamente señalado., Sampieri et al (2014) explica las fases del enfoque cuantitativo de la siguiente manera:

Parte de una idea que va acotándose y, una vez delimitada, se derivan objetivos y preguntas de investigación, se revisa la literatura y se construye un marco o una perspectiva teórica. De las preguntas se establecen hipótesis y determinan variables; se traza un plan para probarlas (diseño); se miden las variables en un determinado contexto; se analizan las mediciones obtenidas utilizando métodos estadísticos, y se extrae una serie de conclusiones. (p. 4).

Enfoque Cualitativo

En el caso del enfoque cualitativo, Sampieri et al. (2014) lo definen como: “Utiliza la recolección y análisis de los datos para afinar las preguntas de investigación o revelar nuevas interrogantes en el proceso de interpretación”. Se diferencia del método cuantitativo porque parte de procesos de recolección de datos basados en la comunicación, más que en procesos lógicos o

estadísticos. El enfoque cualitativo se centra en desarrollar hipótesis y preguntas antes, durante y después de la recolección y análisis de datos, la investigación cualitativa se centra en el “por qué” en lugar del “qué”. Este enfoque proporciona ideas y ayuda a desarrollar hipótesis.

Enfoque Mixto

El enfoque mixto de investigación busca combinar características del cuantitativo y del cualitativo, formando un método que integre ambos. Supera las limitaciones que se pueden encontrar aplicando un solo método, permite validar hallazgos y explora el contexto de manera detallada. Medina et al. (2023) lo define como:

Al combinar los métodos cuantitativos, que se centran en la recopilación y análisis de datos numéricos para identificar patrones, relaciones y tendencias, con los métodos cualitativos, que se enfocan en la comprensión y explicación de significados, experiencias y contextos sociales, la investigación mixta permite capturar la complejidad y multidimensionalidad de los fenómenos investigados. (p.18).

Enfoque de Investigación Seleccionado

El enfoque de investigación seleccionado será el enfoque mixto, ya que el trabajo requiere una comprensión del contexto, entender las necesidades que enfrenta la empresa, así como el funcionamiento de los procesos actuales, para ello es importante tomar en cuenta la opinión del personal, ya que permite recolectar información utilizando técnicas como observación no estructurada o entrevistas abiertas, de esta forma cuenta con características del enfoque cuantitativo.

Del mismo modo, debido a la naturaleza de la empresa, es necesario contar con características de un enfoque cuantitativo, ya que se requiere un análisis de datos numéricos medibles, que puedan servir de parámetro para la toma de decisiones, así como conocer tendencias y poder identificar fortalezas y debilidades. Para ello, existen técnicas de investigación como encuestas, utilizando preguntas cerradas, o realizar un análisis estadístico de los datos.

Tipos de Investigación

Los tipos de investigación representan los distintos enfoques o el modo en que se pueda realizar una investigación. Existen distintos tipos de investigación, en donde se pueden clasificar según el enfoque, ya sea cualitativo o cuantitativo, según el nivel de profundidad, el propósito, el tipo de estudio o según el tipo de datos. Cada tipo de investigación busca obtener o adquirir un conocimiento.

Investigación Descriptiva

La investigación de tipo descriptiva se basa en observaciones empíricas, en esencia es describir lo que existe, ya que se utiliza para recopilar, analizar y presentar datos con la finalidad de describirlos en su contexto natural. Stewart (2025) define la investigación descriptiva como: “La investigación descriptiva se define como un método de investigación que observa y describe las características de un determinado grupo, situación o fenómeno.” (párr. 5).

Investigación Exploratoria

El tipo de investigación exploratoria es utilizado para estudiar temas poco conocidos, con el propósito de identificar patrones y obtener una primera aproximación. Se suele aplicar con instrumentos como entrevistas, estudios de caso u observación. Berumen (2025) la define como: “un tipo de estudio que se lleva a cabo cuando se necesita obtener una comprensión inicial de un problema o identificar áreas de oportunidad.” (párr. 2).

Es un tipo de investigación flexible, puesto que el investigador se adapta con base en sus descubrimientos, y no existe una ruta o un camino previamente marcado. Este tipo de investigación no busca respuestas concluyentes, sino encontrar información que marque el inicio de futuras investigaciones y su propósito radica en generar hipótesis, formular ideas, sin necesidad de verificarlas mediante procedimientos científicos.

Investigación Explicativa

La investigación explicativa busca profundizar en las causas y los efectos, es un tipo de investigación estructurado y se centra en explicar los motivos por los que ocurre un acontecimiento. Sampieri (2014) lo define como: “Los estudios explicativos son más que la descripción de conceptos o fenómenos o el establecimiento de relaciones entre variables; más bien, están diseñados para determinar las causas de los eventos y fenómenos físicos o sociales.” (p. 78).

Tipo de Investigación Seleccionado

El tipo de investigación seleccionado es el descriptivo, ya que la propuesta del sistema, al contar con procesos contables y de facturación, requiere identificar las características de la empresa, así como las necesidades y problemáticas que enfrenta diariamente. En un enfoque descriptivo se enumeran variables de las cuáles se obtiene información para describirlas, y para efectos de la investigación, esta modalidad permitirá recopilar datos del negocio para proponer soluciones que respondan a las necesidades identificadas.

Fuente de Información

Las fuentes de información son los medios por los que se obtiene la información, como lo menciona Ubaldo (2008) son: “todos aquellos medios de los cuáles procede la información, que satisfacen las necesidades de conocimiento de una situación o problema presentado y, que posteriormente será utilizado para lograr los objetivos esperados.” (p. 8). Existen distintos tipos de fuentes de información: primaria, secundaria y terciaria, que se detallan a continuación:

Fuente de Información Primaria

La fuente de información primaria se refiere a aquella que se obtiene de manera directa con el autor, ya sea, mediante instrumentos como una entrevista cara a cara u otros medios como testimonios, grabaciones o documentos oficiales, en donde el autor actúa como la primera fuente. Maranto (2015) refuerza esta idea definiendo estas fuentes de información primaria como:

Contienen información directa antes de ser interpretada, o evaluado por otra persona. Las principales fuentes de información primaria son los libros, monografías, publicaciones periódicas, documentos oficiales o informe técnicos de instituciones públicas o privadas, tesis, trabajos presentados en conferencias o seminarios, testimonios de expertos, artículos periodísticos, videos documentales, foros. (p. 3)

Fuente de Información Secundaria

La fuente de información secundaria interpreta y analiza la información proveniente de las fuentes primarias, ya sea mediante páginas web, libros o artículos. Maranto (2015) define las fuentes de información secundarias como: “El proceso de esta información se pudo dar por una interpretación, un análisis, así como la extracción y reorganización de la información de la fuente primaria.” (p. 3).

Fuente de Información Terciaria

La fuente de información terciaria es una recopilación y organización de fuentes primarias y secundarias, Maranto (2015) define la fuente de información terciaria como las que: “recopilan fuentes de información primarias o secundarias. Estas fuentes son utilizadas para buscar datos para obtener una idea general sobre algún tema.” (p. 3). Algunos ejemplos de esta fuente de información son bibliografías o enciclopedias.

Variables

Una variable de investigación es una característica medible y como su nombre lo indica, puede variar de valor, y su valor permite ser manipulado en un trabajo de investigación. Parra (2023) define las variables como: “Una variable representa un atributo medible que cambia a lo largo de un experimento comprobando los resultados. Estos atributos cuentan con diferentes medidas, dependiendo tanto de las variables, del contexto del estudio o de los límites que los investigadores consideren.” (párr. 3). Existen distintos tipos de variables, a saber, conceptuales, operaciones e instrumentales, que se explican seguidamente:

Variables Conceptuales

Una variable conceptual, como su nombre lo indica, se refiere a una definición conceptual o teórica dentro de una investigación. Salusplay (s. f) define la variable conceptual como: “definiciones de diccionarios o de libros especializados que describen la esencia o las características de una variable, objeto o fenómeno. Constituyen la adecuación de la definición conceptual a los requerimientos prácticos de la investigación.” (párr. 8).

Variables Operacionales

Las variables operacionales, en lugar de centrarse en explicar qué es una variable en teoría, se centra en la medición de esta, Sampieri (2014) destaca: “procedimientos que describe las acciones que un investigador debe realizar para recibir las impresiones sensoriales, las cuales indican la existencia de un concepto teórico en mayor o menor grado.” (p. 88). Describe el proceso que debe realizar para obtener la información requerida.

Variables Instrumentales

Las variables instrumentales hacen referencia a la manera o al instrumento que se va a utilizar para recopilar la información, es la forma en que se permite medir la variable, por ejemplo,

utilizando una entrevista o cuestionario; representan los instrumentos de medición por los cuáles se obtiene la información requerida sobre las variables conceptuales y operacionales para el trabajo de investigación.

Tabla 7
Cuadro de variables

Objetivo Específico	Variable	Variable conceptual	Variable operacional	Variable instrumental
Analizar los requerimientos del negocio con el objetivo de una adecuada implementación del sistema, de acuerdo con las necesidades encontradas.	Requerimientos	Según Cristiá (2024) un requerimiento es: “una característica del sistema o una descripción de algo que el sistema es capaz de hacer con el objeto de satisfacer el propósito del sistema” (p. 4)	Encuesta Entrevista	Cuestionario Guía de entrevista
Diseñar el modelo de la base de datos utilizando la herramienta MySQL Workbench y la lógica de la aplicación, de acuerdo con los requerimientos	Diseño de base de datos Lógica de la aplicación	De acuerdo con Duque (2024) el diseño de una base de datos es: “el proceso de crear una estructura organizada para almacenar, gestionar y recuperar datos de manera eficiente (párr. 12). Muñoz (2024) define la lógica	Documentos de análisis Casos de uso	MySQL Workbench Figma

Objetivo Específico	Variable	Variable conceptual	Variable operacional	Variable instrumental
		de la aplicación, como: “Un conjunto de reglas y conceptos que se utilizan para organizar y planificar las instrucciones de un programa. Sin una buena comprensión de la lógica de programación, es imposible escribir programas que funcionen correctamente.” (párr. 1)		
Desarrollar los módulos del sistema, de acuerdo con el análisis realizado previamente.	Desarrollo de los módulos del sistema	Quiroz (s. f) define el desarrollo de software como: “un producto que satisfaga las necesidades de los usuarios y los objetivos empresariales de forma eficaz, repetible y segura”. (párr. 3)	Diagrama entidad relación Casos de uso Diagramas UML	Visual Studio Code MySQL Workbench Draw.io
Probar los distintos componentes a	Pruebas de sistema	Tamushi (2024) define las pruebas de sistema, como:	Pruebas unitarias Pruebas integrales	Visual Studio Code Casos de prueba

Objetivo Específico	Variable	Variable conceptual	Variable operacional	Variable instrumental
través de pruebas unitarias e integrales.		“un conjunto de procedimientos y evaluaciones que tienen como objetivo verificar que un programa o aplicación funcione según lo esperado. (párr. 2).		

Fuente: Elaboración propia

Población

La población es definida por López (2025) como: “conjunto completo de personas, objetos o elementos que comparten una característica común que nos interesa estudiar.” (párr. 3). El tamaño de la población se determina por la totalidad de individuos que la componen, por lo que, para efectos del prototipo, la empresa Soda Creston cuenta con una población total de 6 colaboradores, que corresponden a la totalidad de individuos que la conforman.

Muestra

López (2025) define la muestra, como: “un subconjunto o una pequeña parte representativa que se extrae de un grupo más grande, conocido como población.” (párr. 3). El cálculo de la muestra se realiza cuando la población corresponde a un gran número de individuos y no es factible obtener respuesta de todos, por lo que se extrae una muestra representativa. Debido al tamaño de la población de la empresa, que está conformada por 6 colaboradores, la muestra se tomará en cuenta como la población total, por lo que no hay cálculo de muestra.

Instrumentos de Recolección de Datos

Los instrumentos de recolección de datos actúan con el objetivo de recolectar información mediante técnicas o herramientas. Vásquez (2023) define los instrumentos de recolección de datos como: “conjunto de herramientas utilizadas por los investigadores para obtener información

relevante sobre los eventos de estudio.” (párr. 4). Existen diversos tipos de instrumentos como entrevistas, observación o cuestionarios, para efectos de la investigación, y debido a que es de tipo mixta se empleará la entrevista, la cual sigue la metodología cualitativa, y la encuesta, que sigue la metodología cuantitativa.

Entrevista

La entrevista es un instrumento de recolección de datos que requiere la participación de un entrevistador, y un entrevistado. Busca recolectar información mediante una serie de preguntas previamente estructuradas o no por el entrevistador. Existen varios tipos de entrevista. Muguira (2025) define el tipo estructurada, como: “es aquella en la que el investigador plantea a los participantes una lista de preguntas preparadas de antemano.” (párr. 16).

La entrevista no estructurada se basa en dar seguimiento a las respuestas brindadas por el entrevistado y se suelen utilizar preguntas abiertas, no planteadas previamente. Las entrevistas semiestructuradas son una mezcla entre las entrevistas estructuradas y no estructuradas, ya que cuenta con preguntas predeterminadas, pero no necesariamente debe seguir un orden en específico.

Encuesta

La encuesta es utilizada en un enfoque cuantitativo y consiste en que, mediante una serie de preguntas cerradas y escritas, la persona encuestada responda y así el investigador pueda recolectar y analizar la información. Duarte (2024) la define como: “una técnica que emplea un conjunto de procedimientos estandarizados para recopilar y analizar datos de una muestra representativa de una población.” (p. 94).

Proceso para Recolección y Análisis de Datos

La recolección de datos se realizará aplicando los instrumentos de entrevista y encuesta, ya que al asumir el enfoque mixto, este cuenta con características cuantitativas, que se miden con una encuesta, y características cualitativas, para la cual se realizará una entrevista semiestructurada. Con respecto a la entrevista, ésta se realizará al personal encargado del área contable, con el objetivo de recopilar información sobre el proceso actual de la empresa con respecto a los ámbitos contables, así como los desafíos que presenta.

En cuanto al aspecto cuantitativo de la investigación, se medirá mediante una encuesta con preguntas cerradas dirigida al personal de la empresa, con el objetivo de recopilar información

acerca del nivel de satisfacción con respecto a la operatividad de la empresa actualmente, frecuencia de error humano en las tareas diarias y la necesidad de automatizar los procesos que se realizan de forma manual.

Con respecto al análisis de datos, para la entrevista se realizará un análisis temático, que según Hecker y Kalpokas (2025): “Permite a los investigadores destilar datos de las transcripciones de entrevistas u otras fuentes cualitativas para encontrar puntos en común y tendencias.” (párr. 3). La información recolectada con la encuesta se podrá visualizar mediante gráficos que permitan realizar el análisis respectivo, para interpretar los resultados obtenidos.

CAPÍTULO IV: ANÁLISIS DE RESULTADOS

En este apartado se presentan los resultados obtenidos de la aplicación de los instrumentos de recolección de datos. Se aplicó el instrumento de cuestionario a los trabajadores de la empresa, y se realizó una entrevista al dueño de la Soda Crestón. Dichos instrumentos permiten entender la perspectiva de los colaboradores en el funcionamiento actual de la empresa y recolectar información sobre procedimientos específicos, tales como la gestión contable, el proceso de facturación y el manejo de gestión de pedidos.

Entrevista

Preguntas

Pregunta 1

¿Cómo realiza la gestión contable de la empresa?

La contabilidad que se realiza es básica, se hace un control de compras y se tributa el porcentaje de las compras a Hacienda. El control se maneja en Excel.

Pregunta 2

¿Quiénes están encargados de realizar el proceso de gestión contable?

El proceso contable está a cargo de 1 persona, Óscar Delgado

Pregunta 3

¿Cómo se realiza la facturación actualmente?

Las facturas de la soda se realizan al colegio, utilizando el sistema de Hacienda.

Pregunta 4

¿Cómo registran las transacciones realizadas por los clientes?

Por Excel se lleva el control

Pregunta 5

¿Cómo gestiona la cantidad de pedidos que recibe diariamente?

Por WhatsApp, los padres de familia realizan los pedidos para el almuerzo del hijo o hija mediante mensajes de WhatsApp.

Pregunta 6

¿Qué dificultades presentan con el sistema actual de gestión de la contabilidad y emisión de facturas?

El error humano a la hora de llevar el control de los pedidos realizados, en ocasiones el pedido se realiza de forma incorrecta, puesto que por equivocación se preparó un plato que no era el que correspondía.

Pregunta 7

¿Considera que un sistema de gestión contable y de facturación mejore el proceso actual?

Sí, aunque se realicen procesos básicos en aspectos contables, tener un sistema que me muestre el registro o reportes relacionados a pagos realizados, o control de clientes, proveedores y demás es una ventaja.

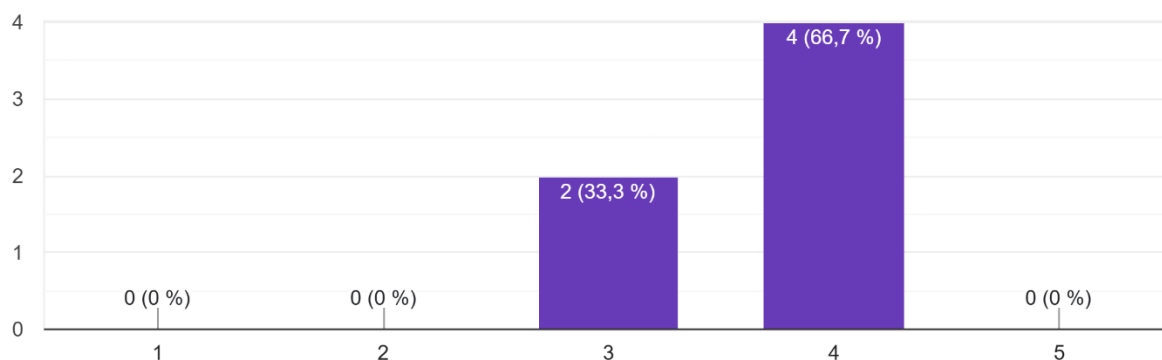
Cuestionario

Figura 7

Pregunta 1

En una escala del 1 al 5, ¿considera efectivo el método actual en que la empresa realiza la gestión contable?

6 respuestas



Fuente: Elaboración propia

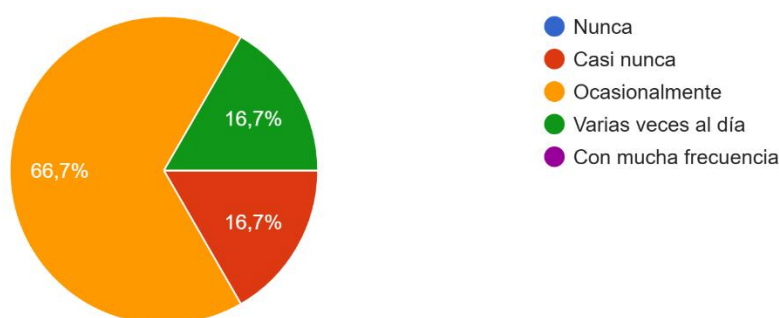
La figura 7 muestra que los encuestados consideran en su mayoría, con un 66,7% que el método de realizar la gestión contable es efectivo, mientras que el 33.3% lo considera moderado.

Esto refleja que la mayoría de los colaboradores consideran que actualmente la empresa cuenta con un método efectivo para realizar la gestión contable y en menor medida lo consideran moderado, por lo que el proceso de gestión contable, aunque en su mayoría se considera efectivo, cuenta con un margen de mejora.

Figura 8
Pregunta 2

¿Qué tan frecuente se generan facturas en la empresa?

6 respuestas



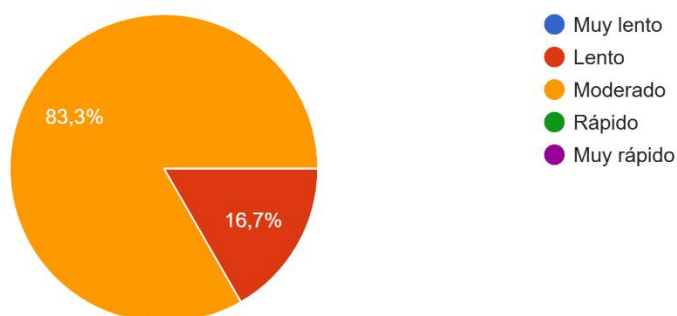
Fuente: Elaboración propia

La figura 8 muestra que un 66,7% de los colaboradores indican que las facturas se generan ocasionalmente en la empresa, mientras que un 16,7% indica que se generan varias veces al día y un 16,7% indica que se generan casi nunca. Esto evidencia que la mayoría de los colaboradores perciben un volumen medio de generación de facturas, sin embargo, se evidencia que no es seguro tener una respuesta certera, ya que hay días con mayor volumen que otros.

Figura 9
Pregunta 3

¿Cómo considera el tiempo que se toma en realizar una factura?

6 respuestas



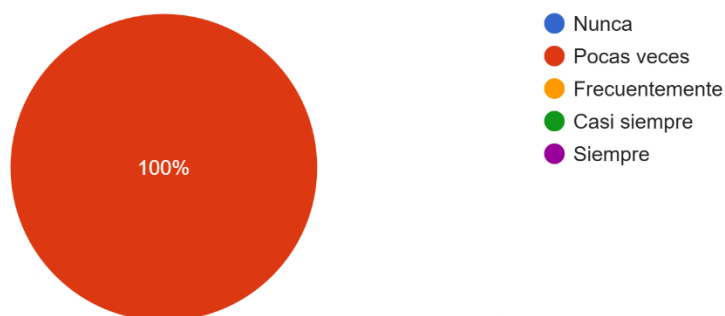
Fuente: Elaboración propia

De acuerdo con la figura 3, se muestra que el 83,3% de los colaboradores consideran que el tiempo que se toma en realizar una factura es moderado, mientras que el 16,7% lo considera lento. Esto refleja que la gran mayoría de los colaboradores percibe que el tiempo empleado en realizar una factura se concreta de forma moderada, lo que permite aplicar un sistema que agilice este proceso.

Figura 10
Pregunta 4

¿Con qué frecuencia se presentan errores en el proceso de facturación?

