

UNIVERSIDAD INTERNACIONAL DE LAS AMÉRICAS

FACULTAD DE INGENIERÍA

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

Título de la investigación:

Sistema web para la gestión contable de la empresa Ópticas Dr. Max, ubicada en Heredia

Nombre del estudiante:

Tyron López Chinchilla

Tutor:

MBD. Olman Núñez Peralta

San José

23 de Julio del 2026

Contenido

Contenido	2
Dedicatoria	13
Agradecimiento.....	14
APROBACIÓN DEL TRIBUNAL EXAMINADOR.....	16
CAPÍTULO I: Introducción	17
Planteamiento del problema	17
Objetivos	17
Objetivo general	17
Objetivos específicos	17
Justificación.....	18
Viabilidad técnica.....	18
Viabilidad operativa.....	20
Viabilidad económica.....	20
Viabilidad legal.....	21
Proyecciones	22
Alcance funcional	22
Alcance metodológico.....	25
Alcance tecnológico	26
Capítulo II: Marco referencial	28
Relevancia en la automatización de los sistemas contables	28
Características de un sistema contable	29
Conceptos de un sistema contable	30
Aspectos legales de un sistema contable en Costa Rica	32
Metodología cascada (waterfall).....	34
Software en un sistema contable	34
Visual Studio como motor de desarrollo	35
Componentes por utilizar en Visual Studio.....	36
Lenguaje de programación C-Sharp	37
Modelo MVC (modelo-vista-controlador).....	38
Componentes del front-end.....	39
Entity Framework.....	39

Base de datos relacional y su utilidad en un sistema contable	40
SQL Management Studio	41
Capítulo III: Marco metodológico	43
Enfoques de investigación.....	43
Enfoque cuantitativo.....	43
Enfoque cualitativo	44
Enfoque mixto.....	46
Enfoque de investigación seleccionado.....	46
Tipos de investigación	47
Investigación exploratoria	47
Investigación descriptiva	49
Investigación explicativa.....	50
Tipo de investigación seleccionado	50
Fuentes de información	51
Fuentes primarias.....	51
Fuentes secundarias	52
Fuentes terciarias	52
Según University of Minnesota Crookston (2026), las fuentes terciarias son aquellas que:	52
Variables	53
Variables conceptuales.....	53
Variables instrumentales	53
Variables operacionales	54
Instrumentos de recolección de datos	57
La observación.....	57
La entrevista	58
Proceso para la recolección y el análisis de datos	59
Capítulo IV: Análisis de los resultados.....	60
Análisis del instrumento de la entrevista	60
Pregunta n.º 1:	60
Pregunta n.º 2:	60
Pregunta n.º 3:	61
Pregunta n.º 4:	61
Pregunta n.º 5:	62

Pregunta n.º 6:	62
Pregunta n.º 7:	63
Pregunta n.º 8:.....	63
Pregunta n.º 9:	64
Pregunta n.º 10:	65
Análisis del instrumento: la observación	65
Capítulo V: Propuesta.....	73
Requerimientos del sistema	73
Gestionar cuentas contables	73
Gestionar facturación	74
Gestión de cuentas por cobrar	75
Gestión de cuentas por pagar	75
Gestionar los fondos de la caja chica	76
Gestionar cierre de caja	76
Calcular los gastos.....	76
Gestión del presupuesto	76
Gestión de ingresos.....	77
Seguridad	77
Consultas.....	77
Reportes	77
Mantenimientos	77
Matriz de requerimientos	78
Análisis detallado del software	79
Análisis detallado del hardware	80
Análisis detallado de los elementos que se relacionan con las telecomunicaciones	81
Descripción detallada de las herramientas técnicas que se utilizan	82
Casos de uso.....	82
Arquitectura del sistema.....	123
Arquitectura del software.....	124
Diseño de entradas	126
Diseño físico de entradas	137
Diseño físico de la base de datos	150
Diseño de procesos	151
Diseño de salidas	160

Diseño de UML	169
Programación/entradas	169
Programación/salidas	175
Programación/procesos	181
Programación/validaciones.....	185
Programación/módulos señalados en el alcance	192
Pruebas	201
Capítulo VI. Conclusiones y recomendaciones	213
Conclusiones	213
Conclusión 1: analizar los requerimientos funcionales y no funcionales	213
Conclusión 2: construir el diseño de la base de datos mediante el uso de MySQL Workbench.....	214
Conclusión 3: programar los módulos mediante el uso de C# y con la ayuda de Entity Framework.....	214
Conclusión 4: probar los diferentes módulos de forma independiente y también de manera integral, garantizando los resultados correctos del sistema	214
Recomendaciones.....	215
Recomendación 2: establecer redundancia de enlaces de conexión a Internet y a la base de datos.....	215
Recomendación 3: migrar el sistema a un entorno de producción con servidor dedicado y certificado TLS	215
Recomendación 4: implementar un plan de contingencia y recuperación ante desastres	216
Referencias	217
Anexos.....	221
Guía de la entrevista	221
Guía de la observación	223

Tablas

Tabla 1 Viabilidad técnica.....	33
Tabla 2 Viabilidad económica.....	34
Tabla 3 Módulos del sistema	36
Tabla 4 Conceptos de la contabilidad	44
Tabla 5 Variables.....	68
Tabla 6 Matriz de requerimientos	92
Tabla 7 Caja Chica 1	96
Tabla 8 Caja Chica 2.....	98
Tabla 9 Caja Chica 3.....	99
Tabla 10 Cierre de caja 1	101
Tabla 11 Cierre de caja 2	102
Tabla 12 Cierre de caja 3	104
Tabla 13 Calcular los gastos 1	105
Tabla 14 Calcular los gastos 2	107
Tabla 15 Calcular los gastos 3	108
Tabla 16 Gestionar facturación 1	110
Tabla 17 Gestionar facturación 2	111
Tabla 18 Gestionar facturación 3	112
Tabla 19 Gestión de Cuentas por pagar 1	114
Tabla 20 Gestión de Cuentas por pagar 2	115
Tabla 21 Gestión de Cuentas por pagar 3	117
Tabla 22 Gestión del presupuesto 1	118
Tabla 23 Gestión del presupuesto 2	120
Tabla 24 Gestión del presupuesto 3	121
Tabla 25 Gestión de ingresos 1	123
Tabla 26 Gestión de ingresos 2	124
Tabla 27 Gestión de ingresos 3	126
Tabla 28 Gestionar cuentas contables 1	127
Tabla 29 Gestionar cuentas contables 2	129
Tabla 30 Gestionar cuentas contables 3	130
Tabla 31 Gestión de cuentas por cobrar 1	132

Tabla 32	<i>Gestión de cuentas por cobrar 2</i>	133
Tabla 33	<i>Gestión de cuentas por cobrar 3</i>	135
Tabla 34	<i>Catalogo contable agregar</i>	139
Tabla 35	<i>Catalogo contable modificar</i>	139
Tabla 36	