6 respuestas



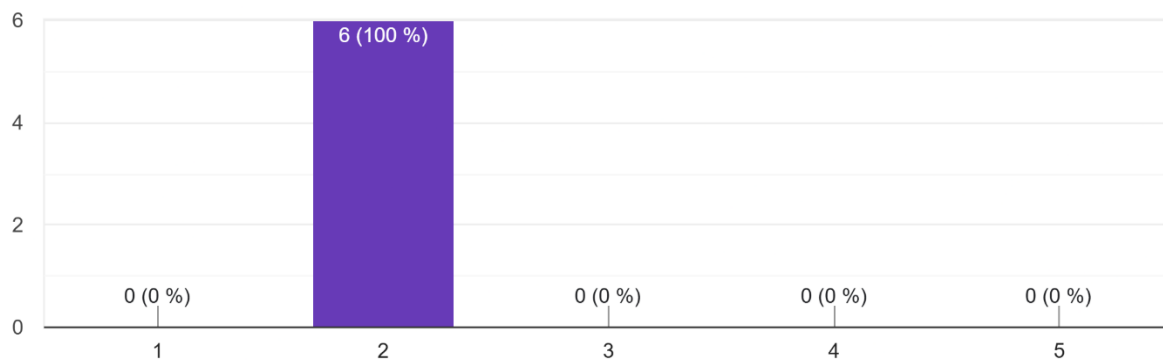
Fuente: Elaboración propia

Con respecto a la pregunta 4, se muestra que el 100% de los colaboradores concuerdan y consideran que la frecuencia en que se presentan errores en el proceso de facturación es poco; lo que demuestra que el proceso actual de facturación es efectivo.

Figura 11
Pregunta 5

En una escala del 1 al 5, ¿considera efectivo el método de gestionar pedidos por parte de los clientes?

6 respuestas



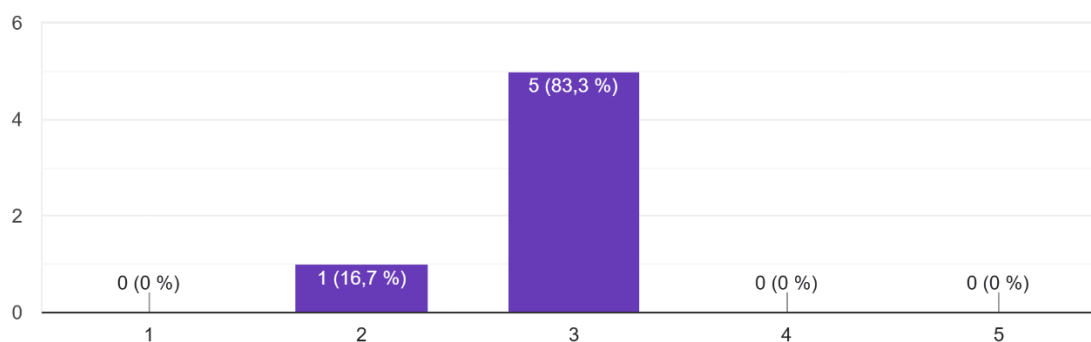
Fuente: Elaboración propia

La figura 11 muestra que el 100% de los colaboradores considera que el método cómo se gestionan los pedidos de los clientes es poco efectivo, mostrando un descontento con el procedimiento actual de la empresa para esta labor.

Figura 12
Pregunta 6

En una escala del 1 al 5, ¿considera efectiva la gestión de pedidos a los proveedores?

6 respuestas



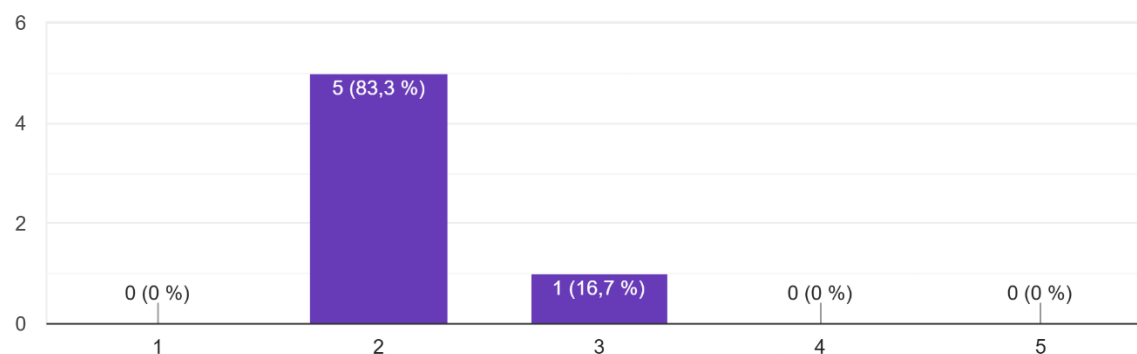
Fuente: Elaboración propia

La pregunta 6 evidencia que el 83,3% de los colaboradores consideran que la gestión de los pedidos a los proveedores se encuentra en un nivel moderado, mientras que el 16,67% de los colaboradores, lo considera poco efectivo, demostrando que este proceso no es efectivo.

Figura 13
Pregunta 7

En una escala del 1 al 5, ¿considera efectiva la gestión de las cuentas por cobrar a los clientes?

6 respuestas



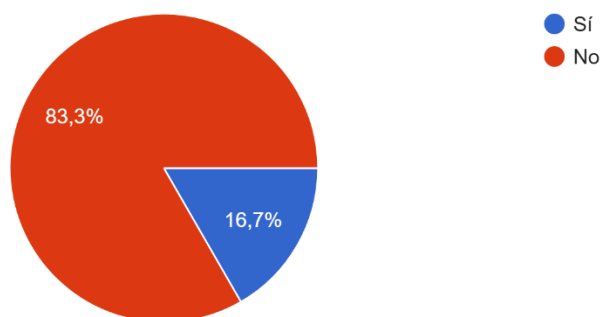
Fuente: Elaboración propia

La figura 13 muestra que el 83,5% de los colaboradores considera que el método de gestión de cuentas por cobrar es poco efectivo, mientras que el 16,67% de los colaboradores lo considera moderado. Se demuestra que la mayoría de los colaboradores no consideran efectivo el proceso de gestión de cuentas por cobrar.

Figura 14
Pregunta 8

¿Se encuentra satisfecho/a con el sistema actual en que opera la empresa las gestiones contables y de facturación?

6 respuestas



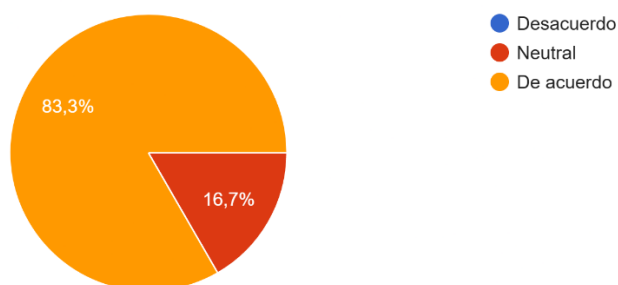
Fuente: Elaboración propia

La figura 14 muestra que el 83,3% de los colaboradores no se encuentran satisfecho con el proceso actual de la empresa en cuanto a gestiones contables y de facturación, mientras que el 16,7% se encuentra satisfecho. Esto demuestra que la gran mayoría de colaboradores no se encuentran satisfechos con este proceso.

Figura 15
Pregunta 9

¿Considera que un sistema automatizado sería de ayuda para reducir errores en el proceso de facturación?

6 respuestas



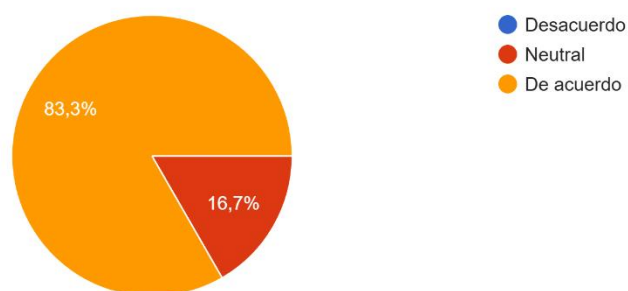
Fuente: Elaboración propia

La figura 15 demuestra que la gran mayoría de colaboradores, con el 83,3% está de acuerdo con la implementación de un sistema automatizado para reducir errores en el proceso de facturación, mientras que el 16,7% toma una posición neutral.

Figura 16
Pregunta 10

¿Considera que un sistema automatizado mejoraría la gestión contable de la empresa?

6 respuestas



Fuente: Elaboración propia

De acuerdo con la figura 16, se evidencia que el 83,3% de los colaboradores considera que un sistema automatizado mejoraría la gestión contable de la empresa, mientras que el 16,7% mantiene una posición neutral al respecto.

CAPÍTULO V: PROPUESTA

Análisis detallado del Software por desarrollar

El sistema de gestión contable y de facturación tiene como propósito solventar las necesidades que enfrenta la empresa, para ello se proponen módulos específicos que se encarguen de solventar estos requerimientos. A continuación, se describen, de forma detallada, los módulos que conforman el sistema:

Módulo de gestión de cuentas contables

El módulo para gestionar cuentas contables es un módulo que tiene como función principal definir y estructurar el catálogo contable de la soda, de esta forma el usuario, utilizando el sistema, podrá clasificar los activos, pasivos, ingresos y gastos, lo que permitirá la generación de asientos contables.

El módulo efectuará la gestión de cuentas, por lo que, el usuario podrá crear nuevas cuentas contables, desactivarlas, en caso de ser necesario, y mediante la integración de los módulos de facturación y de ingresos y gastos, podrá visualizar de manera sencilla la información respectiva de las distintas cuentas activas.

Módulo de administración de cuentas por pagar

Llevar el correcto manejo de las cuentas por pagar es una problemática que ha enfrentado la empresa, debido al alto nivel de flujo de pedidos, por lo que el módulo para administrar cuentas por pagar se creará con las características necesarias para llevar el control de los montos pendientes por pagar a los proveedores. El usuario podrá visualizar de forma detallada la totalidad de cuentas por pagar, así como su detalle, ya que podrá registrar la deuda mostrando la totalidad por pagar, si el pago es a contado o a crédito, y su respectiva factura.

Módulo de gestión de facturación

El módulo para gestionar la facturación toma en cuenta los pedidos realizados y genera una factura, mostrando el producto y su respectivo detalle. El usuario podrá gestionar si la compra es a crédito o a contado; en caso de ser a crédito, el sistema genera una cuenta por cobrar, integrando el módulo específico de las cuentas por cobrar y el usuario podrá visualizar el listado de estas cuentas.

Módulo de gestión de ingresos

El usuario, mediante el módulo de gestionar ingresos, podrá clasificar y visualizar los ingresos que recibe de parte del módulo de facturación, cuentas por cobrar o cualquier tipo de ingreso extraordinario. Este módulo contará con una integración directa con el módulo de gestión de cuentas contables, en donde los ingresos estarán clasificados de acuerdo con su categoría, y el usuario podrá visualizar el detalle respectivo.

Módulo de gestión de gastos

El módulo de gestión de gastos registra y clasifica los gastos de la empresa, ya sean operativos, administrativos o de mantenimiento. El sistema facilitará esta clasificación y visualización ya que, al igual que el módulo de ingresos, se integrará con el módulo de gestión de cuentas contables y con el módulo de presupuesto mensual. Esta integración facilitará el proceso de toma de decisiones de la empresa al mostrar reportes de egresos y se podrá comparar con el gasto presupuestado mensual.

Módulo de administración de cuentas por cobrar

La correcta administración de las cuentas por cobrar es fundamental en un sistema contable, y la empresa presenta dificultades para llevar el control de las cuentas por cobrar, por lo que el módulo registrará y permitirá dar seguimiento a los pagos pendientes por parte de los clientes. El sistema registrará los abonos realizados o el pago total, también estará integrado con el módulo de Ingresos, actualizando la clasificación de ingresos al momento de recibir pagos de cuentas por cobrar.

Módulo de generación del presupuesto mensual

El módulo para generar el presupuesto mensual permitirá al usuario crear, controlar y ajustar el presupuesto mensual de la empresa. Los datos registrados en el módulo son datos provenientes del módulo de gestión de cuentas contables y permitirá al usuario comparar los datos reales contra los datos planificados al iniciar el mes, mostrando gráficos de desempeño y facilitando el proceso de toma de decisiones.

Análisis detallado del hardware requerido

En la etapa de desarrollo del prototipo se requiere de un hardware que cumpla con las características mínimas para su correcto desempeño, es necesario un equipo que permita compilar código, simular escenarios reales y que cuente con la robustez para soportar pruebas. Las características del equipo utilizado se describen en la tabla 8, a continuación:

Tabla 8

Hardware utilizado

Componente	Característica
Procesador	Intel Core i7-9750H
Memoria RAM	16 GB
Almacenamiento	512 GB
Sistema Operativo	Windows 11
Conectividad	100 Mbps

Fuente: Elaboración propia

Los datos descritos en la tabla 8 son las especificaciones del equipo utilizado en la fase de desarrollo del prototipo, no corresponden a las especificaciones mínimas requeridas para su uso. El procesador utilizado ofrece un rendimiento óptimo para el desarrollo de código, compilación y pruebas. El costo estimado es de 24 000 colones aproximadamente. Para un rendimiento fluido se recomienda 16 GB de memoria RAM, con un costo de 70 000 colones.

El almacenamiento se utiliza un SSD de 512 GB, el cual es ideal para el desarrollo del prototipo, sin embargo, no es el mínimo, puesto que, con un almacenamiento de 256 GB es suficiente. El costo estimado oscila entre los 41 500 y 60 000 colones. En cuanto a la conectividad, la conexión de 100 Mbps es recomendable para contar con estabilidad y el precio depende del plan contratado, puede rondar entre los 15 000 colones a 25 000 colones mensuales.

La implementación en ambiente productivo se realizará en el equipo local de la empresa, la cual ya cuenta con el equipo necesario y la conectividad para ejecutar el prototipo. Estas características se visualizan en la tabla 1.

Análisis detallado de telecomunicaciones

Tipo de red

Para acceder al sistema se requiere de una red local, que permita la comunicación entre el equipo del usuario con el servidor de la aplicación. El acceso se realizará mediante el protocolo HTTPS utilizando el navegador web de usuario.

Uso de internet

Es requerida una conectividad estable, para asegurar el óptimo funcionamiento de la aplicación. Es recomendable una conexión de al menos 100 Mbps.

Puertos y protocolos

Se utilizará el puerto 443, el cual es el puerto estándar HTTPS para el tráfico web seguro. Se utilizará un cifrado SSL/TLS, esto con el objetivo de proteger la información entre el navegador y el servidor.

Equipos especializados

La empresa ya cuenta con el equipo requerido para acceder al sistema y un enrutador que administra la conectividad de red.

Herramientas técnicas utilizadas para el desarrollo

Lenguaje de programación

El lenguaje de programación utilizado es C# junto al framework Blazor, que permiten desarrollar aplicaciones dinámicas e interactivas con la arquitectura de Modelo-Vista-Controlador. Cuenta con licenciamiento gratuito y estas herramientas ofrecen facilidades para conectarse al motor de base de datos.

Entorno de desarrollo integrado

El entorno de desarrollo integrado (IDE) utilizado es Visual Studio, el cual es una herramienta que facilita la programación y depuración del sistema. Se utiliza la versión Community, la cual posee licencia gratuita. Es un editor de código que permite trabajar con el

lenguaje de programación C#, el motor de base de datos SQL Server y cuenta con integración con Git, para el control de versiones del proyecto.

Motor de base de datos

Se utilizará el motor de base de datos de SQL Server Express para el almacenamiento de los datos de la información. La edición Express es gratuita y permite almacenar la información de manera segura y estructurada. Permite realizar transacciones seguras y cuenta con operaciones como procedimientos almacenados, funciones y *triggers*.

Control de versiones

Como herramienta de respaldo y gestión del código se utilizará la plataforma de GitHub, la cual facilita el control de versiones del proyecto, permitiendo la integridad y documentación de los avances realizados. Su licenciamiento es gratuito.

Conocimiento básico requerido para la operación del sistema

El usuario que opere el sistema no requiere contar con un conocimiento técnico avanzado en computación ya que el sistema ha sido diseñado de manera intuitiva y de fácil acceso, sin embargo, sí requiere tener con conocimiento básico en el uso de herramientas informáticas y procesos administrados.

El usuario debe poder utilizar una computadora, estar familiarizado con el sistema operativo y saber utilizar el navegador web de su preferencia, comprender cómo acceder a la aplicación utilizando una dirección en el navegador y el empleo de dispositivos de entrada como teclado y ratón.

Asimismo, el usuario deberá contar con conocimiento en procesos administrativos y contables, como el registro de cuentas contables, control de ingresos y gastos y manejo de cuentas por cobrar y pagar.

El usuario administrador debe contar con conocimiento básico de protección de información personal y confidencial y seguir los lineamientos legales del código penal costarricense sobre la protección de datos personales.

Casos de uso

Tabla 9
CU-01

Prototipo: Prototipo funcional para el desarrollo de un sistema de gestión contable y de facturación para la Soda Creston, ubicada en San José.	
Número Caso de Uso: CU-01	Nombre del caso de uso: Inicio de sesión.
Fecha elaboración:	22/04/2026.
Descripción Caso de Uso:	Permite al usuario ingresar al sistema, ingresando el usuario y contraseña.
Autor caso de uso:	Alexis Francisco Mora Mora.
Actores relacionados	Usuario del sistema.
Precondiciones	El usuario debe estar activo y con credenciales válidas.
Flujo Básico del caso de uso 1. El usuario ingresa a la pantalla de iniciar sesión. 2. Si el usuario olvidó la contraseña, dar clic en “¿Olvidó su contraseña?” SF-01. 3. El usuario digita en los campos respectivos el nombre de usuario y contraseña. 4. El usuario da clic en el botón de Ingresar. 5. El sistema valida la información ingresada por el usuario. FA-01. 6. El sistema inicia sesión y muestra la pantalla principal.	
Sub-Flujos	
SF-01 Recuperar contraseña	1. El usuario selecciona la opción “¿Olvidaste tu contraseña?” 2. El sistema le pedirá al usuario ingresar su correo electrónico registrado. FA-02. 3. El usuario ingresa el correo y recibe un correo para que el usuario modifique la contraseña.

4. El sistema modifica la contraseña anterior por la nueva digitada por el usuario.	
Flujos Alternos	
FA-01: Credenciales incorrectas.	El sistema valida si el usuario y la contraseña ingresados por el usuario no están correctos, rechazando el ingreso y pidiendo al usuario ingresar la información nuevamente
FA-02: Validación correo existente	El sistema valida si el correo ingresado por el usuario corresponde al correo existente en base de datos, en caso de que el correo sea incorrecto se mostrará un mensaje de error indicándole al usuario que el correo digitado no está asociado a su cuenta.
Requerimientos especiales	
Las contraseñas digitadas por los usuarios se reflejan como encriptadas a nivel de base de datos, protegiendo la información del usuario.	
Postcondiciones	
El usuario ingresa exitosamente al sistema.	

Fuente: Elaboración propia

Tabla 10
CU-02

Prototipo: Prototipo funcional para el desarrollo de un sistema de gestión contable y de facturación para la Soda Creston, ubicada en San José.	
Número Caso de Uso: CU-02	Nombre del caso de uso: Gestionar cuentas contables.
Fecha elaboración:	22/04/2026.
Descripción Caso de Uso:	Permite al usuario autorizado estructurar y consultar el catálogo contable de la empresa.

Autor caso de uso:	Alexis Francisco Mora Mora.
Actores relacionados	Usuario administrador.
Precondiciones	El usuario debe estar activo y autorizado y debe haber ingresado al sistema.
Flujo Básico del caso de uso 1. El usuario ingresa al módulo de Gestión de cuentas contables. 2. El sistema muestra el listado de las cuentas registradas. 3. El usuario selecciona la opción “Nueva Cuenta”. SF-01. 4. El usuario selecciona la opción de editar una cuenta registrada. SF-02. 5. El usuario elimina una cuenta contable registrada. SF.03.	
Sub-Flujos	
SF-01: Crear cuenta contable	1. El usuario selecciona la opción “Nueva Cuenta Contable”. 2. El sistema habilita un formulario para registrar la cuenta nueva. 3. El usuario ingresa los datos requeridos. 4. El usuario da clic en el botón “Guardar”. 5. El sistema valida la información ingresada por el usuario. FA-01.
SF-02: Editar cuenta contable	1. El usuario da clic a la opción de editar de una cuenta registrada. 2. El sistema habilita el formulario de la cuenta seleccionada, con opción de editar. 3. El usuario modifica la información requerida. 4. El sistema valida los cambios realizados. FA-01.

SF-03: Eliminar cuenta contable	5. El usuario da clic en el botón de Guardar para registrar el cambio del registro que seleccionó.
	1. El usuario selecciona la opción de eliminar una cuenta contable. 2. El sistema muestra un mensaje de advertencia indicándole al usuario si está seguro de eliminar la cuenta seleccionada. 3. El usuario confirma la eliminación del registro. 4. La cuenta se elimina de base de datos y el usuario no podrá visualizarla.
Flujos Alternos	
FA-01: Datos incorrectos	El sistema valida si los datos ingresados por el usuario están incorrectos, en caso de que alguno no respete el tipo de dato o sea inválido, el sistema le indicará al usuario que la información ingresada es incorrecta y no dejará guardar el registro.
Requerimientos especiales	
El sistema valida que este módulo se muestra únicamente al usuario administrador. El sistema no permitirá la modificación o ingreso de datos no válidos o incorrectos.	
Postcondiciones	
El usuario podrá catalogar sus cuentas contables y gestionar de acuerdo con la clasificación y permitirá controlar las cuentas activas y no activas.	

Fuente: Elaboración propia

Tabla 11
CU-03

Prototipo: Prototipo funcional para el desarrollo de un sistema de gestión contable y de facturación para la Soda Creston, ubicada en San José.	
Número Caso de Uso: CU-03	Nombre del caso de uso: Administrar cuentas por pagar.
Fecha elaboración:	22/04/2026.
Descripción Caso de Uso:	Permite al usuario autorizado registrar y visualizar las cuentas por pagar a los proveedores.
Autor caso de uso:	Alexis Francisco Mora Mora.
Actores relacionados	Usuario administrador.
Precondiciones	El usuario debe estar activo y autorizado y debe haber ingresado al sistema. El proveedor debe estar registrado en el sistema. El usuario debe haber ingresado al módulo de administrar cuentas por pagar.
Flujo Básico del caso de uso 1. El sistema muestra la información de las cuentas por pagar a los proveedores con su respectiva fecha de vencimiento. 2. El usuario selecciona la opción de “Nueva Cuenta por Pagar”. 3. El sistema habilita un formulario con la información requerida para registrar el pago por realizar al proveedor específico. FA-01. 4. El usuario completa la información, seleccionando el modo de pago, y registrando la cuenta contable. SF-01. 5. El sistema guarda la información y muestra el registro actualizado. 6. El usuario podrá realizar abonos a la cuenta por pagar o pagar la totalidad del saldo pendiente. SF-02.	

Sub-Flujos

SF-01: Medio de pago

1. El usuario al ingresar al registro del pago por realizar deberá seleccionar el medio de pago.
2. El sistema habilita los campos requeridos de acuerdo con el medio de pago, si es efectivo, tarjeta o por transferencia.

SF-02 Realizar abonos

1. El usuario selecciona la opción de “Abonar”.
2. El sistema le muestra el formulario de la cuenta seleccionada, con el dato del saldo pendiente, y un campo para digitar el monto del abono.
3. El usuario ingresa el monto del abono.
4. Si el monto ingresado es menor al saldo pendiente, la cuenta por pagar tendrá un estado “Pendiente” con el saldo actualizado.
5. Si el monto del abono es la totalidad del saldo pendiente, la cuenta por pagar tendrá el estado de “Pagada”.
6. El sistema valida que el monto ingresado sea válido. FA-01.

Flujos Alternos

FA-01: Validar registro de abono

En caso de que el usuario digite un monto a abonar superior al saldo pendiente, el sistema mostrará una alerta indicando que el monto

	ingresado no es válido y debe ser menor o igual al saldo pendiente.
Requerimientos especiales	
Al realizar pagos pendientes, estos pagos se le sumará el IVA respectivo, el cálculo se hará de forma automática, mostrando el valor por pagar con el IVA incluido.	
Postcondiciones	
El usuario podrá llevar el control de las cuentas por pagar con los proveedores.	

Fuente: Elaboración propia

Tabla 12
CU-04

Prototipo: Prototipo funcional para el desarrollo de un sistema de gestión contable y de facturación para la Soda Creston, ubicada en San José.	
Número Caso de Uso: CU-04	Nombre del caso de uso: Gestión de facturación.
Fecha elaboración:	22/04/2026.
Descripción Caso de Uso:	Crea la factura, así como el detalle, tomando en cuenta los pedidos realizados por parte de los clientes.
Autor caso de uso:	Alexis Francisco Mora Mora.
Actores relacionados	Usuario de caja.
Precondiciones	El usuario debe estar activo y autorizado y debe haber ingresado al sistema.
Flujo Básico del caso de uso 1. El usuario ingresa al módulo de facturación. 2. El sistema muestra la pantalla principal de creación de factura y visualización de registro de facturas. 3. El usuario ingresa en la opción de “Nueva factura” SF-01. 4. El usuario podrá exportar la información de las facturas.	

Sub-Flujos	
SF-01: Crear factura	1. El sistema muestra el formulario de creación de factura. 2. El usuario llena la información respectiva. 3. El usuario indica si la factura se encuentra en estado de “Pagada” o “Pendiente” SF-02. 4. El sistema valida la información ingresada. FA-01. 5. El sistema calcula automáticamente el total de la factura. SF-03. 6. Las facturas pagadas, se integran en el módulo de “Ingresos”.
SF-02: Factura pendiente	1. Al estar con estado de “Pendiente”, el sistema genera un nuevo registro en el módulo de Cuentas por Cobrar. 2. En la columna de “Acciones” el usuario puede imprimir la factura, anularla o colocar el estado de “Pagada”.
SF-03: Cálculo total	1. El sistema registra el subtotal ingresado por el usuario. 2. El sistema añade el valor respectivo al impuesto y muestra el total.
Flujos Alternos	
FA-01: Validación de información	El sistema valida la información digitada por el usuario, respetando el tipo de dato del campo, si el usuario ingresa un tipo de dato que no corresponde al del campo, el sistema indicará

	mediante un mensaje de error, que el valor no es válido.
Requerimientos especiales	
No hay	
Postcondiciones	
El usuario genera la factura.	

Fuente: Elaboración propia

Tabla 13

CU-05

Prototipo: Prototipo funcional para el desarrollo de un sistema de gestión contable y de facturación para la Soda Creston, ubicada en San José.	
Número Caso de Uso: CU-05	Nombre del caso de uso: Gestionar ingresos.
Fecha elaboración:	22/04/2026.
Descripción Caso de Uso:	Registra los ingresos provenientes del módulo de facturación y cuentas por cobrar.
Autor caso de uso:	Alexis Francisco Mora Mora.
Actores relacionados	Usuario administrador, Usuario de caja.
Precondiciones	El usuario debe estar activo y autorizado y debe haber ingresado al sistema.
Flujo Básico del caso de uso	
1. El usuario ingresa al módulo de Ingresos. 2. El sistema muestra el registro de los ingresos, proveniente de los módulos de facturación y de cuentas por cobrar. 3. El usuario da clic en el botón de registrar “Nuevo Ingreso”. SF-01. 4. El sistema permite conciliar el ingreso o eliminarlo.	
Sub-Flujos	
SF-01: Registro ingreso	1. El sistema muestra el formulario de registrar un nuevo ingreso.

2. El usuario digita la información requerida para ingresar un nuevo registro. 3. El sistema valida la información ingresada por el usuario. FA-01.	
Flujos Alternos	
FA-01: Validación de información	El sistema valida la información digitada por el usuario, respetando el tipo de dato del campo, si el usuario ingresa un tipo de dato que no corresponde al del campo, el sistema indicará mediante un mensaje de error que el valor no es válido. El sistema valida que la información que el usuario selecciona en la cuenta del débito no sea la misma que la cuenta de crédito, siguiendo los principios de contabilidad.
Requerimientos especiales	
No hay.	
Postcondiciones	
El usuario visualiza los ingresos de parte de los módulos de facturación y cuentas por cobrar y puede registrar nuevos ingresos extraordinarios.	

Fuente: Elaboración propia

Tabla 14

CU-06

Prototipo: Prototipo funcional para el desarrollo de un sistema de gestión contable y de facturación para la Soda Creston, ubicada en San José.	
Número Caso de Uso: CU-06	Nombre del caso de uso: Gestionar gastos.
Fecha elaboración:	22/04/2026.

Descripción Caso de Uso:	Registra los gastos directos de la empresa y permite visualizar la clasificación de gastos en que incurre la empresa.
Autor caso de uso:	Alexis Francisco Mora Mora.
Actores relacionados	Usuario administrador, Usuario de caja.
Precondiciones	El usuario debe estar activo y autorizado y debe haber ingresado al sistema.
Flujo Básico del caso de uso 1. El usuario ingresa al módulo de Gastos. 2. El sistema muestra la totalidad de gastos registrados. 3. El usuario filtra los gastos por categoría. 4. El sistema muestra los registros de acuerdo con el filtro seleccionado SF-01. 5. El sistema actualiza la lista de gastos.	
Sub-Flujos	
SF-01: Categoría de gastos	1. El usuario ingresa en la categoría de gastos que desea visualizar. 2. El sistema muestra el registro de los tipos de gastos en que incurre la empresa de acuerdo con la categoría seleccionada. 3. El usuario registra un nuevo registro dando clic en el botón “Nuevo Gasto”. SF-02.
SF-02: Añadir gastos	1. El sistema muestra un formulario con los datos requeridos para llenar la información del gasto. 2. El usuario llena la información respectiva y guarda.

	3. El sistema valida la información digitada. FA-01. 4. El sistema guarda la información y permite visualizarla como un nuevo registro.
Flujos Alternos	
FA-01: Validación de información	El sistema valida la información digitada por el usuario, respetando el tipo de dato del campo; si el usuario ingresa un tipo de dato que no corresponde al del campo, el sistema indicará mediante un mensaje de error que el valor no es válido.
Requerimientos especiales	
No hay.	
Postcondiciones	
El usuario visualiza, por categorías, los tipos de gastos y permite ingresar nuevos registros.	

Fuente: Elaboración propia

Tabla 15

CU-07

Prototipo: Prototipo funcional para el desarrollo de un sistema de gestión contable y de facturación para la Soda Creston, ubicada en San José.	
Número Caso de Uso: CU-07	Nombre del caso de uso: Administrar cuentas por cobrar.
Fecha elaboración:	22/04/2026.
Descripción Caso de Uso:	Muestra el registro y da seguimiento de los pagos pendientes por parte de los clientes.
Autor caso de uso:	Alexis Francisco Mora Mora.
Actores relacionados	Usuario administrador, Usuario de caja.
Precondiciones	El usuario debe estar activo y autorizado y debe haber ingresado al sistema.

	El cliente debe estar registrado. Debe existir una factura o venta asociada. Debe existir al menos una cuenta contable.
Flujo Básico del caso de uso 1. El usuario ingresa al módulo de Cuentas por cobrar. 2. El sistema muestra la totalidad de cuentas por cobrar. 3. El usuario da clic en “Nueva Cuenta por Cobrar” SF-01. 4. El usuario realiza un abono a la cuenta por cobrar SF-02. 5. El sistema registra las cuentas por cobrar con el saldo pendiente. 6. El sistema actualiza la totalidad de cuentas por cobrar.	
Sub-Flujos	
SF-01: Registrar cuenta por cobrar	1. El sistema muestra un formulario con los datos por rellenar para registrar la cuenta. 2. El usuario ingresa los datos requeridos, la factura, el tipo de pago, la cuenta contable y el saldo pendiente FA-01. 3. El usuario guarda la información. 4. El sistema valida la información FA-02. 5. El sistema guarda la información.
SF-02: Registrar pago	1. El usuario le da clic al botón de “Registrar Pago” de una cuenta por cobrar. 2. El sistema muestra un formulario para ingresar el monto a pagar. FA-01 3. El sistema valida la información. FA-02. 4. El sistema guarda la información.

5. Al realizar un abono, el sistema realiza el cálculo automático del saldo pendiente.	
Flujos Alternos	
FA-01: Pago mayor al saldo de la cuenta.	El sistema valida si el monto digitado por el usuario excede el saldo de la cuenta por cobrar, en caso de sobrepasarse, el sistema muestra un mensaje indicando que el monto ingresado excede el saldo pendiente de la cuenta seleccionada.
FA-02: Validación de información	El sistema valida la información digitada por el usuario, respetando el tipo de dato del campo, si el usuario ingresa un tipo de dato que no corresponde al del campo, el sistema indicará mediante un mensaje de error que el valor no es válido.
Requerimientos especiales	
No hay.	
Postcondiciones	
El usuario visualiza las cuentas por cobrar registradas y actualizadas, manteniendo el control de los saldos pendientes.	

Fuente: Elaboración propia

Tabla 16

CU-08

Prototipo: Prototipo funcional para el desarrollo de un sistema de gestión contable y de facturación para la Soda Creston, ubicada en San José.	
Número Caso de Uso: CU-08	Nombre del caso de uso: Generar presupuesto mensual.
Fecha elaboración:	22/04/2026.

Descripción Caso de Uso:	Permite al usuario autorizado estructurar y ajustar el presupuesto mensual.
Autor caso de uso:	Alexis Francisco Mora Mora.
Actores relacionados	Usuario administrador.
Precondiciones	El usuario debe estar activo, autorizado y debe haber ingresado al sistema. Deben existir cuentas registradas.
Flujo Básico del caso de uso 1. El usuario ingresa al módulo de generar presupuesto mensual. 2. El sistema muestra los presupuestos existentes, en caso de haber. 3. El usuario selecciona el presupuesto que desea visualizar. 4. El sistema muestra la totalidad de ingresos y gastos del mes, con el balance actualizado. 5. El usuario da clic en “Comparar” SF-01. 6. El usuario selecciona la opción “Nuevo Presupuesto” SF-02. 7. El usuario desactiva el presupuesto SF-03.	
Sub-Flujos	
SF-01: Comparar presupuesto	1. El usuario ingresa al detalle de la comparación del presupuesto. 2. El sistema muestra la comparación del presupuesto real, contra el presupuestado inicialmente. 3. El sistema realiza el cálculo automático de la diferencia entre ambos presupuestos. 4. El sistema, mediante gráficos y alertas visuales, muestra el detalle al usuario.
SF-01: Crear presupuesto	1. El sistema muestra un formulario para el llenado de información. 2. El usuario ingresa los datos relacionados al presupuesto mensual.

SF-02: Desactivar presupuesto	3. El sistema valida la información FA-01.
	1. El usuario selecciona la opción de desactivar el presupuesto en la columna de “Acciones”.
	2. El presupuesto pasa a un estado de “Inactivo”.
Flujos Alternos	
FA-01: Validación de información	El sistema valida la información digitada por el usuario, respetando el tipo de dato del campo; si el usuario ingresa un tipo de dato que no corresponde al del campo, el sistema indicará mediante un mensaje de error que el valor no es válido.
Requerimientos especiales	
No hay.	
Postcondiciones	
Se registra el presupuesto mensual, permitiendo realizar su debido análisis financiero.	

Fuente: Elaboración propia

Tabla 17
CU-09

Prototipo: Prototipo funcional para el desarrollo de un sistema de gestión contable y de facturación para la Soda Creston, ubicada en San José.	
Número Caso de Uso: CU-09	Nombre del caso de uso: Realizar mantenimientos.
Fecha elaboración:	22/04/2026.
Descripción Caso de Uso:	Permite al usuario autorizado insertar, modificar, actualizar y eliminar datos.
Autor caso de uso:	Alexis Francisco Mora Mora.
Actores relacionados	Usuario administrador.

Precondiciones	El usuario debe estar activo y autorizado, y debe haber ingresado al sistema.
Flujo Básico del caso de uso	
1. El usuario administrador ingresa al módulo de Mantenimientos. 2. El sistema da la opción de crear, editar o eliminar proveedores, clientes, usuarios y productos. 3. El sistema valida la información ingresada por el usuario FA-01. 4. El sistema guarda la información.	
Sub-Flujos	
No hay.	
Flujos Alternos	
FA-01: Validación de información	El sistema valida la información digitada por el usuario, respetando el tipo de dato del campo, si el usuario ingresa un tipo de dato que no corresponde al del campo, el sistema indicará mediante un mensaje de error que el valor no es válido.
Requerimientos especiales	
No hay.	
Postcondiciones	
Mantenimiento de datos registrados en el sistema.	

Fuente: Elaboración propia

Tabla 18
CU-10

Prototipo: Prototipo funcional para el desarrollo de un sistema de gestión contable y de facturación para la Soda Creston, ubicada en San José.	
Número Caso de Uso: CU-10	Nombre del caso de uso: Realizar consultas.
Fecha elaboración:	22/04/2026.

Descripción Caso de Uso:	Permite al usuario consultar información proveniente de los distintos módulos.
Autor caso de uso:	Alexis Francisco Mora Mora.
Actores relacionados	Usuario administrador, Usuario general.
Precondiciones	El usuario debe estar activo, autorizado y debe haber ingresado al sistema.
Flujo Básico del caso de uso 1. El usuario ingresa al módulo de Consultas. 2. El sistema muestra las opciones de consulta, entre las que se encuentran las consultas de facturas, asientos contables, cuentas, libro diario, mayor contable y la bitácora. 3. El usuario selecciona la consulta deseada. 4. El usuario contará con la opción de filtrar dentro de la consulta seleccionada. 5. El sistema mostrará la información de acuerdo con el filtro seleccionado por el usuario.	
Sub-Flujos No hay.	
Flujos Alternos	
FA-01: No encontró resultados	El sistema no muestra registros de acuerdo con los filtros señalados por el usuario, por lo que se muestra que no se encontraron resultados.
Requerimientos especiales	
No hay	
Postcondiciones	
El usuario puede visualizar la información específica del módulo o módulos consultados.	

Fuente: Elaboración propia

Tabla 19
CU-11

Prototipo: Prototipo funcional para el desarrollo de un sistema de gestión contable y de facturación para la Soda Creston, ubicada en San José.	
Número Caso de Uso: CU-11	Nombre del caso de uso: Generar reportes.

Fecha elaboración:	22/04/2026.
Descripción Caso de Uso:	Genera reportes de información.
Autor caso de uso:	Alexis Francisco Mora Mora.
Actores relacionados	Usuario administrador.
Precondiciones	El usuario debe estar activo y autorizado, y debe haber ingresado al sistema.
Flujo Básico del caso de uso	
1. El usuario ingresa al módulo de reportes. 2. El sistema mostrará las opciones disponibles para la generación de reportes. 3. El usuario podrá generar reportes de Ingresos, Gastos, presupuesto y balance general. 4. El sistema muestra la información FA-01. 5. El sistema permite la opción de descargar el PDF o de imprimir cada reporte seleccionado.	
Sub-Flujos	
Flujos Alternos	
FA-01: No encontró resultados	El sistema no muestra registros de acuerdo con los filtros señalados por el usuario, por lo que se muestra que no se encontraron resultados.
Requerimientos especiales	
No hay	
Postcondiciones	
Se generan reportes de información.	

Fuente: Elaboración propia

Tabla 20
CU-12

Prototipo: Prototipo funcional para el desarrollo de un sistema de gestión contable y de facturación para la Soda Creston, ubicada en San José.
--

Número Caso de Uso: CU-12	Nombre del caso de uso: Administrar seguridad.
Fecha elaboración:	22/04/2026.
Descripción Caso de Uso:	Definición de roles y autenticación de perfiles.
Autor caso de uso:	Alexis Francisco Mora Mora.
Actores relacionados	Usuario administrador.
Precondiciones	El usuario debe estar activo y autorizado y debe haber ingresado al sistema.
Flujo Básico del caso de uso 1. El usuario administrador ingresa al mantenimiento de usuarios. 2. El sistema proporciona opciones para crear usuarios SF-01, editar usuarios SF-02 o eliminar usuarios SF-03. 3. El sistema muestra los usuarios con sus respectivos roles. 4. El usuario gestiona los roles de los demás usuarios. 5. El sistema guarda los cambios realizados.	
Sub-Flujos	
SF-01: Crear usuarios.	1. El sistema muestra un formulario para crear un nuevo usuario. 2. El usuario completa la información, especificando el nombre de usuario y su respectivo rol.
SF-02: Editar usuarios.	1. El sistema permite al usuario modificar la información del usuario FA-01.
SF-03: Eliminar usuarios.	1. El sistema muestra la totalidad de usuarios registrados. 2. El usuario selecciona el usuario que desea eliminar. 3. El sistema muestra el estado del usuario seleccionado como “Inactivo”.

y el usuario no podrá ingresar al sistema.	
Flujos Alternos	
FA-01: Información incorrecta	El sistema valida si la información digitada por el usuario corresponde a la información requerida para la acción que desea realizar.
Requerimientos especiales	
No hay.	
Postcondiciones	
Roles aplicados y creación, edición y eliminación de usuarios.	

Fuente: Elaboración propia.

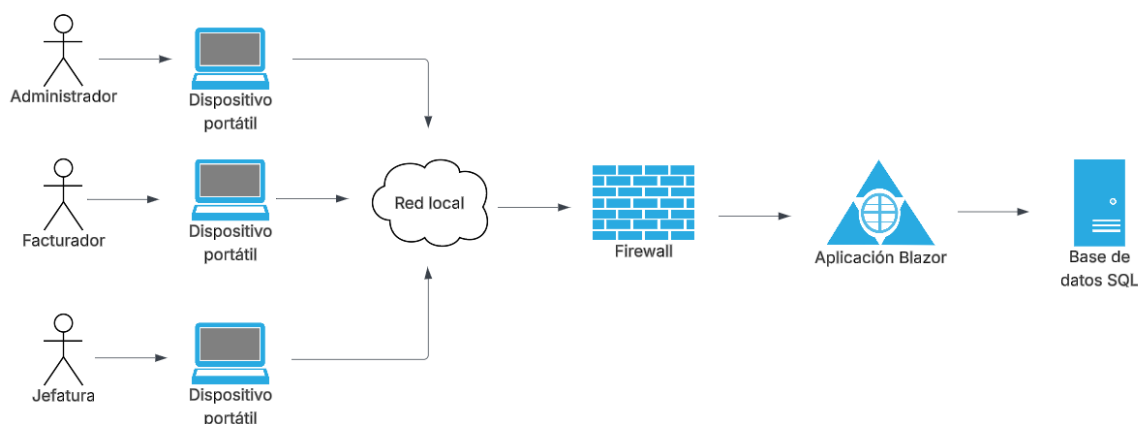
Diseño

En la sección de diseño, se presenta el diseño del software del sistema de gestión contable y de facturación, mostrando elementos como la arquitectura del sistema, la cual corresponde a una vista del prototipo a nivel general, la arquitectura del software, el diseño de entradas, el diseño físico de la base de datos, el diseño de procesos y el diseño de salidas y diagramas UML. Estos elementos representan una vista de la lógica del software, así como de los distintos elementos que lo componen y su interacción.

Arquitectura del sistema

A continuación, se presenta la figura correspondiente a la vista del prototipo a nivel general dentro de la infraestructura del cliente.

Figura 17
Arquitectura del sistema



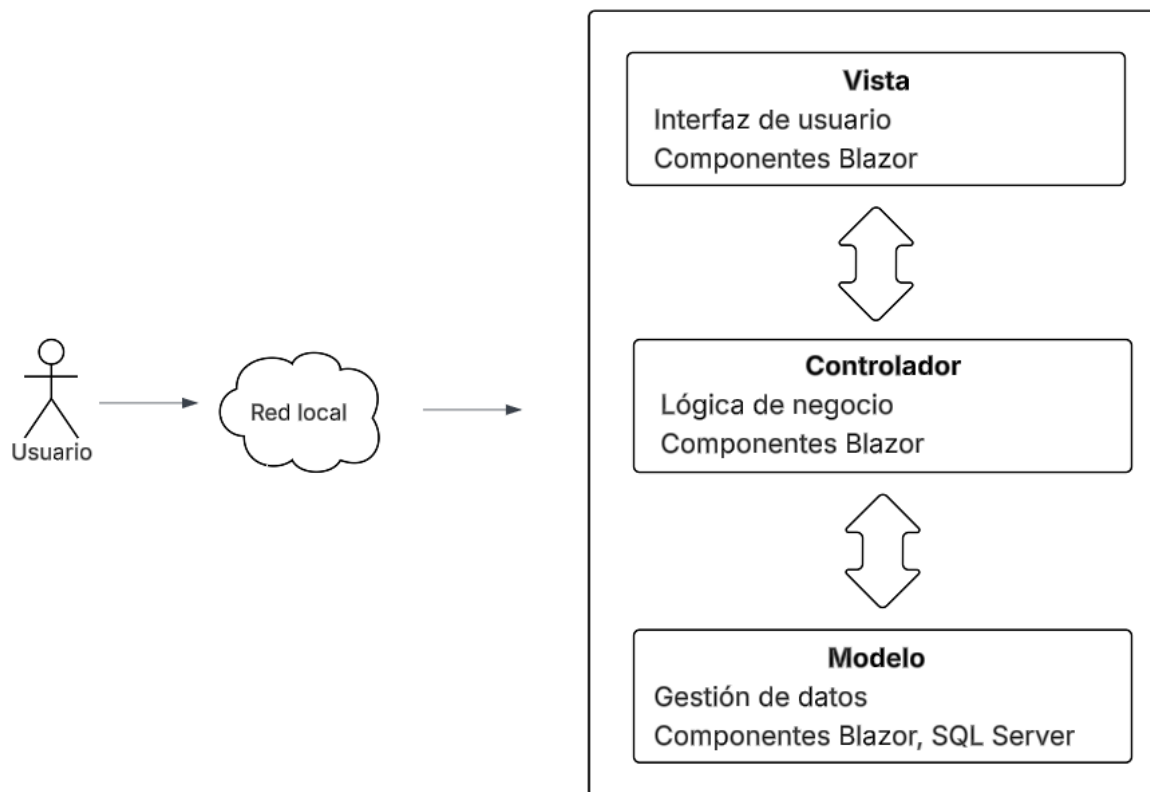
Fuente: Elaboración propia

La figura 7 representa la arquitectura del sistema, un modelo cliente-servidor, en el que los usuarios de roles de administrador, facturador y jefatura utilizan sus dispositivos portátiles personales para que, mediante la red local y un firewall como mecanismo de seguridad, puedan conectarse a la aplicación, la cual cuenta con conexión a la base de datos para el almacenamiento de la información.

Arquitectura del software

La arquitectura de software se demuestra con un patrón de Modelo-Vista-Controlador (MVC), este patrón permite separar la interfaz de usuario, los datos y la lógica de negocio. Esta separación de módulos favorece la escalabilidad y el mantenimiento del sistema. Utilizando el framework de Blazor, se trabaja la lógica de negocio y la vista al usuario, y utilizando SQL Server, la capa de los datos.

Figura 18
Arquitectura del software



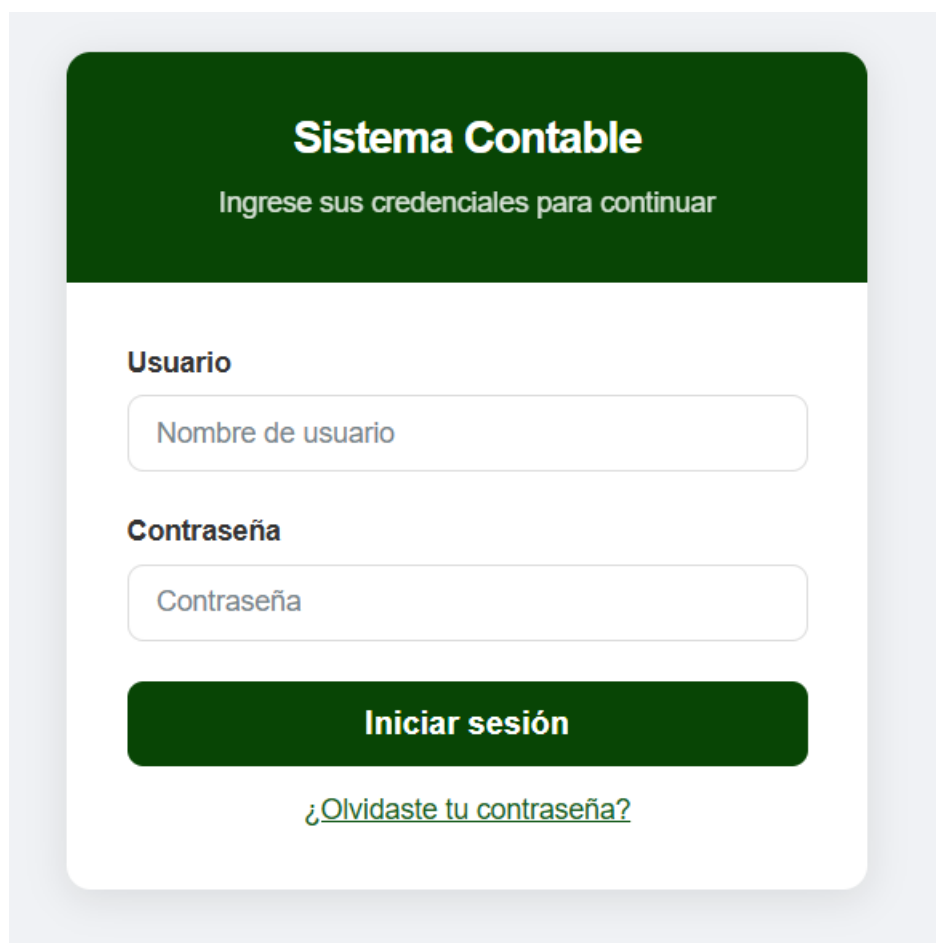
Fuente: Elaboración propia

La figura 8 representa la arquitectura del software, la cual se basa en el patrón de MVC, el proceso se inicia cuando el usuario mediante la red local accede al sistema, mostrando la vista, la cual está desarrollada en C# utilizando el framework de Blazor, encargado de desarrollar el front-end de la aplicación. El controlador se encarga de la lógica de negocio, desarrollado en Blazor, procesa las solicitudes provenientes del *front-end* y gestiona la comunicación con la base de datos, la cual se encuentra en la capa del modelo, desarrollado con los componentes Blazor y la base de datos de SQL Server.

Diseño de entradas

Figura 19

Pantalla de inicio de sesión



The image shows a login interface for a system titled "Sistema Contable". At the top, a dark green header contains the title and the instruction "Ingrese sus credenciales para continuar". Below this, the form is divided into two sections: "Usuario" and "Contraseña". Each section has a text input field with a placeholder label. At the bottom of the form is a large green button labeled "Iniciar sesión". Below the button is a link that says "¿Olvidaste tu contraseña?".

Sistema Contable

Ingrese sus credenciales para continuar

Usuario

Nombre de usuario

Contraseña

Contraseña

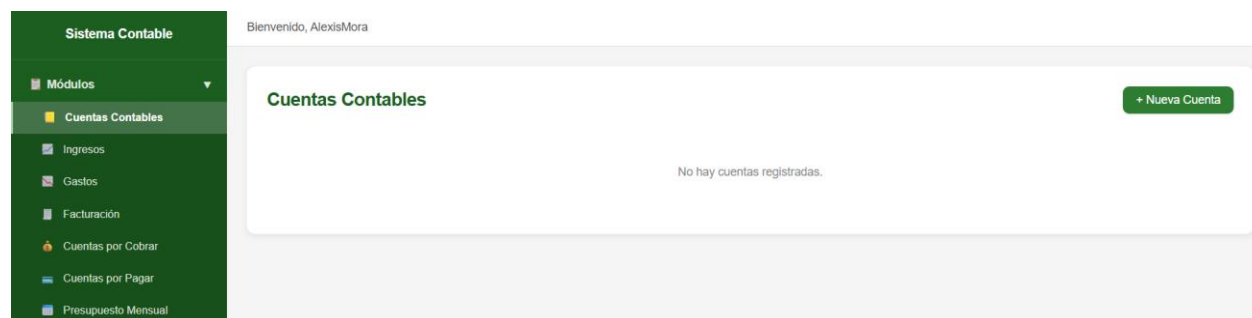
Iniciar sesión

[¿Olvidaste tu contraseña?](#)

Fuente: Elaboración propia

La figura 9 muestra la pantalla de inicio de sesión del sistema contable y de facturación de la soda; en ella, el usuario digita su respectivo usuario y contraseña para poder ingresar al sistema. En caso de requerirlo, el usuario puede utilizar la opción de recuperar la contraseña, en donde recibirá al correo electrónico registrado, un formulario para reestablecer la contraseña.

Figura 20
Gestión de Cuentas Contables



Fuente: Elaboración propia

La figura 10 muestra la pantalla principal del módulo de gestión de cuentas contables, en dicho módulo el usuario podrá visualizar la cantidad total de cuentas, así como distintos tipos de clasificación, ya sea que estén activas o cuentas que correspondan a ingresos. El usuario podrá crear una nueva cuenta, la cual se verá reflejada en la pantalla principal del módulo.

Figura 21
Formulario Proveedor

Nuevo Proveedor

×

Nombre

Correo

Teléfono

Tipo de Identificación

-- Seleccione --

Número de Identificación

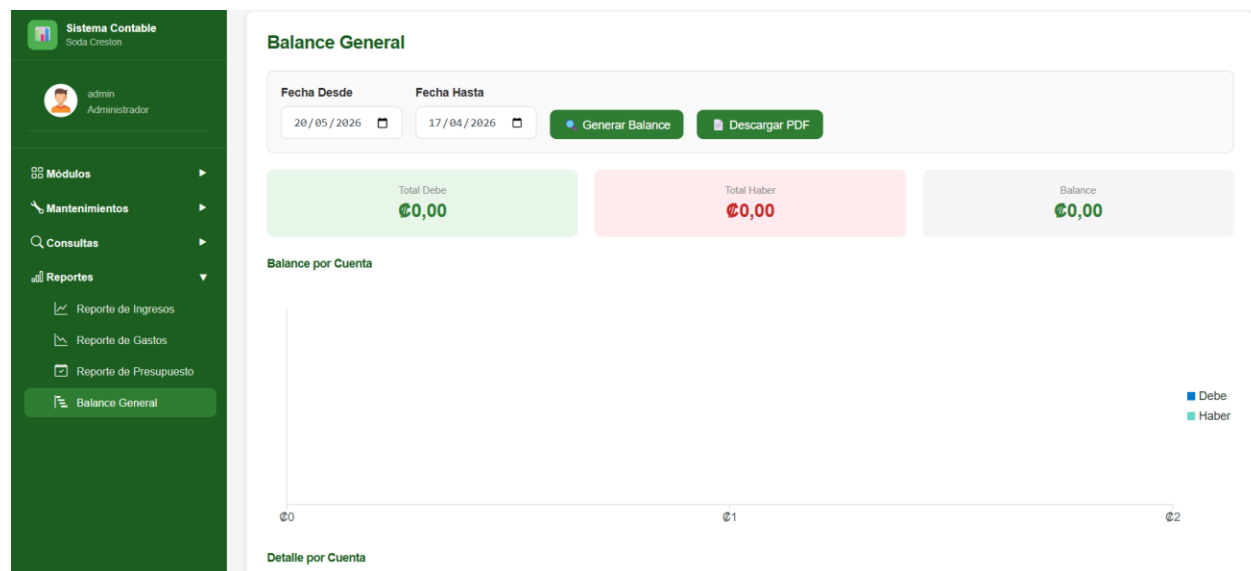
Cancelar

Guardar

Fuente: Elaboración propia

La figura 11 representa la creación de un nuevo proveedor. En este formulario el usuario correspondiente ingresa los datos requeridos para guardar un nuevo registro de proveedor, ingresando el nombre, el correo, el teléfono, además del tipo y número de identificación.

Figura 22
Vista Reporte Balance General



Fuente: Elaboración propia

La figura 12 representa la vista del módulo de balance general, en donde el usuario puede filtrar por fechas el balance que desea visualizar y revisar, mediante un gráfico, el balance de las cuentas registradas. Además, puede visualizar el detalle de la cuenta específica y, a nivel general, visualmente se muestra el total del Debe, del Haber y el balance.

Figura 23
Vista Usuario

El formulario 'Nuevo Usuario' presenta los siguientes campos:

- Identificación:** Campo de texto con el valor '0'.
- Nombre:** Campo de texto vacío.
- Primer Apellido:** Campo de texto vacío.
- Segundo Apellido:** Campo de texto vacío.
- Correo:** Campo de texto vacío.
- Nombre de Usuario:** Campo de texto vacío.
- Contraseña:** Campo de texto vacío.
- Rol:** Selector de lista desplegable con la opción '-- Seleccione --'.

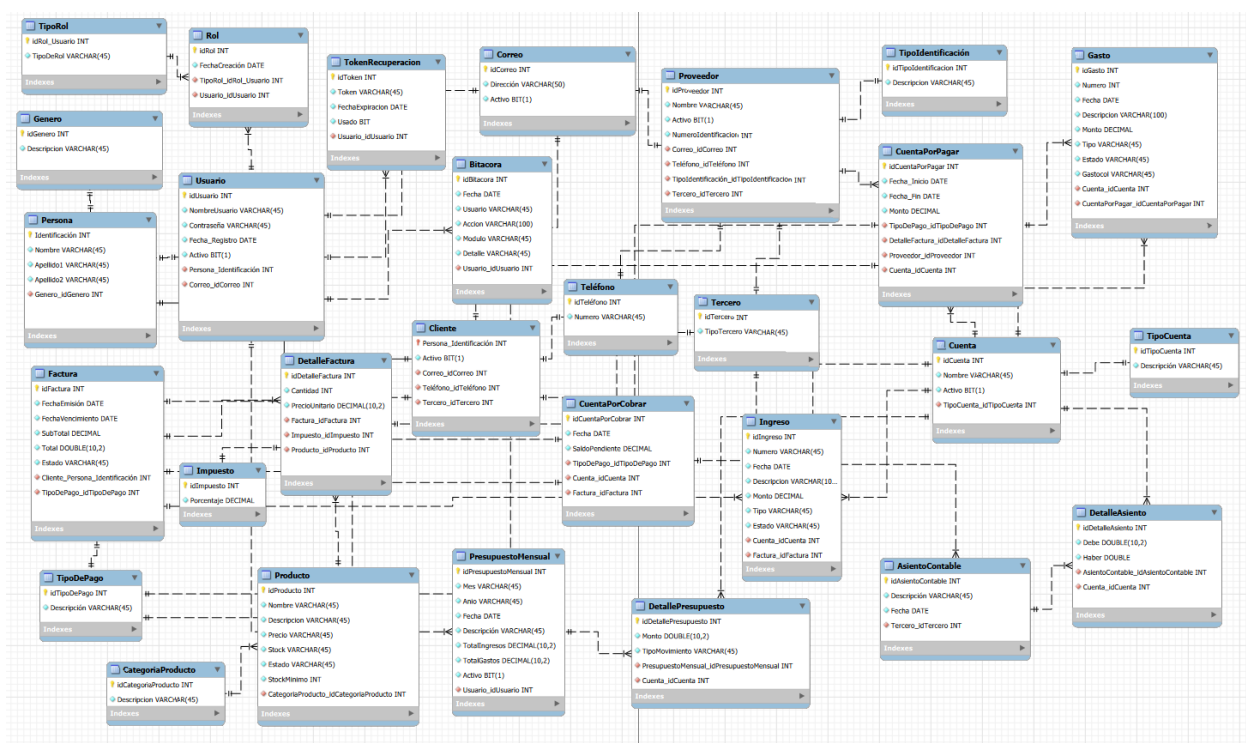
En la parte inferior del formulario se encuentran dos botones: 'Cancelar' (gris) y 'Guardar' (verde).

Fuente: Elaboración propia

La figura 13 representa el formulario de creación de usuario. En esta interfaz el usuario administrador completa la información respectiva de los datos de usuario y la creación de este, asignando credenciales y un tipo de usuario, el cual garantiza el rol indicado.

Diseño físico de la base de datos

Figura 24
Modelo de entidad-relación



Fuente: Elaboración propia

Diccionario de datos

Tabla 21

Tabla Persona

Nombre_Columna	Tipo_Dato	PK	FK	Null
Identificación	INT	Sí	No	No
Nombre	VARCHAR (45)	No	No	No
Apellido1	VARCHAR (45)	No	No	No
Apellido2	VARCHAR (45)	No	No	No
Género_idGénero	INT	No	Sí	No

Fuente: Elaboración propia

Tabla 22

Tabla Género

Nombre_Columna	Tipo_Dato	PK	FK	Null
idGénero	INT	Sí	No	No
Descripción	VARCHAR (45)	No	No	No

Fuente: Elaboración propia

Tabla 23

Tabla Usuario

Nombre_Columna	Tipo_Dato	PK	FK	Null
idUsuario	INT	Sí	No	No
NombreUsuario	VARCHAR (45)	No	No	No
Contraseña	VARCHAR (45)	No	No	No
Fecha_Registro	DATE	No	No	No
Activo	BIT (1)	No	No	No
Persona_Identificación	INT	No	Sí	No
Correo_idCorreo	INT	No	Sí	No

Fuente: Elaboración propia

Tabla 24*Tabla Rol*

Nombre_Columna	Tipo_Dato	PK	FK	Null
idRol	INT	Sí	No	No
FechaCreación	DATE	No	No	No
TipoRol_idRol_Usuario	INT	No	Sí	No
Usuario_idUsuario	INT	No	Sí	No

Fuente: Elaboración propia

Tabla 25*Tabla TipoRol*

Nombre_Columna	Tipo_Dato	PK	FK	Null
idRol_Usuario	INT	Sí	No	No
TipoDeRol	VARCHAR (45)	No	No	No

Fuente: Elaboración propia

Tabla 26*Tabla Cliente*

Nombre_Columna	Tipo_Dato	PK	FK	Null
Persona_Identificación	INT	Sí	No	No
Activo	BIT (1)	No	No	No
Correo_idCorreo	INT	No	Sí	No
Teléfono_idTeléfono	INT	No	Sí	No
Tercero_idTercero	INT	No	Sí	No

Fuente: Elaboración propia

Tabla 27*Tabla Correo*

Nombre_Columna	Tipo_Dato	PK	FK	Null
idCorreo	INT	Sí	No	No
Dirección	VARCHAR (50)	No	No	No
Activo	BIT (1)	No	No	No

Fuente: Elaboración propia

Tabla 28*Tabla Teléfono*

Nombre_Columna	Tipo_Dato	PK	FK	Null
idTeléfono	INT	Sí	No	No
Número	VARCHAR (45)	No	No	No

Fuente: Elaboración propia

Tabla 29*Tabla Proveedor*

Nombre_Columna	Tipo_Dato	PK	FK	Null
idProveedor	INT	Sí	No	No
Nombre	VARCHAR (45)	No	No	No
Activo	BIT (1)	No	No	No
NúmeroIdentificación	VARCHAR (20)	No	No	No
Correo_idCorreo	INT	No	Sí	No
Teléfono_idTeléfono	INT	No	Sí	No
TipoIdentificación_idTipoIdentificación	INT	No	Sí	No
Tercero_idTercero	INT	No	Sí	No

Elaboración propia

Tabla 30*Tabla TipoIdentificación*

Nombre_Columna	Tipo_Dato	PK	FK	Null
idTipoIdentificación	INT	Sí	No	No
Descripción	VARCHAR (45)	No	No	No

Fuente: Elaboración propia

Tabla 31*Tabla CuentaPorPagar*

Nombre_Columna	Tipo_Dato	PK	FK	Null
idCuentaPorPagar	INT	Sí	No	No
Fecha_Inicio	DATE	No	No	No
Fecha_Fin	DATE	No	No	No
Monto	DECIMAL	No	No	No
TipoDePago_idTipoDePago	INT	No	Sí	No
DetalleFactura_idDetalleFactura	INT	No	Sí	No
Cuenta_idCuenta	INT	No	Sí	No

Elaboración propia

Tabla 32*Tabla Tercero*

Nombre_Columna	Tipo_Dato	PK	FK	Null
idTercero	INT	Sí	No	No
TipoTercero	VARCHAR (45)	No	No	No

Fuente: Elaboración propia

Tabla 33*Tabla TipoCuenta*

Nombre_Columna	Tipo_Dato	PK	FK	Null
idTipoCuenta	INT	Sí	No	No
Descripción	VARCHAR (45)	No	No	No

Fuente: Elaboración propia

Tabla 34*Tabla Cuenta*

Nombre_Columna	Tipo_Dato	PK	FK	Null
idCuenta	INT	Sí	No	No
Nombre	VARCHAR (45)	No	No	No
Activo	BIT (1)	No	No	No
TipoCuenta_idTipoCuenta	INT	No	Sí	No

Fuente: Elaboración propia

Tabla 35*Tabla CuentaPorCobrar*

Nombre_Columna	Tipo_Dato	PK	FK	Null
idCuentaPorCobrar	INT	Sí	No	No
Fecha	DATE	No	No	No
Monto	DECIMAL	No	No	No
Factura_idFactura	INT	No	Sí	No
TipoDePago_idTipoDePago	INT	No	Sí	No
Cuenta_idCuenta	INT	No	Sí	No

Fuente: Elaboración propia

Tabla 36*AsientoContable*

Nombre_Columna	Tipo_Dato	PK	FK	Null
idAsientoContable	INT	Sí	No	No
Descripción	VARCHAR (45)	No	No	No
Fecha	DATE	No	No	No
Tercero_idTercero	INT	No	Sí	No

Fuente: Elaboración propia

Tabla 37*Tabla DetalleAsiento*

Nombre_Columna	Tipo_Dato	PK	FK	Null
idDetalleAsiento	INT	Sí	No	No
Debe	DOUBLE (10,2)	No	No	No
Haber	DOUBLE (10,2)	No	No	No
AsientoContable_idAsientoContable	INT	No	Sí	No
Cuenta_idCuenta	INT	No	Sí	No

Fuente: Elaboración propia

Tabla 38*PresupuestoMensual*

Nombre_Columna	Tipo_Dato	PK	FK	Null
idPresupuestoMensual	INT	Sí	No	No
Mes	VARCHAR (45)	No	No	No
Anio	VARCHAR (45)	No	No	No
Fecha	DATE	No	No	No
Descripción	VARCHAR (45)	No	No	No
TotalIngresos	DECIMAL (10,2)	No	No	No

TotalGastos	DECIMAL (10,2)	No	No	No
Activo	BIT (1)	No	No	No
Usuario_idUsuario	INT	No	Sí	No

Elaboración propia

Tabla 39

DetallePresupuesto

Nombre_Columna	Tipo_Dato	PK	FK	Null
idDetallePresupuesto	INT	Sí	No	No
Monto	DOUBLE (10,2)	No	No	No
TipoMovimiento	VARCHAR (45)	No	No	No
PresupuestoMensual_idPresupuestoMensual	INT	No	Sí	No
Cuenta_idCuenta	INT	No	Sí	No

Fuente: Elaboración propia

Tabla 40

Tabla Producto

Nombre_Columna	Tipo_Dato	PK	FK	Null
idProducto	INT	Sí	No	No
Nombre	VARCHAR (45)	No	No	No
Descripción	VARCHAR (45)	No	No	No
Precio	VARCHAR (45)	No	No	No
Stock	VARCHAR (45)	No	No	No
Estado	VARCHAR (45)	No	No	No

StockMínimo	INT	No	No	No
CategoríaProducto_idCategoríaProducto	INT	No	Sí	No

Fuente: Elaboración propia

Tabla 41

Tabla CategoríaProducto

Nombre_Columna	Tipo_Dato	PK	FK	Null
idCategoríaProducto	INT	Sí	No	No
Descripción	VARCHAR (45)	No	No	No

Fuente: Elaboración propia

Tabla 42

Tabla TipoDePago

Nombre_Columna	Tipo_Dato	PK	FK	Null
idTipoDePago	INT	Sí	No	No
Descripción	VARCHAR (45)	No	No	No

Fuente: Elaboración propia

Tabla 43

Tabla Impuesto

Nombre_Columna	Tipo_Dato	PK	FK	Null
idImpuesto	INT	Sí	No	No
Porcentaje	DECIMAL (10,2)	No	No	No

Fuente: Elaboración propia

Tabla 44

Tabla DetalleFactura

Nombre_Columna	Tipo_Dato	PK	FK	Null
idDetalleFactura	INT	Sí	No	No
Cantidad	INT	No	No	No
PrecioUnitario	DECIMAL (10,2)	No	No	No
Factura_idFactura	INT	No	Sí	No

Impuesto_idImpuesto	INT	No	Sí	No
Producto_idProducto	INT	No	Sí	No

Fuente: Elaboración propia

Tabla 45

Tabla Factura

Nombre_Columna	Tipo_Dato	PK	FK	Null
idFactura	INT	Sí	No	No
FechaEmisión	DATE	No	No	No
FechaVencimiento	DATE	No	No	No
SubTotal	DECIMAL (10,2)	No	No	No
Total	DOUBLE (10,2)	No	No	No
Estado	VARCHAR (45)	No	No	No
Cliente_Persona_Identificación	INT	No	Sí	No
TipoDePago_idTipoDePago	INT	No	Sí	No

Fuente: Elaboración propia

Tabla 46

Tabla Ingreso

Nombre_Columna	Tipo_Dato	PK	FK	Null
idIngreso	INT	Sí	No	No
Número	INT	No	No	No
Fecha	DATE	No	No	No
Descripción	VARCHAR (100)	No	No	No
Monto	DECIMAL (10,2)	No	No	No
Tipo	VARCHAR (10,2)	No	No	No
Estado	VARCHAR (45)	No	No	No

idCuenta	INT	No	Sí	No
idFactura	INT	No	Sí	No

Fuente: Elaboración propia

Tabla 47

Tabla Gasto

Nombre_Columna	Tipo_Dato	PK	FK	Null
idGasto	INT	Sí	No	No
Número	INT	No	No	No
Fecha	DATE	No	No	No
Descripción	VARCHAR (100)	No	No	No
Monto	DECIMAL (10,2)	No	No	No
Tipo	VARCHAR (10,2)	No	No	No
Estado	VARCHAR (45)	No	No	No
idCuenta	INT	No	Sí	No
idCuentaPorPagar	INT	No	Sí	No

Tabla 48

Tabla Bitácora

Nombre_Columna	Tipo_Dato	PK	FK	Null
idBitácora	INT	Sí	No	No
Fecha	DATE	No	No	No
Usuario	VARCHAR (45)	No	No	No
Acción	DECIMAL (10,2)	No	No	No
Módulo	VARCHAR (10,2)	No	No	No
Detalle	VARCHAR (45)	No	No	No
idUsuario	INT	No	Sí	No

Fuente: Elaboración propia.

Tabla 49*Tabla TokenRecuperacion*

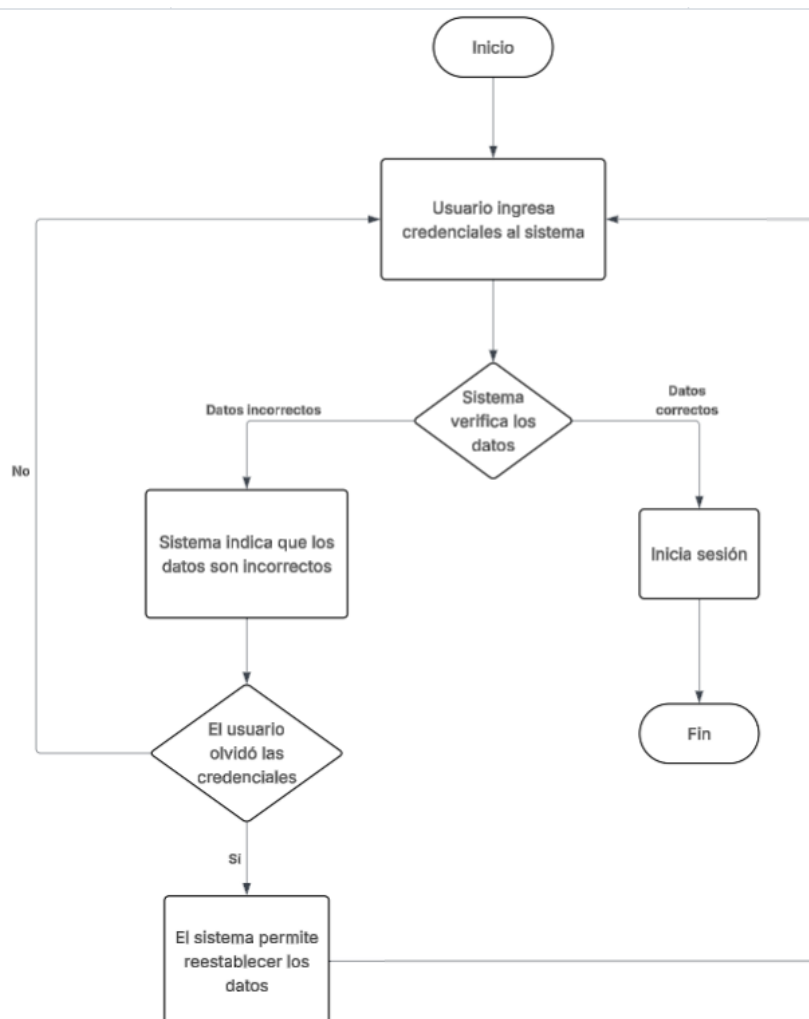
Nombre_Columna	Tipo_Dato	PK	FK	Null
idToken	INT	Sí	No	No
Token	VARCHAR (45)	No	No	No
FechaExpiración	DATE	No	No	No
Usado	BIT	No	Sí	No
IdUsuario	INT	No	Sí	No

Fuente: Elaboración propia

Diseño de procesos

En la sección de diseño de procesos se presentan los diagramas de flujo de los principales procesos del prototipo.

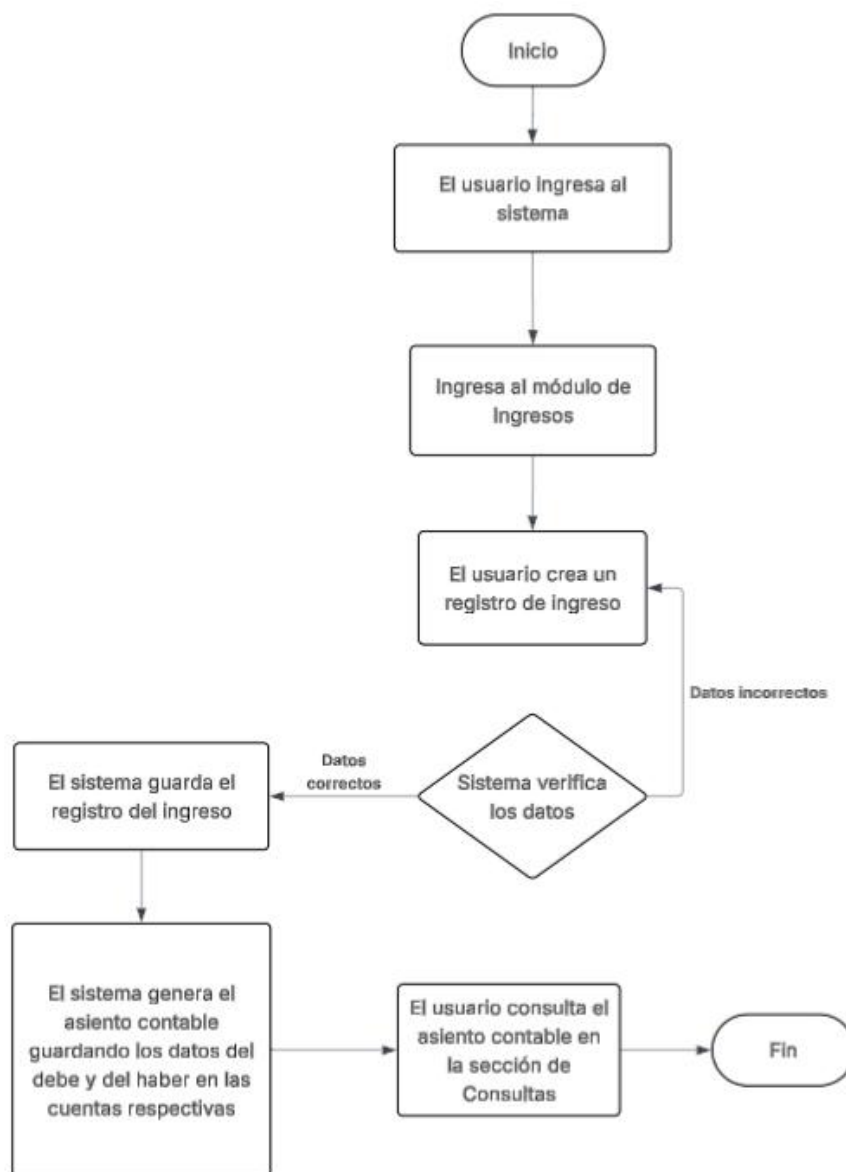
Figura 25
Diagrama de flujo inicio de sesión



Fuente: Elaboración propia

El diagrama de flujo refleja el proceso de iniciar sesión en el sistema. Primeramente, el usuario ingresa las credenciales y el sistema verifica los datos ingresados. En caso de que las credenciales estén incorrectas, el sistema indica mediante un mensaje de error que las credenciales ingresadas no son válidas, por lo que el usuario tiene la oportunidad de digitarlas nuevamente o ingresar en la opción “¿Olvidó su contraseña?” para reestablecerla. En caso de indicar las credenciales correctas, el sistema permite el ingreso al usuario.

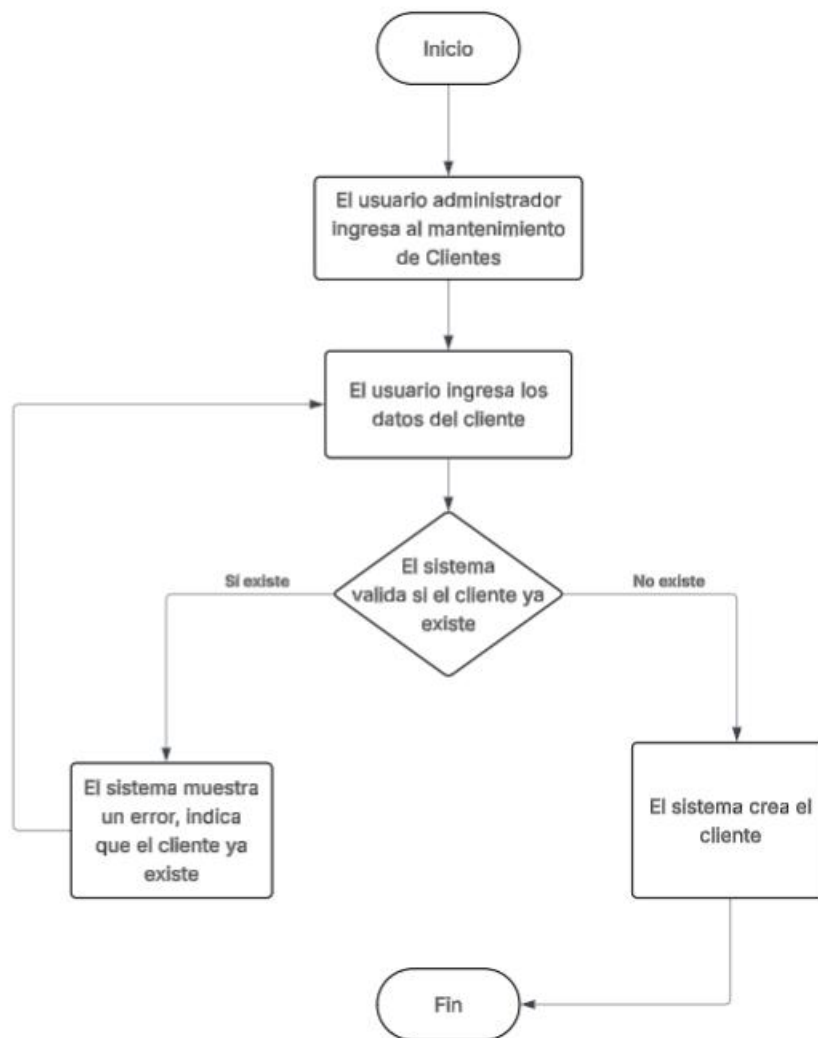
Figura 26
Diagrama de flujo Registro Asiento Contable



Fuente: Elaboración propia

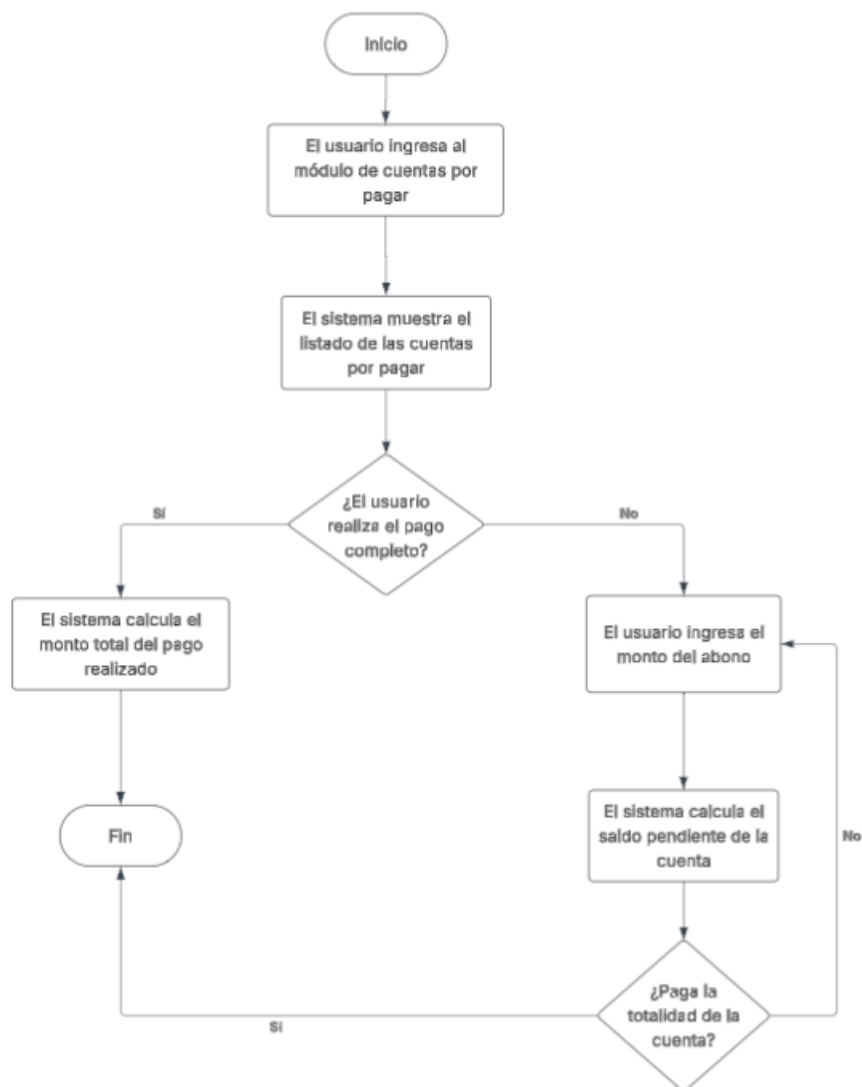
El diagrama de flujo del proceso de registrar un asiento contable inicia cuando el usuario ingresa al sistema mediante los módulos contables. En el caso de la figura 26 se utiliza el módulo de ingresos. Seguidamente realiza un registro y una vez validado el registro, el sistema genera el asiento automáticamente incluyendo la información del debe y del haber en las respectivas cuentas contables. El usuario podrá visualizar el detalle del asiento contable generado en la sección de consultas.

Figura 27
Diagrama de flujo Crear cliente



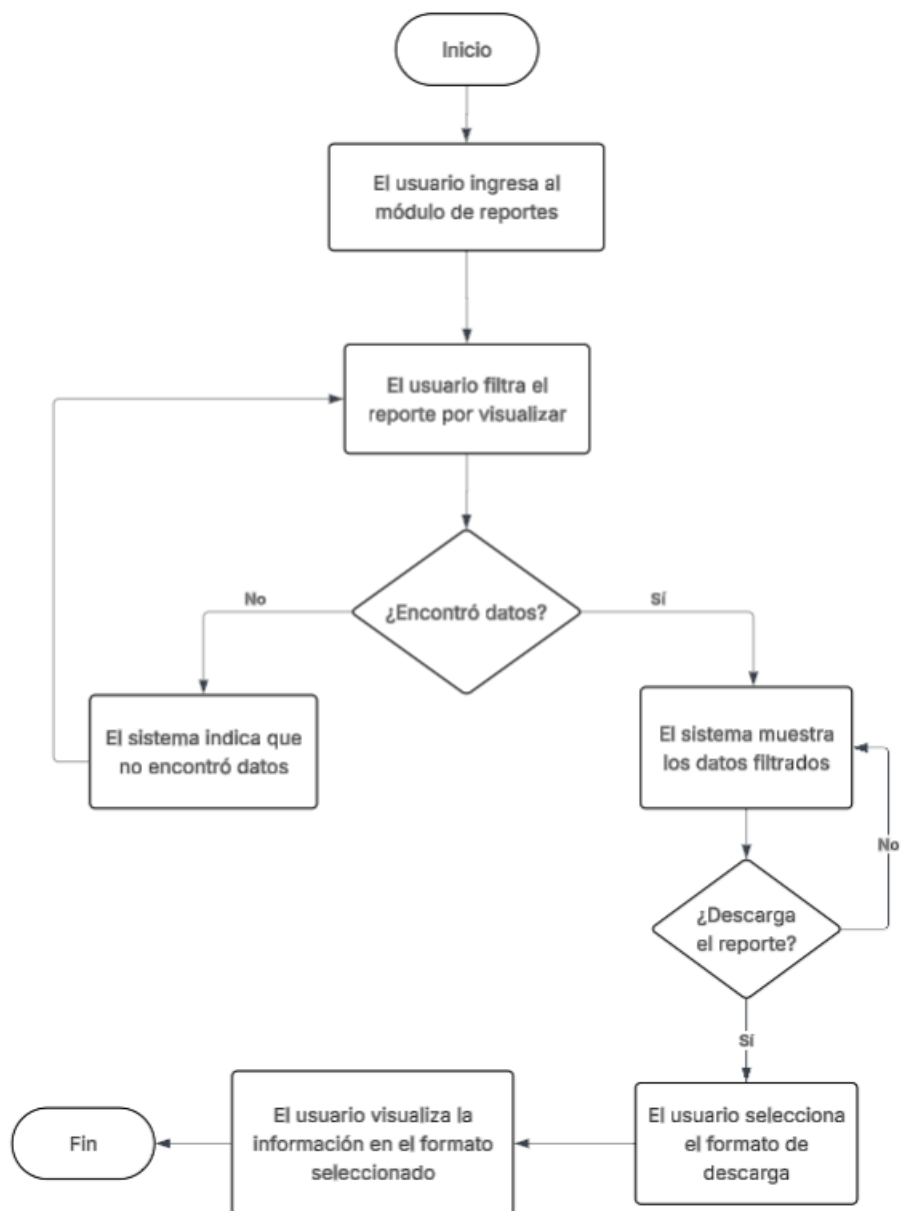
Fuente: Elaboración propia

En el diagrama de flujo del proceso de crear, el cliente inicia con el usuario administrador ingresando al mantenimiento de Clientes. De seguido, el usuario crea un nuevo cliente, ingresando la información respectiva y guarda. El sistema valida la información ingresada por el usuario y en caso de que el cliente ya exista, el sistema mostrará un error indicándolo; si el cliente no existe, el sistema crea el cliente.

Figura 28*Diagrama de flujo Cuentas por pagar*

Fuente: Elaboración propia

Figura 29
Diagrama de flujo Reportes



Fuente: Elaboración propia

Diseño de salidas

Figura 30

Vista módulo cuentas contables

Bienvenido, AlexisMora

Cuentas Contables + Nueva Cuenta			
Nombre	Tipo de Cuenta	Estado	Acciones
Caja	Activo	Activa	 
Banco	Activo	Activa	 
Ventas	Ingreso	Activa	 
Costo de Ventas	Gasto	Activa	 
Proveedores por Pagar	Pasivo	Activa	 

Fuente: Elaboración propia

La figura 30 muestra el módulo de gestión de cuentas contables. La vista presenta la información de la cuenta contable: nombre de la cuenta, tipo de cuenta y estado. Permite editar o desactivar el registro que el usuario decida, facilitando al usuario llevar un registro y control de las cuentas contables.

Figura 31
Vista Mantenimiento Proveedor

Proveedores						+ Nuevo Proveedor
Nombre	Correo	Teléfono	Tipo Identificación	Identificación	Estado	Acciones
Dos Pinos	dospinos@gmail.com	80808080	Cédula Jurídica	3004045002	Activo	✎ ✖
Distribuidora Comida S.A	comidasa@gmail.com	20202020	Cédula Jurídica	3000104050	Activo	✎ ✖

Fuente: Elaboración propia

La figura 31 muestra el detalle de la pestaña Mantenimientos, cuando se visualizan los proveedores en esta pestaña, el usuario puede ver la cantidad total de proveedores registrados, con la siguiente información: identificación, nombre, correo y teléfono, y tiene la posibilidad de eliminar el registro o editar.

Figura 32
Vista Usuarios

Usuarios					+ Nuevo Usuario
Usuario	Nombre Completo	Correo	Fecha Registro	Estado	Acciones
admin	Administrador Sistema Contable	admin@sodaescolar.com	05/06/2026	Activo	✎ ✖
amora	Alexis Mora Mora	moraalexis396@gmail.com	05/06/2026	Activo	✎ ✖

Fuente: Elaboración propia

La Figura 32 muestra la información relacionada con los usuarios registrados, esta vista está al alcance del usuario administrador que puede gestionar el tipo de rol de los usuarios registrados y tiene las acciones de editar o eliminar registros.

Figura 33
Vista Cuenta por cobrar

Cuentas por Cobrar						+ Nueva Cuenta por Cobrar
Fecha	Factura	Cliente	Tipo de Pago	Saldo Pendiente	Acciones	
05/06/2026	#19	Sergio Pérez	Efectivo	€1 200,00	 Registrar Pago	

Fuente: Elaboración propia

La figura 33 muestra el listado de cuentas por cobrar. El usuario puede visualizar la fecha en que se creó la cuenta por cobrar, el cliente, el saldo pendiente y el estado de la cuenta, permitiendo llevar el control de estas. Además, el usuario podrá registrar pagos realizados, en caso de que el cliente realice abonos a la cuenta, el saldo se recalculará de acuerdo con el monto ingresado.

Figura 34
Vista Facturación

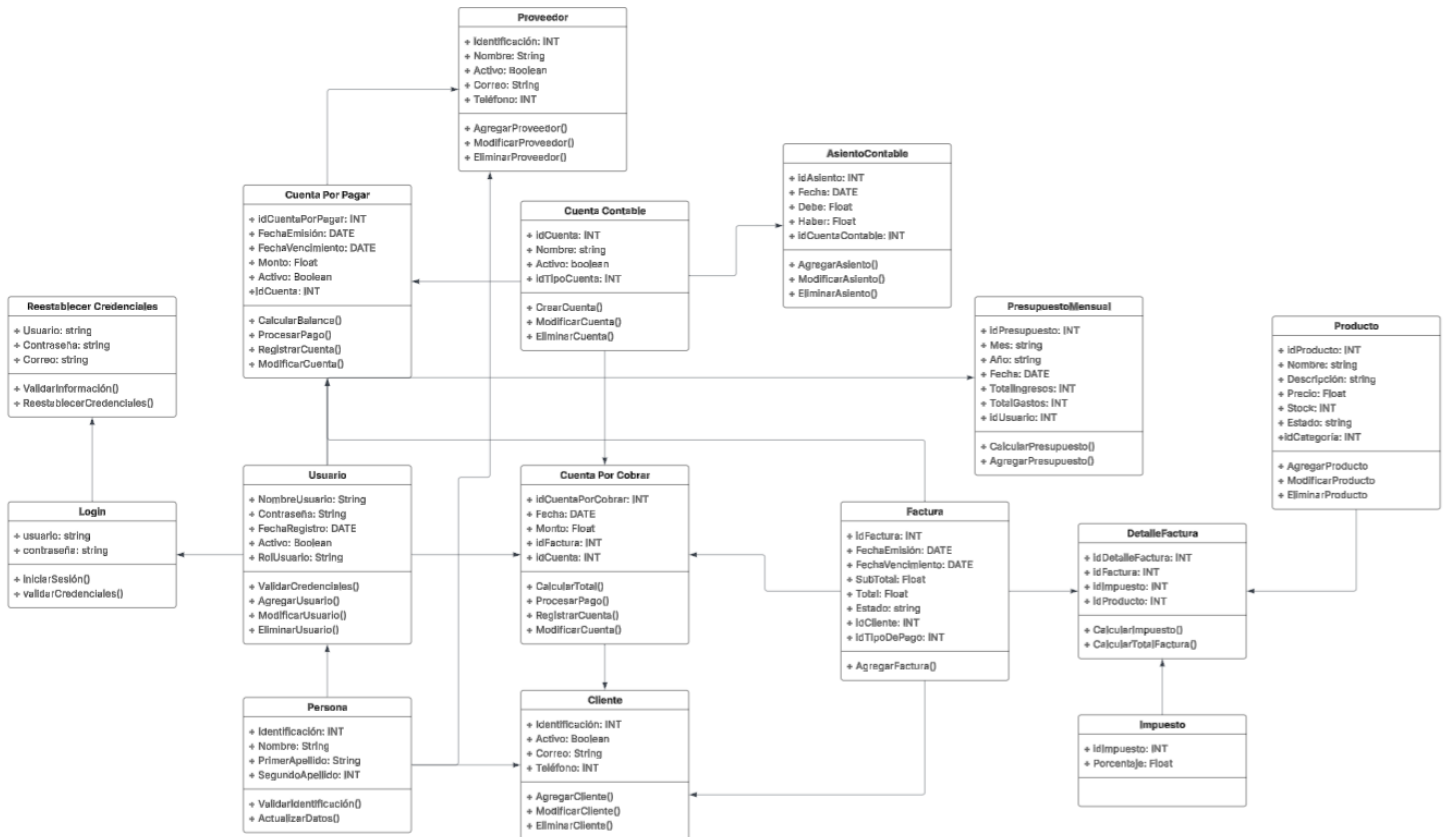
Facturación								+ Nueva Factura
N° Factura	Fecha Emisión	Fecha Vencimiento	Cliente	SubTotal	Total	Estado	Acciones	
18	05/06/2026	05/07/2026	Sergio Pérez	€3 000,00	€3 390,00	Pagada		
19	05/06/2026	05/07/2026	Sergio Pérez	€1 200,00	€1 200,00	Pendiente	  	

Fuente: Elaboración propia

La figura 34 muestra el resumen del detalle de las facturas realizadas, en el módulo se podrán agregar nuevas facturas, así como visualizar el estado de las facturas existentes, en caso de estar como “pagada” o “pendiente”; además, permite al usuario guardar la factura como PDF o imprimirla.

Diagramas UML

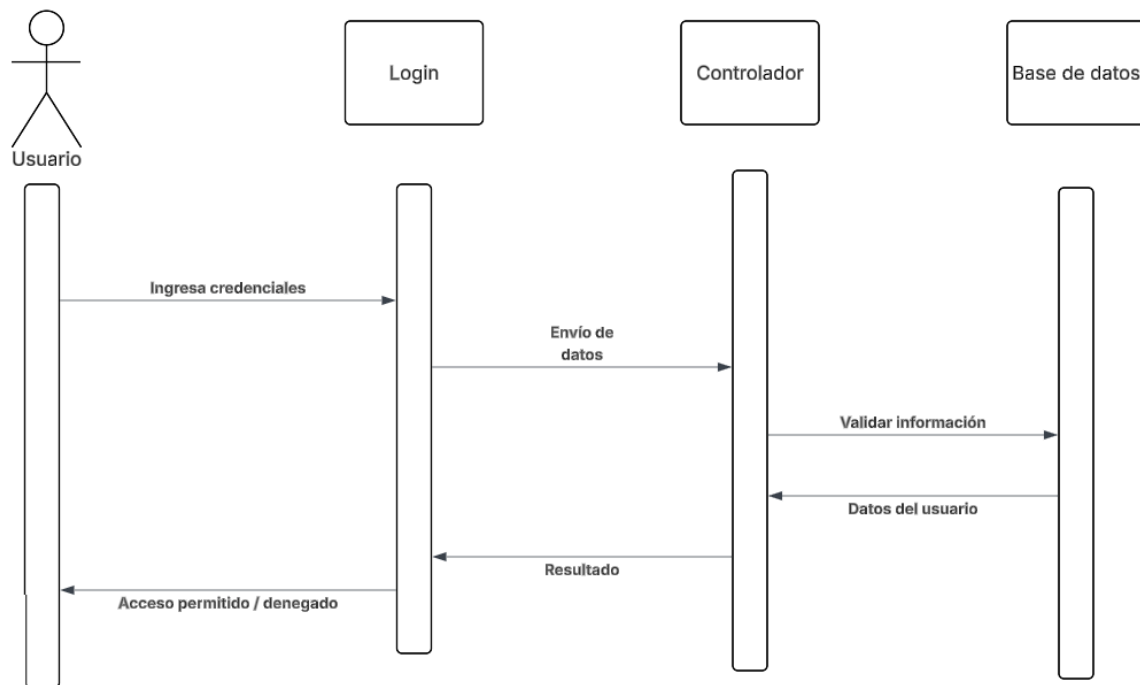
Figura 35
Diagrama de Clases



Fuente: Elaboración propia

El diagrama de clases, presentado en la figura 35, muestra la estructura del sistema, las principales tablas, los atributos y métodos que se ejecutan en los procesos del sistema y la relación que tienen los distintos procesos entre sí, mostrando el funcionamiento general.

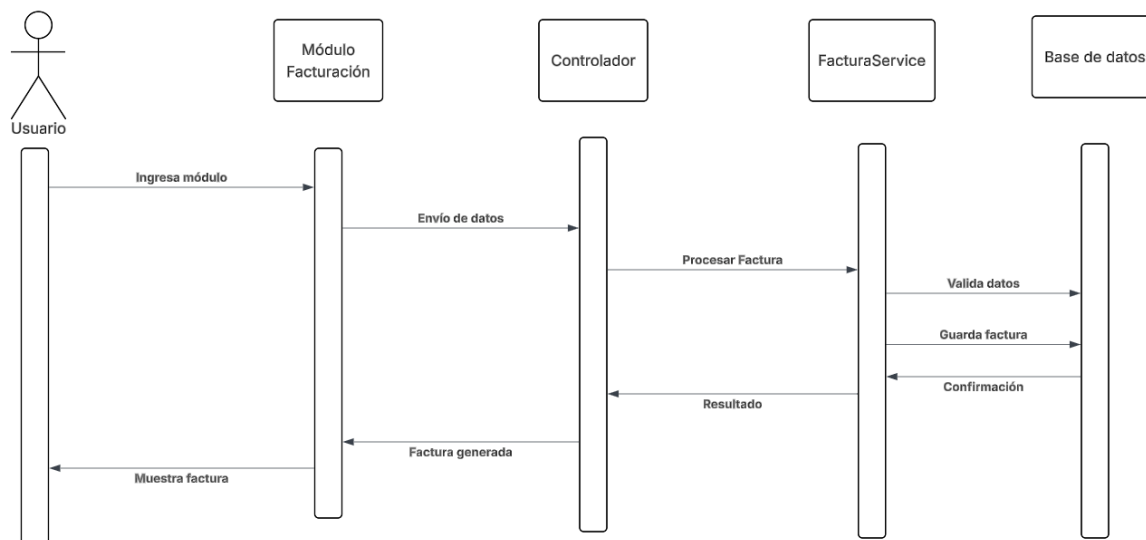
Figura 36
Diagrama de secuencia Inicio de Sesión



Fuente: Elaboración propia

La figura 36 muestra el diagrama de secuencia del proceso de iniciar sesión, se visualiza la interacción que tiene el usuario con el *Login*, el controlador de autenticación y la base de datos. Este proceso muestra el envío de credenciales, la validación de información y el acceso permitido o denegado al usuario.

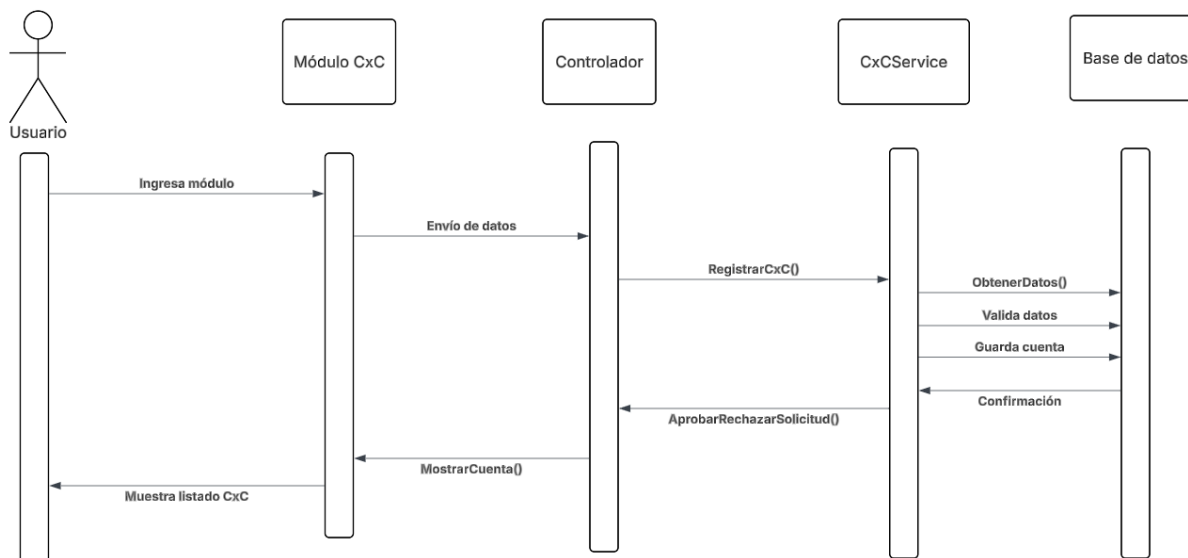
Figura 37
Diagrama de secuencia Facturación



Fuente: Elaboración propia

El diagrama de secuencia de facturación muestra el proceso del usuario al registrar una factura y cómo interaccionan el controlador y la capa lógica para procesar la factura y validar los datos registrados en base de datos.

Figura 38
Diagrama de secuencia Cuenta por cobrar

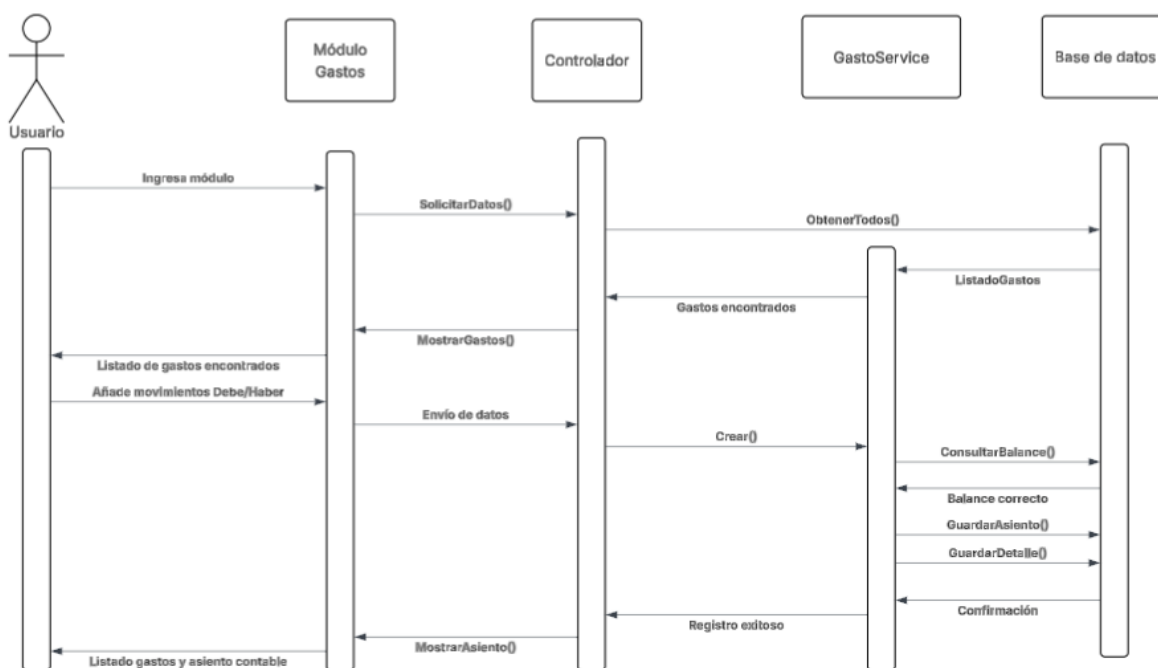


Fuente: Elaboración propia

El diagrama de secuencia de las cuentas por cobrar representa la interacción del usuario con la interfaz del sistema, el controlador y la capa lógica para registrar las cuentas por cobrar del usuario, validando los datos y mostrando en la vista la información consultada.

Figura 39

Diagrama de secuencia Asiento contable



Fuente: Elaboración propia

El diagrama de secuencia del registro de un asiento contable refleja la interacción entre el usuario y el sistema en el momento de registrar movimientos en las cuentas contables. El usuario consulta los gastos en el módulo respectivo y añade movimientos de debe/haber, posteriormente, el sistema valida la información y el balance de la cuenta y guarda el asiento y su detalle, mostrando el registro al usuario en la interfaz del módulo.

Programación

En esta sección se muestran pequeños ejemplos de código que conforman el sistema, con el fin de evidenciar los elementos más importantes del proyecto como lo son las entradas y salidas, las validaciones, lógica de negocio y el desarrollo de los módulos.

Entradas y salidas

Figura 40

Registro Usuario

```
<div class="form-group">
  <label>Identificación</label>
  <div style="display: flex; gap: 8px;">
    <input type="number" @bind="usuarioSeleccionado.Identificacion" disabled="@esEdicion" style="flex: 1;" />
    @if (!esEdicion){
      <button class="btn-agregar" @onclick="BuscarCedula" disabled="@buscandoCedula" style="white-space: nowrap;">
        @if (buscandoCedula){
          <span>Buscando...</span>
        }else{
          <span>🔍 Buscar</span>
        }
      </button>
    }
  </div>
</div>
<div class="form-group">
  <label>Nombre</label>
  <input type="text" @bind="usuarioSeleccionado.Persona.Nombre" />
</div>
<div class="form-group">
  <label>Primer Apellido</label>
  <input type="text" @bind="usuarioSeleccionado.Persona.Apellido1" />
</div>
<div class="form-group">
  <label>Segundo Apellido</label>
  <input type="text" @bind="usuarioSeleccionado.Persona.Apellido2" />
</div>
<div class="form-group">
  <label>Correo</label>
  <input type="email" @bind="usuarioSeleccionado.Correo.Direccion" />
</div>
<div class="form-group">
  <label>Nombre de Usuario</label>
  <input type="text" @bind="usuarioSeleccionado.NombreUsuario" />
</div>
<div class="form-group">
  <label>Contraseña</label>
  <input type="password" @bind="usuarioSeleccionado.Contrasena" />
</div>
<div class="form-group">
  <label>Rol</label>
  <select @bind="idTipoRolSeleccionado">
    <option value="0">-- Seleccione --</option>
    @foreach(var tipo in tiposRol)
    {
      <option value="@tipo.IdRol_Usuario">@tipo.TipoDeRol</option>
    }
  </select>
</div>
```

Fuente: Elaboración propia

Este fragmento de código representa la creación y edición de los usuarios. Primeramente y al consumir el API del TSE, cuando el usuario digita la cédula, el sistema consulta el nombre y los apellidos, de acuerdo con su identificación y seguidamente se completan los campos restantes que

se digitan para crear el usuario con su rol respectivo, que le permitirá acceder a los módulos correspondientes.

Figura 41

Salida Productos Stock Mínimo

```
@if (productos.Any(p => p.Stock <= p.StockMinimo && p.Estado == "Activo"))
{
    <div style="background: #ffebee; border-left: 4px solid #c62828; padding: 12px 16px; border-radius: 8px; margin-bottom: 16px;">
        <strong style="color: #c62828;">⚠️ Productos con stock bajo:</strong>
        <ul style="margin: 8px 0 0 0; padding-left: 20px;">
            @foreach (var p in productos.Where(p => p.Stock <= p.StockMinimo && p.Estado == "Activo"))
            {
                <li style="color: #c62828;">@p.Nombre - Stock actual: @p.Stock (mínimo: @p.StockMinimo)</li>
            }
        </ul>
    </div>
}
```

Fuente: Elaboración propia

El fragmento de código de la figura 41 representa a una salida del sistema que está condicionada cuándo un producto se encuentra con el stock menor al mínimo recomendado, es una salida dinámica ya que toma el dato actual de la base de datos y lo compara directamente con el campo de stock mínimo, en caso de que se cumpla esa condición y el producto se encuentre activo, el usuario puede visualizar en el *dashboard* general, que el producto se encuentra en stock bajo.

Procesos

Figura 42
Generación asientos contables

```

if(factura.Estado == "Pagada")
{
    var cuentaCaja = await _context.Cuentas.FirstOrDefaultAsync(c => c.Nombre == "Caja");
    var cuentaVentas = await _context.Cuentas.FirstOrDefaultAsync(c => c.Nombre == "Ventas");
    var ultimoIngreso = await _context.Ingresos.OrderByDescending(i => i.Numero).FirstOrDefaultAsync();
    var ingreso = new Ingreso
    {
        Numero = ultimoIngreso == null ? 1 : ultimoIngreso.Numero + 1,
        Fecha = factura.FechaEmision,
        Descripcion = $"Pago de Factura # {factura.IdFactura}",
        Tipo = "Factura",
        Monto = factura.Total,
        Estado = "Conciliado",
        IdCuenta = cuentaCaja!.IdCuenta
    };
    _context.Ingresos.Add(ingreso);
    await _context.SaveChangesAsync();

    await _context.Entry(factura).Reference(f => f.Cliente).LoadAsync();
    await _context.Entry(factura.Cliente).Reference(c => c.Persona).LoadAsync();

    var tercero = await _context.Terceros.FirstOrDefaultAsync(t => t.TipoTercero == "Cliente");
    var asiento = new AsientoContable
    {
        Descripcion = $"Factura #{factura.IdFactura}: {factura.Cliente.Persona.Nombre}",
        Fecha = factura.FechaEmision,
        IdTercero = tercero!.IdTercero
    };
    _context.AsientosContables.Add(asiento);
    await _context.SaveChangesAsync();

    _context.DetallesAsiento.Add(new DetalleAsiento
    {
        IdAsientoContable = asiento.IdAsientoContable,
        IdCuenta = cuentaCaja!.IdCuenta,
        Debe = factura.Total,
        Haber = 0
    });

    _context.DetallesAsiento.Add(new DetalleAsiento
    {
        IdAsientoContable = asiento.IdAsientoContable,
        IdCuenta = cuentaVentas!.IdCuenta,
        Debe = 0,
        Haber = factura.Total
    });
    await _context.SaveChangesAsync();
    await _bitacoraService.Registrar(usuarioActual, "Crear", "Facturación", $"Factura #{factura.IdFactura} creada por ${factura.Total:N2}.");
}

```

Fuente: Elaboración propia

El proceso de generar el asiento contable se representa en la figura 42, en donde se registra el ingreso cuando la factura se encuentra en estado “Pagada”. El sistema consulta las cuentas contables y realiza los movimientos del debe y del haber, siguiendo el principio de partida doble. De igual manera, se registra el detalle del asiento, con la posibilidad de que el usuario pueda verificar, en el módulo de consultas, el asiento contable generado.

Validaciones

Figura 43
Validar credenciales

```
public async Task<bool> ValidarCredenciales(string nombreUsuario, string contrasena)
{
    var usuario = await _usuarioService.ObtenerPorNombreUsuario(nombreUsuario);

    if (usuario == null || !usuario.Activo)
        return false;

    if (!BCrypt.Net.BCrypt.Verify(contrasena, usuario.Contrasena))
        return false;

    return true;
}
```

Fuente: Elaboración propia

Figura 44
Iniciar Sesión

```
private async Task IniciarSesion()
{
    cargando = true;
    errorMensaje = string.Empty;

    if (string.IsNullOrEmpty(nombreUsuario) || string.IsNullOrEmpty(contrasena))
    {
        errorMensaje = "Por favor ingrese el usuario y la contraseña";
        cargando = false;
        return;
    }

    var resultado = await AuthService.ValidarCredenciales(nombreUsuario, contrasena);

    if (resultado)
    {
        Navigation.NavigateTo($" /autenticar?usuario={nombreUsuario}&clave={contrasena}", true);
    }
    else
    {
        errorMensaje = "Usuario o contraseña incorrectos";
        cargando = false;
    }
}
```

Fuente: Elaboración propia

Las figuras 43 y 44 representan fragmentos del código que demuestran la validación de credenciales y el inicio de sesión, validando que no se ingresen datos vacíos y que los datos existan en la base de datos. Internamente, el sistema valida la contraseña ingresada por el usuario con el *hash* almacenado en la base de datos para garantizar la seguridad de la información y para evitar los múltiples intentos. De manera simultánea se aplica la idempotencia en el botón de ingresar al sistema.

Módulos

Figura 45

Módulo cuentas contables

```
private async Task CargarDatos()
{
    cargando = true;
    cuentas = await CuentaService.ObtenerTodos();
    tiposCuenta = await TipoCuentaService.ObtenerTodos();
    cargando = false;
}

private void MostrarFormulario()
{
    esEdicion = false;
    cuentaSeleccionada = new Cuenta { Activo = true };
    mensajeError = string.Empty;
    mostrarModal = true;
}

private void EditarCuenta(Cuenta cuenta)
{
    esEdicion = true;
    cuentaSeleccionada = cuenta;
    mensajeError = string.Empty;
    mostrarModal = true;
}

private async Task GuardarCuenta()
{
    if (guardando) return;
    guardando = true;

    if (string.IsNullOrEmpty(cuentaSeleccionada.Nombre) || cuentaSeleccionada.IdTipoCuenta == 0)
    {
        mensajeError = "Por favor complete todos los campos obligatorios.";
        guardando = false;
        return;
    }

    if (esEdicion)
        await CuentaService.Actualizar(cuentaSeleccionada);
    else
        await CuentaService.Crear(cuentaSeleccionada);

    await CargarDatos();
    CerrarModal();
    guardando = false;
}

private async Task DesactivarCuenta(int id)
{
    await CuentaService.Desactivar(id);
    await CargarDatos();
}
```

Fuente: Elaboración propia

La figura 45 representa el fragmento de código con las operaciones CRUD del módulo de cuentas contables. El método “GuardarCuenta” se encarga de crear y editar la cuenta contable, validando la integridad de los datos; y el método de “DesactivarCuenta” se encarga de modificar el estado de la cuenta seleccionada por el usuario, en un estado de inactivo. Las cuentas se muestran en el formulario cuando el sistema carga los datos registrados y permite al usuario realizar las operaciones necesarias.

Figura 46
Módulo Cuentas por cobrar

```
private async Task CargarDatos()
{
    cargando = true;
    cuentas = await CuentaPorCobrarService.ObtenerTodos();
    facturas = await FacturaService.ObtenerTodos();
    cuentasContables = await CuentaService.ObtenerActivas();
    tiposPago = await TipoDePagoService.ObtenerTodos();
    cargando = false;
}

private void MostrarFormulario()
{
    cuentaSeleccionada = new CuentaPorCobrar { Fecha = DateOnly.FromDateTime(DateTime.Today) };
    mensajeError = string.Empty;
    mostrarModalNuevo = true;
}

private void MostrarPago(CuentaPorCobrar cuenta)
{
    cuentaSeleccionada = cuenta;
    montoPago = 0;
    mensajeError = string.Empty;
    mostrarModalPago = true;
}

private async Task GuardarCuenta()
{
    if (guardando) return;
    guardando = true;

    if (cuentaSeleccionada.IdFactura == 0 || cuentaSeleccionada.IdTipoDePago == 0 || cuentaSeleccionada.IdCuenta == 0 || cuentaSeleccionada.SaldoPendiente <= 0)
    {
        mensajeError = "Por favor complete todos los campos obligatorios.";
        guardando = false;
        return;
    }

    await CuentaPorCobrarService.Crear(cuentaSeleccionada);
    await CargarDatos();
    CerrarModal();
    guardando = false;
}

private async Task RegistrarPago()
{
    if (montoPago <= 0 || montoPago > cuentaSeleccionada.SaldoPendiente)
    {
        mensajeError = "El monto debe ser mayor a 0 y no puede superar el saldo pendiente.";
        return;
    }

    await CuentaPorCobrarService.RegistrarPago(cuentaSeleccionada.IdCuentaPorCobrar, montoPago);
    await CargarDatos();
    CerrarModal();
}
}
```

Fuente: Elaboración propia

La figura 46 representa al módulo de cuentas por cobrar, que efectúa las operaciones de guardar el registro de la cuenta por cobrar y de mostrar el pago realizado. Mediante el método de “CargarDatos”, el sistema obtiene la información registrada para que el usuario la pueda visualizar, y de igual forma, el módulo permite registrar pagos realizados a la cuenta, en caso de necesitar realizar abonos, o la totalidad del pago.

Figura 47
Módulo Cuentas por pagar

```
private async Task CargarDatos()
{
    cargando = true;
    cuentas = await CuentaPorPagarService.ObtenerTodos();
    proveedores = await ProveedorService.ObtenerActivos();
    cuentasContables = await CuentaService.ObtenerActivas();
    tiposPago = await TipoDePagoService.ObtenerTodos();
    cargando = false;
}

private void MostrarFormulario()
{
    cuentaSeleccionada = new CuentaPorPagar
    {
        Fecha_Inicio = DateOnly.FromDateTime(DateTime.Today),
        Fecha_Fin = DateOnly.FromDateTime(DateTime.Today.AddDays(30)),
        IdDetalleFactura = null
    };
    mensajeError = string.Empty;
    mostrarModal = true;
}

private async Task GuardarCuenta()
{
    if (guardando) return;
    guardando = true;

    if (cuentaSeleccionada.IdProveedor == 0 || cuentaSeleccionada.IdTipoDePago == 0 || cuentaSeleccionada.IdCuenta == 0 || cuentaSeleccionada.Monto <= 0)
    {
        mensajeError = "Por favor complete todos los campos obligatorios.";
        guardando = false;
        return;
    }

    await CuentaPorPagarService.Crear(cuentaSeleccionada);
    await CargarDatos();
    CerrarModal();
    guardando = false;
}

private async Task PagarCuenta(int id)
{
    await CuentaPorPagarService.RegistrarPago(id, usuarioActual);
    await CargarDatos();
}

private void CerrarModal()
{
    mostrarModal = false;
    mensajeError = string.Empty;
}
```

Fuente: Elaboración propia

La figura 47 muestra las operaciones de guardar y pagar las cuentas por pagar, el método de “CargarDatos” obtiene los datos provenientes del servicio de cuentas por pagar y los muestra en el formulario para que el usuario pueda realizar las acciones necesarias.

Figura 48
Módulo Facturación

```
private void AgregarDetalle()
{
    if (productoSeleccionadoId == 0 || impuestoSeleccionadoId == 0 || cantidadSeleccionada <= 0)
    {
        mensajeError = "Seleccione un producto, impuesto y cantidad válida.";
        return;
    }
    var producto = productos.FirstOrDefault(p => p.IdProducto == productoSeleccionadoId);
    if (producto == null) return;

    detalles.Add(new DetalleFactura
    {
        IdProducto = productoSeleccionadoId,
        IdImpuesto = impuestoSeleccionadoId,
        Cantidad = cantidadSeleccionada,
        PrecioUnitario = producto.Precio
    });

    productoSeleccionadoId = 0;
    impuestoSeleccionadoId = 0;
    cantidadSeleccionada = 1;
    mensajeError = string.Empty;
}

private void EliminarDetalle(DetalleFactura detalle)
{
    detalles.Remove(detalle);
}

private decimal CalcularTotal()
{
    decimal total = 0;
    foreach (var detalle in detalles)
    {
        var impuesto = impuestos.FirstOrDefault(i => i.IdImpuesto == detalle.IdImpuesto);
        decimal subtotal = detalle.PrecioUnitario * detalle.Cantidad;
        decimal montoImpuesto = subtotal * ((impuesto?.Porcentaje ?? 0) / 100);
        total += subtotal + montoImpuesto;
    }
    return total;
}

private async Task GuardarFactura()
{
    if (guardando) return;
    guardando = true;

    if (facturaSeleccionada.IdCliente == 0 || facturaSeleccionada.IdTipoDePago == 0 || detalles.Count == 0)
    {
        mensajeError = "Por favor complete todos los campos y agregue al menos un producto.";
        guardando = false;
        return;
    }
    await FacturaService.Crear(facturaSeleccionada, detalles, usuarioActual);
    await CargarDatos();
    CerrarModal();
    guardando = false;
}
```

Fuente: Elaboración propia

La figura 48 muestra el módulo de facturación, que permite al usuario añadir facturas al sistema con su debido detalle. El sistema realiza el cálculo total de la venta, tomando en cuenta el impuesto, y lo muestra al usuario. También, permite eliminar el detalle y cambiar el estado de la factura.

Figura 49
Módulo Gastos

```
private async Task CargarDatos()
{
    cargando = true;
    gastos = await GastoService.ObtenerTodos();
    cuentas = await CuentaService.ObtenerActivas();
    cargando = false;
}

private void MostrarFormulario()
{
    gastoSeleccionado = new Gasto { Fecha = DateOnly.FromDateTime(DateTime.Today), Estado = "No conciliado" };
    idCuentaCredito = 0;
    mensajeError = string.Empty;
    mostrarModal = true;
}

private async Task GuardarGasto()
{
    if (guardando) return;
    guardando = true;

    if (string.IsNullOrEmpty(gastoSeleccionado.Descripcion) || string.IsNullOrEmpty(gastoSeleccionado.Tipo) || gastoSeleccionado.IdCuenta == 0 || idCuentaCredito == 0 || gastoSeleccionado.Monto <= 0)
    {
        mensajeError = "Por favor complete todos los campos obligatorios.";
        guardando = false;
        return;
    }
    if (gastoSeleccionado.IdCuenta == idCuentaCredito)
    {
        mensajeError = "La cuenta débito y crédito no pueden ser la misma.";
        return;
    }
    await GastoService.Crear(gastoSeleccionado, idCuentaCredito);
    await CargarDatos();
    CerrarModal();
    guardando = false;
}

private async Task ConciliarGasto(int id)
{
    await GastoService.CambiarEstado(id, "Conciliado");
    await CargarDatos();
}

private async Task EliminarGasto(int id)
{
    await GastoService.Eliminar(id);
    await CargarDatos();
}

private void CerrarModal()
{
    mostrarModal = false;
    mensajeError = string.Empty;
}
```

Fuente: Elaboración propia

La figura 49 muestra el fragmento de código del módulo que controla los gastos de la empresa, permite al usuario ingresar los gastos, visualizarlos en formato de tabla y con posibilidad de editarlos, así como conciliar el pago o eliminar el registro.

Figura 50
Módulo Ingresos

```
private void MostrarFormulario()
{
    ingresoSeleccionado = new Ingreso { Fecha = DateOnly.FromDateTime(DateTime.Today), Estado = "No conciliado" };
    idCuentaCredito = 0;
    mensajeError = string.Empty;
    mostrarModal = true;
}

private async Task GuardarIngreso()
{
    if (guardando) return;
    guardando = true;

    if (string.IsNullOrEmpty(ingresoSeleccionado.Descripcion) || string.IsNullOrEmpty(ingresoSeleccionado.Tipo) || ingresoSeleccionado.IdCuenta == 0 ||
        idCuentaCredito == 0 || ingresoSeleccionado.Monto <= 0)
    {
        mensajeError = "Por favor complete todos los campos obligatorios.";
        guardando = false;
        return;
    }

    if (ingresoSeleccionado.IdCuenta == idCuentaCredito)
    {
        mensajeError = "La cuenta débito y crédito no pueden ser la misma.";
        return;
    }

    await IngresoService.Crear(ingresoSeleccionado, idCuentaCredito);
    await CargarDatos();
    CerrarModal();
    guardando = false;
}

private async Task ConciliarIngreso(int id)
{
    await IngresoService.CambiarEstado(id, "Conciliado");
    await CargarDatos();
}

private async Task EliminarIngreso(int id)
{
    await IngresoService.Eliminar(id);
    await CargarDatos();
}

private void CerrarModal()
{
    mostrarModal = false;
    mensajeError = string.Empty;
}
}
```

Fuente: Elaboración propia

El módulo de registrar los ingresos de la empresa se demuestra en el fragmento de código expuesto en la figura 50, en donde el módulo permite guardar los ingresos, visualizarlos y editarlos, con la opción de conciliar el ingreso.

Figura 51
Módulo Presupuesto Mensual

```
private async Task CargarDatos()
{
    cargando = true;
    presupuestos = await PresupuestoMensualService.ObtenerTodos();
    cuentas = await CuentaService.ObtenerActivas();
    cargando = false;
}

private void MostrarFormulario()
{
    presupuestoSeleccionado = new PresupuestoMensualModel
    {
        Fecha = DateTime.Today,
        Activo = true,
        Anio = DateTime.Today.Year,
        IdUsuario = idUsuarioActual
    };
    mensajeError = string.Empty;
    mostrarModalNuevo = true;
}

private void VerDetalle(Proyecto.TFG.Models.PresupuestoMensualModel presupuesto)
{
    presupuestoSeleccionado = presupuesto;
    mostrarModalDetalle = true;
}

private async Task GuardarPresupuesto()
{
    if (guardando) return;
    guardando = true;

    if (string.IsNullOrEmpty(presupuestoSeleccionado.Mes) || string.IsNullOrEmpty(presupuestoSeleccionado.Descripcion) || presupuestoSeleccionado.TotalIngresos <= 0)
    {
        mensajeError = "Por favor complete todos los campos obligatorios.";
        guardando = false;
        return;
    }

    await PresupuestoMensualService.Crear(presupuestoSeleccionado);
    await CargarDatos();
    CerrarModal();
    guardando = false;
}

private async Task DesactivarPresupuesto(int id)
{
    await PresupuestoMensualService.Desactivar(id);
    await CargarDatos();
}

private void CerrarModal()
{
    mostrarModalNuevo = false;
    mostrarModalDetalle = false;
    mensajeError = string.Empty;
}
```

Fuente: Elaboración propia

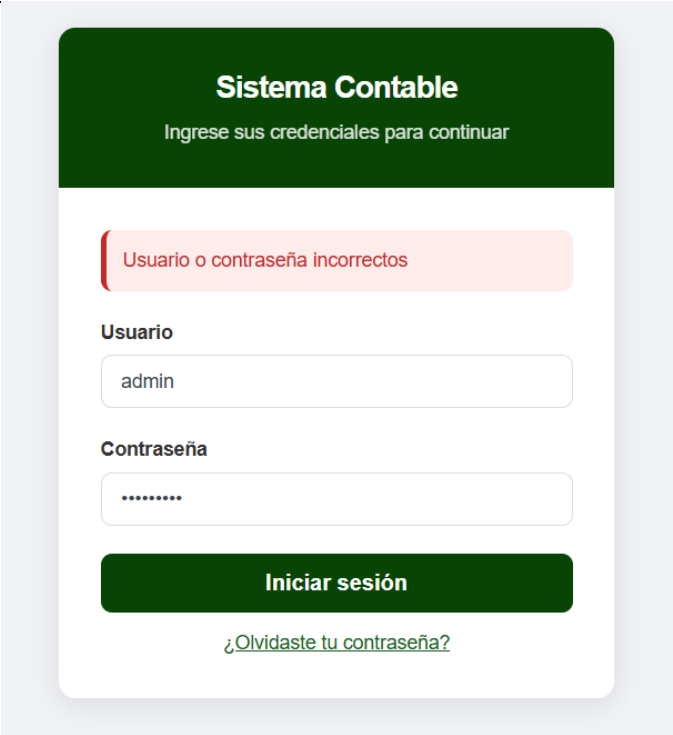
El módulo de presupuesto mensual se demuestra en la figura 51, permitiendo al usuario agregar el presupuesto estimado del mes y realizar comparaciones, así como consultar y visualizar reportes de balance general.

Pruebas

En esta sección se describen las pruebas realizadas al sistema, con el objetivo de verificar el comportamiento y funcionamiento del sistema. Las pruebas permiten identificar errores que se puedan presentar en distintos escenarios, validar los principales procesos, y garantizar el correcto funcionamiento de los módulos del sistema.

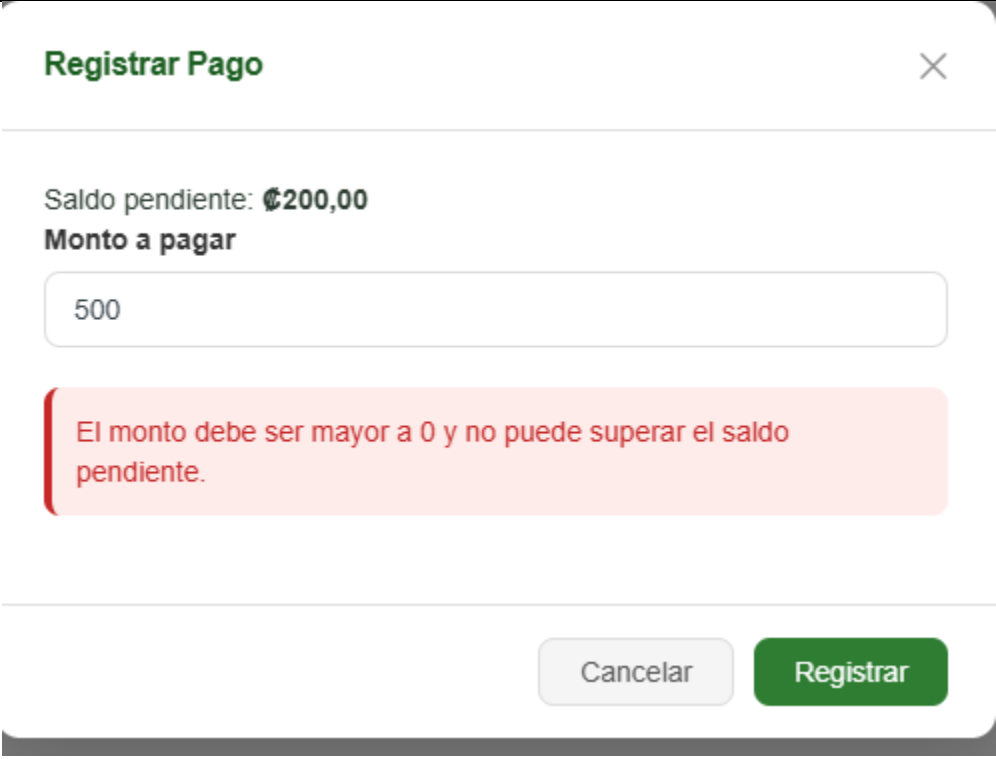
Tabla 50

Prueba Iniciar Sesión

Verificación de inicio de sesión	
Caso de prueba: 01	Módulo por probar: Inicio de sesión.
Descripción	
Se realiza la validación del sistema a la hora de ingresar credenciales incorrectas al sistema.	
Resultado deseado	
Ingreso denegado por credenciales incorrectas.	
Prueba realizada:	Se digitan credenciales no existentes.
Resultado obtenido	
	

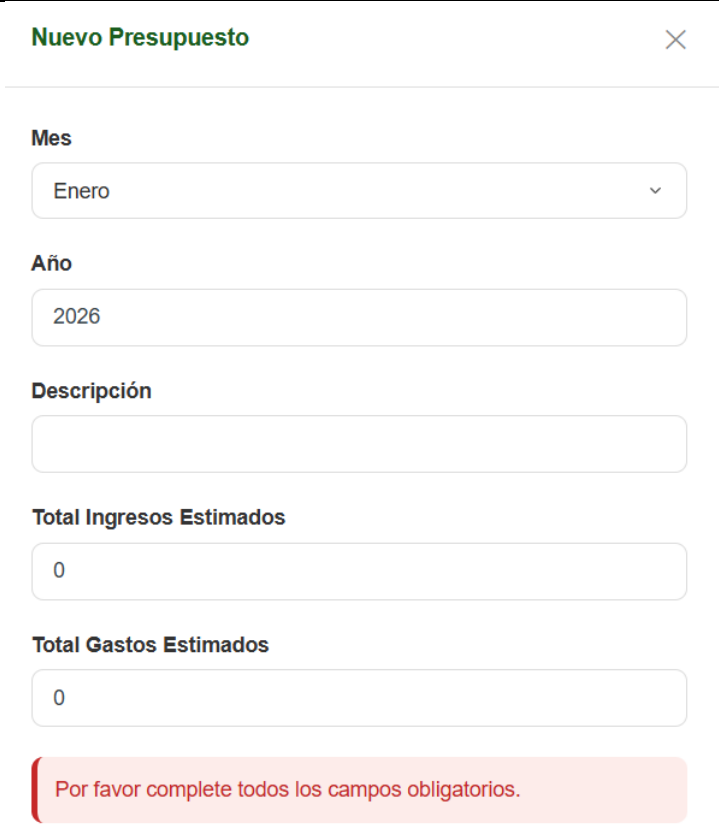
Fuente: Elaboración propia

Tabla 51*Prueba Cuenta por pagar*

Verificación de cuenta por pagar	
Caso de prueba: 02	Módulo por probar: Cuenta por pagar.
Descripción	
Se realiza la validación del sistema a la hora de ingresar un monto a depositar mayor al saldo pendiente.	
Resultado deseado	
No se realiza el pago, debido a que el monto ingresado es mayor al monto pendiente.	
Prueba realizada:	Se digita un monto a pagar mayor al pendiente.
Resultado obtenido	
	

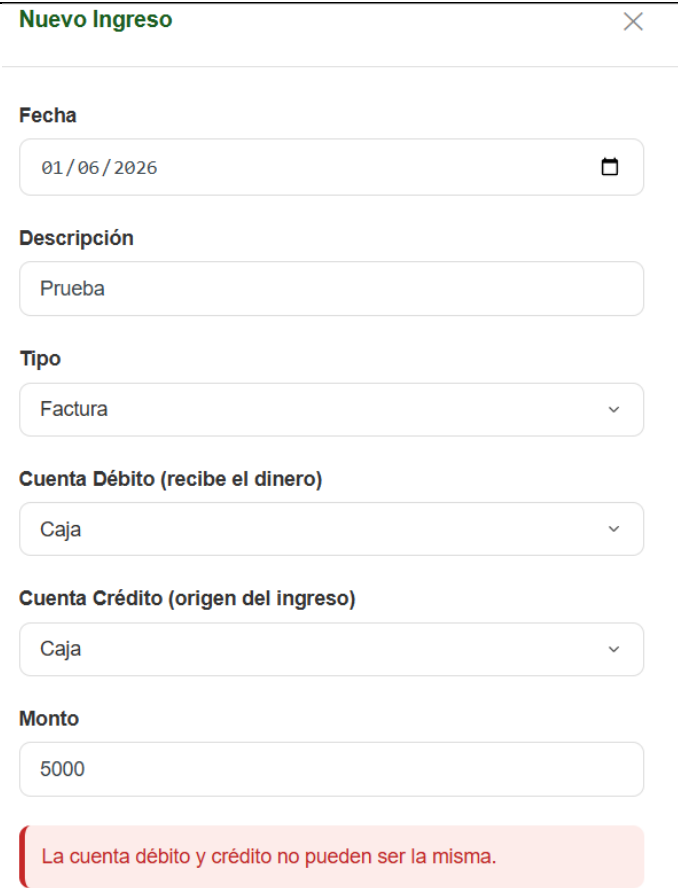
Fuente: Elaboración propia

Tabla 52*Prueba Ingresar datos vacíos*

Verificación de ingreso de datos vacíos	
Caso de prueba: 03	Módulo por probar: Presupuesto mensual.
Descripción	
Se realiza la validación del sistema a la hora de intentar guardar datos vacíos.	
Resultado deseado	
El sistema no permite guardar la información, debido a que los campos no pueden estar vacíos.	
Prueba realizada:	Se intenta guardar un presupuesto mensual con datos vacíos.
Resultado obtenido	
 The screenshot shows a web form titled 'Nuevo Presupuesto' with a close button (X) in the top right corner. The form contains several input fields: a dropdown menu for 'Mes' (Month) with 'Enero' selected, a text input for 'Año' (Year) with '2026' entered, an empty text input for 'Descripción', a text input for 'Total Ingresos Estimados' with '0' entered, and another text input for 'Total Gastos Estimados' with '0' entered. At the bottom of the form, there is a red error message box that reads: 'Por favor complete todos los campos obligatorios.' (Please complete all required fields).	

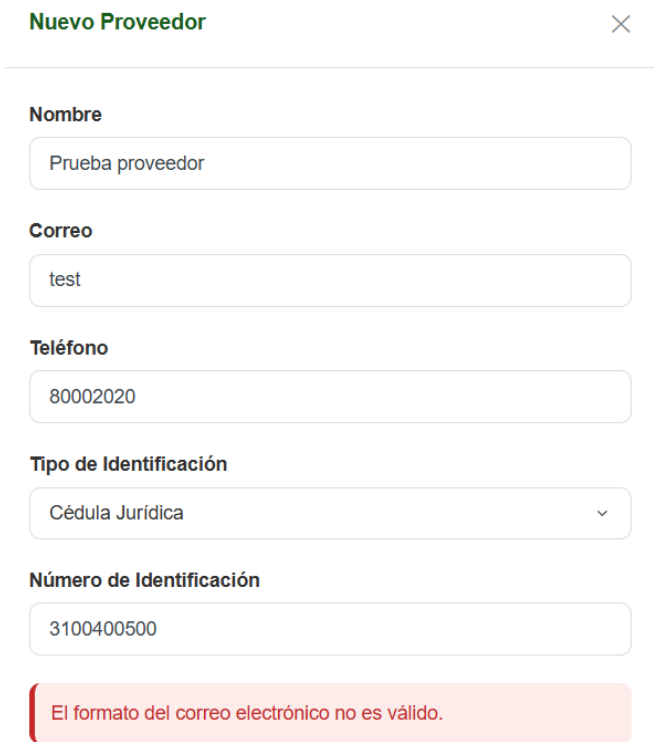
Fuente: Elaboración propia

Tabla 53*Prueba cuentas iguales*

Verificación de cuentas iguales	
Caso de prueba: 04	Módulo por probar: Ingresos.
Descripción	
Se realiza la validación del sistema a la hora de ingresar un registro en el módulo de Ingresos en donde la cuenta de débito y crédito son la misma.	
Resultado deseado	
El sistema no permite guardar la información, debido a que las cuentas de débito y crédito deben ser distintas.	
Prueba realizada:	Se intenta guardar un ingreso con cuentas de débito y crédito idénticas
Resultado obtenido	
	

Fuente: Elaboración propia

Tabla 54*Prueba formato de correo*

Verificación de formato correo electrónico	
Caso de prueba: 05	Módulo por probar: Registro proveedor
Descripción	
Se realiza la validación del sistema a la hora de registrar un nuevo proveedor con el formato de correo electrónico incorrecto.	
Resultado deseado	
El sistema no permite guardar la información, debido a no se respeta el tipo de formato para el campo de correo electrónico	
Prueba realizada:	Se intenta guardar un proveedor con el campo de correo electrónico en un formato incorrecto.
Resultado obtenido	
 The screenshot shows a web form titled 'Nuevo Proveedor' with a close button (X). The form contains several input fields: 'Nombre' (with value 'Prueba proveedor'), 'Correo' (with value 'test'), 'Teléfono' (with value '80002020'), 'Tipo de Identificación' (a dropdown menu with 'Cédula Jurídica' selected), and 'Número de Identificación' (with value '3100400500'). At the bottom, a red error message box states: 'El formato del correo electrónico no es válido.'	

Fuente: Elaboración propia

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Se logró cumplir de manera satisfactoria con los objetivos planteados al iniciar el proyecto. El objetivo general de desarrollar un sistema de gestión contable y facturación para la soda Creston fue alcanzado mediante la implementación de una solución tecnológica funcional y adaptada a las necesidades de la empresa. En lo que respecta a los objetivos específicos se realizó el debido análisis de requerimientos permitiendo identificar las problemáticas existentes. El diseño de la base de datos se desarrolló satisfactoriamente utilizando la herramienta MySQL Workbench, así como la lógica de la aplicación que, mediante herramientas como C# y Blazor se desarrollaron los módulos funcionales del sistema y finalmente, mediante pruebas de validación y verificación se probaron los componentes del sistema.

El desarrollo del sistema de gestión contable y de facturación evidenció la importancia de contar con herramientas y soluciones tecnológicas en las organizaciones. La implementación de esta solución no se resume únicamente en resolver las problemáticas identificadas en la soda Creston, sino también sienta las bases para una gestión profesional y ordenada del negocio. La automatización y digitalización de procesos contables y administrativos representan una ventaja estratégica que permite a las organizaciones tomar decisiones oportunas, reduciendo la dependencia de procedimientos manuales, los cuáles son propensos a errores.

La automatización de procesos y de cálculos que se realizan a nivel interno en el sistema, demuestra la importancia de optimizar las operaciones de una empresa para mejorar su eficiencia. En el caso de la soda Creston, los procesos realizados anteriormente de forma manual, ahora se ejecutan de forma automática, de esta forma, no solo se reduce de manera significativa el error humano, sino que libera al personal de tiempo que pueden enfocarlo en actividades que representen un mayor valor estratégico para el negocio.

Al realizar el análisis de requerimientos, se evidenció que la empresa presentaba ciertas problemáticas para llevar el control de registros, con respecto a productos, cuentas por pagar, y cobrar. Estas problemáticas se lograron identificar, analizar y se convirtieron en módulos y mantenimientos funcionales que el usuario puede manipular. De esta forma se evidencia la mejora en la administración del negocio, ya que el personal cuenta con herramientas que les permite

consultar en tiempo real el estado de inventario, cuentas por pagar con proveedores, o saldos pendientes de clientes, facilitando la toma de decisiones operativas.

El desarrollo de la solución, utilizando el patrón de diseño MVC, permite la separación de las capas del proyecto, la lógica de negocio, el acceso a los datos y la interfaz de usuario. Esta separación tiene implicaciones importantes para el mantenimiento y escalabilidad del sistema, permitiendo modificar alguna capa sin afectar a las demás. El uso de herramientas como Blazor Server y Entity Framework que se integran de forma nativa con el ecosistema Microsoft, agilizó el desarrollo y garantizó una solución robusta, segura y escalable.

Igualmente, la fase de pruebas fue fundamental para validar y verificar que el sistema desarrollado cumple con los requisitos previamente establecidos. Las pruebas realizadas abarcaron escenarios incluyendo validaciones de entrada de datos, control de acceso por roles o manejo de excepciones que tenían el objetivo de comprobar el funcionamiento del sistema ante excepciones o errores que se podrían presentar, ante esto, el sistema respondió favorablemente en los escenarios evaluados, demostrando un manejo adecuado de errores y excepciones que garantiza la estabilidad y confiabilidad de la solución en un ambiente de uso real

Finalmente, el sistema representa una solución a problemáticas que enfrenta la empresa en su día a día. Al fortalecer el orden de la información, automatizar procesos contables y proporcionar reportes actualizados, el sistema se convierte en una herramienta de valor estratégico que fundamenta la toma de decisiones en datos precisos y actualizados. El acceso a información confiable sobre ingresos, gastos, balances y presupuestos permite a la administración del negocio identificar tendencias y tomar acciones, contribuyendo así al crecimiento sostenible y fortalecimiento de la gestión administrativa de la organización.

Recomendaciones

Para garantizar que el sistema se utilice correctamente, se recomienda a la administración de la soda Creston organizar sesiones de capacitación especializadas para cada perfil de usuario del sistema. Estas sesiones deben realizarse mediante talleres prácticos donde cada usuario interactúe directamente con el sistema en los módulos correspondientes a su rol, utilizando datos de prueba para familiarizarse con las funcionalidades disponibles, garantizando que el personal esté preparado para utilizar el sistema desde el primer día.

Se recomienda a la empresa implementar políticas formales de seguridad de la información, dirigida a los usuarios que tengan acceso al sistema, para proteger la confidencialidad de la información financiera. Dicha política debe establecerse mediante un documento interno que defina las normas de uso del sistema, incluyendo la obligatoriedad de crear contraseñas seguras, siguiendo patrones de seguridad como contar con al menos ocho caracteres, combinando letras, números y caracteres especiales, la prohibición de compartir credenciales entre usuarios y obligación de cerrar sesión al abandonar el puesto de trabajo, además de una política de cambio de contraseña de manera periódica para todos los usuarios. Estas políticas deben estar vigentes desde el inicio de operaciones del sistema.

Se recomienda a la empresa, que, al manejar información financiera, establecer una rutina de respaldo de la base de datos de manera semanal, especialmente al finalizar la jornada laboral del último día hábil de cada semana, estos respaldos los debe ejecutar el departamento de TI a cargo del mantenimiento del sistema. Los respaldos deben almacenarse de manera que, ante cualquier situación adversa, como fallo de hardware o pérdida de datos, se pueda recuperar la información. El archivo de respaldo debe almacenarse en un dispositivo de almacenamiento externo como una memoria USB o disco duro portátil, o en una plataforma de almacenamiento en la nube como Google Drive u OneDrive. Esta práctica debe implementarse desde el primer día de operaciones del sistema.

Se recomienda a la empresa realizar encuestas de satisfacción a todos los usuarios del sistema cada seis meses, con el objetivo de medir el impacto del sistema en los procesos operativos y detectar oportunidades de mejora. Estas encuestas deben aplicarse mediante formularios digitales

distribuidos por correo electrónico, incluyendo preguntas sobre la facilidad de uso, utilidad de módulos, errores encontrados y funcionarios que los usuarios consideren necesarias. Los resultados deben ser analizados por el administrador del sistema y compartidos con el encargado del desarrollo para priorizar mejoras en futuras versiones.

Referencias

- Amazon Web Services. (s. f.-b). *What is SQL (Structured Query Language)?* Amazon Web Services, Inc. <https://aws.amazon.com/what-is/sql/>
- Barriuso, R. (2024, 25 enero). *C# (C Sharp): Qué es, dónde se utiliza y para qué sirve*. TRBL Services. <https://trbl-services.eu/blog-c-sharp-que-es-para-que-sirve/>
- Berumen. (2025, 30 julio). *Como la investigación exploratoria abre nuevas oportunidades en el mercado*. Berumen. <https://berumen.com.mx/que-es-y-para-que-sirve-la-investigacion-exploratoria/#:~:text=%C2%BFQu%C3%A9%20es%20y%20para%20qu%C3%A9%20sirve%20la%20investigaci%C3%B3n%20exploratoria?,estrat%C3%A9gicas%20basadas%20en%20hallazgos%20preliminares.>
- Chica, M. (2026, 23 enero). *¿Qué es una aplicación web y para qué sirve?* cdmon. <https://www.cdmon.com/es/blog/aplicacion-web-que-es>
- Cristiá, M. (2024). *Introducción a la Ingeniería de Requerimientos*. Centro Internacional Franco-Argentino de Ciencias de la Información y de Sistemas (CIFASIS). Argentina
- Darias, S. (s. f). *¿Qué es Microsoft SQL Server y para qué sirve?* Intelequia. <https://intelequia.com/es/blog/post/qu%C3%A9-es-microsoft-sql-server-y-para-qu%C3%A9-sirve>
- Duarte, D., & Guerrero, R. (2024). La encuesta como instrumento de recolección de datos, confiabilidad y validez en investigación científica. *Revista de Ciencias Empresariales, Tributarias, Comerciales y Administrativas*, 3(2), 94-107. Recuperado a partir de <https://educaciontributaria.com.py/revista/index.php/rcetca/article/view/70>
- Duque, M. (2024). *Diseño de Bases de Datos: Fundamentos y Aplicaciones*. <https://www.manuduke.com/disenio-bases-datos-fundamentos-aplicaciones/>
- Gti. (s. f.). *Factura electrónica*. <https://www.facturaelectronica.cr/>
- Hecker, J., & Kalpokas, N (2025, 11 febrero). *Análisis de Entrevistas. Técnicas y métodos*. <https://atlasti.com/es/guias/guia-del-analisis-de-entrevistas/analisis-de-entrevistas>

- Holdsworth, J., & Kosinski, M. (2025, 27 noviembre). Seguridad de la información. *¿Qué es la seguridad de la información (InfoSec)?* <https://www.ibm.com/es-es/think/topics/information-security>
- López, J. (2025, 1 agosto). Economipedia. Población estadística: Qué es, tipos y ejemplos. <https://economipedia.com/definiciones/poblacion-estadistica.html>
- López, J. (2025, 4 agosto). Economipedia. Muestra estadística: Qué es, tipos y ejemplos. <https://economipedia.com/definiciones/muestra-estadistica.html>
- Maranto, M., González, M (2015). Fuentes de Información. Universidad autónoma del estado de hidalgo. <http://www.uaeh.edu.mx/virtual>
- MDN (2023, 13 noviembre). *MVC - Glosario de MDN Web Docs* <https://developer.mozilla.org/es/docs/Glossary/MVC>
- Medina, M., Hurtado, D., Muñoz, J., Ochoa, D & Izundegui, G. (2023). Método mixto de investigación: Cuantitativo y cualitativo. Instituto Universitario de Innovación Ciencia y Tecnología Inudi Perú S.A.C. <https://doi.org/10.35622/inudi.b.105>
- Miro. (s. f.). *Cómo crear un presupuesto mensual* | Miro. <https://miro.com/.https://miro.com/es/gestion-proyectos/que-es-presupuesto-mensual/>
- Moonflow (2023, 4 noviembre). Blog de Cobranza, Recuperación de Pagos y Tecnología. Cuentas por cobrar ¿Qué son? <https://www.moonflow.ai/blog/cuentas-por-cobrar-que-son>
- Montesdeoca del Puerto, E. (s. f.). Introducción a Blazor: ¿Qué es y cómo crear una aplicación en Blazor? *Intelequia*. <https://intelequia.com/es/blog/post/introducci%C3%B3n-a-blazor-qu%C3%A9-es-y-c%C3%B3mo-crear-una-aplicaci%C3%B3n-en-blazor>
- Muguira, A. (2025, 14 mayo). *¿Qué es una entrevista? Tipos, ejemplos y cómo responder*. QuestionPro. <https://www.questionpro.com/blog/es/tecnicas-de-recoleccion-de-datos-entrevista/>
- Muñoz, A. (2024). International Business School. La lógica de programación y su relación con los fundamentos de la programación. <https://eiposgrados.com/blog-desarrollo-web-fullstack/logica-programacion/>

- Oracle América Latina. (s. f.). ¿Qué es el software de contabilidad? <https://www.oracle.com/latam/erp/what-is-accounting-software/>
- Ostrowska, K. (2024, 20 agosto). *¿Qué es el CRUD?* LearnSQL.es. <https://learnsql.es/blog/que-es-el-crud/>
- Parra, A. (2023, 7 agosto). *Cuáles son los tipos de variables en una investigación*. QuestionPro. <https://www.questionpro.com/blog/es/tipos-de-variables-en-una-investigacion/#:~:text=Una%20variable%20de%20investigaci%C3%B3n%20o,un%20experimento%20comprobando%20los%20resultados.>
- Quiroz, C. (s. f.). ¿Qué es el desarrollo de software? <https://www.ibm.com/es-es/think/topics/software-development>
- Sage. (2025, 12 marzo). *Cuentas Contables: Definición, tipos y ejemplos prácticos* | Sage. Sage Advice España. <https://www.sage.com/es-es/blog/diccionario-empresarial/cuentas-contables/>
- SalusPlay. (s. f.). Las variables de investigación. <https://www.salusplay.com/apuntes/apuntes-metodologia-de-la-investigacion/tema-2-las-variables-de-investigacion>
- Sampieri, R., Fernández, C., & Baptista, L. (2014). *Metodología de la Investigación*. Mc Graw HILL. Sexta Edición. México.
- Sampieri, R., Fernández, C., & Baptista, L. (2014). Definiciones de los enfoques cuantitativo y cualitativo, sus similitudes y diferencias. *RH Sampieri, Metodología de la Investigación*, 22.
- Smith, S (n. f). *Información general de ASP.NET Core MVC*. Microsoft Learn. https://learn.microsoft.com/es-es/aspnet/core/mvc/overview?view=aspnetcore-10.0&WT.mc_id=dotnet-35129-website
- Square (s. f). *Glosario para negocios. ¿Qué es una transacción?* <https://squareup.com/es/es/glossary/transaccion>
- Stewart, L. (2025, 11 febrero). *¿Qué es la investigación descriptiva y cómo se utiliza?* ATLAS.ti. <https://atlasti.com/es/research-hub/investigacion-descriptiva>

- Stripe*. (2024, 5 noviembre). Guía de los sistemas de facturación. <https://stripe.com/es/resources/more/invoicing-systems-101-what-they-are-how-they-work-and-how-to-choose-one>
- Ubaldo, E., Zully, A. (2008). Fuentes de información para la recolección de información cuantitativa y cualitativa.
- Úrzua, F. (2026). ¿Qué son las cuentas por pagar y cómo se clasifican? Mi sitio de Framer. <https://www.chipax.com/es/blog/cuentas-por-pagar>
- Vásquez, A. (2023, 22 febrero). *Sobre las técnicas e instrumentos para la recolección de información (en la perspectiva metodológica cuantitativa)*. Red Holos XXI. <https://redholosxxi.com/sobre-las-tecnicas-e-instrumentos-para-la-recoleccion-de-informacion/>
- Vieira, D. (2024, 21 octubre). Prototipo: ¿qué es y cuál es su importancia en la tecnología? *HostGator*. <https://www.hostgator.mx/blog/prototipo-que-es/>
- Wadepickett. (s. f.). *Introducción a ASP.NET Core SignalR*. Microsoft Learn. https://learn.microsoft.com/es-es/aspnet/core/signalr/introduction?view=aspnetcore-10.0&WT.mc_id=dotnet-35129-website

Apéndices

Apéndice A: Guía de Entrevista

Entidad:	Soda Creston
Nombre del entrevistado	Óscar Delgado Naranjo
Puesto del entrevistado	Encargado de la soda
Nombre del estudiante:	Alexis Francisco Mora Mora
Fecha de la entrevista	17/04/2026
Lugar o medio de la entrevista:	Escuela Creston

Fuente: Elaboración propia

Preguntas:

Pregunta 1

¿Cómo realiza la gestión contable de la empresa?

Pregunta 2

¿Quiénes están encargados de realizar el proceso de gestión contable?

Pregunta 3

¿Cómo se realiza la facturación actualmente?

Pregunta 4

¿Cómo se registran las transacciones realizadas por los clientes?

Pregunta 5

¿Cómo gestiona la cantidad de pedidos que recibe diariamente?

Pregunta 6

¿Qué dificultades presentan con el sistema actual de gestionar la contabilidad y emisión de facturas?

Pregunta 7

¿Considera que un sistema de gestión contable y de facturación mejore el proceso actual?

Apéndice B: Cuestionario

En el marco de una investigación sobre la gestión contable y de facturación, le invitamos a completar este cuestionario. Su participación es de gran importancia para comprender cómo el tema en estudio influye en la actividad de la organización.

Este cuestionario es confidencial. Sus respuestas solo se utilizarán con fines académicos y no serán compartidos con otras personas. Completar el cuestionario tomará aproximadamente 10 minutos.

Pregunta 1

En una escala del 1 al 5, ¿considera efectivo el método actual en que la empresa realiza la gestión contable?

- 1 (muy poco efectivo)
- 2 (poco efectivo)
- 3 (moderado)
- 4 (efectivo)
- 5 (muy efectivo)

Pregunta 2

¿Qué tan frecuente se generan facturas en la empresa?

- Nunca
- Casi nunca
- Ocasionalmente
- Varias veces al día
- Con mucha frecuencia

Pregunta 3

¿Cómo considera el tiempo que se toma en realizar una factura?

- 1 (muy lento)
- 2 (lento)

- 3 (moderado)
- 4 (rápido)
- 5 (muy rápido)

Pregunta 4

¿Con qué frecuencia se presentan errores en el proceso de facturación?

- Nunca
- Pocas veces
- Frecuentemente
- Casi siempre
- Siempre

Pregunta 5

En una escala del 1 al 5, ¿considera efectivo el método de gestionar pedidos por parte de los clientes?

- 1 (muy poco efectivo)
- 2 (poco efectivo)
- 3 (moderado)
- 4 (efectivo)
- 5 (muy efectivo)

Pregunta 6

En una escala del 1 al 5, ¿considera efectiva la gestión de pedidos a los proveedores?

- 1 (muy poco efectivo)
- 2 (poco efectivo)
- 3 (moderado)
- 4 (efectivo)
- 5 (muy efectivo)

Pregunta 7

En una escala del 1 al 5, ¿considera efectiva la gestión de las cuentas por cobrar a los clientes?

- 1 (muy poco efectivo)
- 2 (poco efectivo)
- 3 (moderado)
- 4 (efectivo)
- 5 (muy efectivo)

Pregunta 8

¿Se encuentra satisfecho con el sistema actual con que la empresa opera las gestiones contables y de facturación?

- Sí
- No

Pregunta 9

¿Considera que un sistema automatizado sería de ayuda para reducir errores en el proceso de facturación?

- Desacuerdo
- Neutral
- De acuerdo

Pregunta 10

¿Considera que un sistema automatizado mejoraría la gestión contable de la empresa?

- Desacuerdo
- Neutral
- De acuerdo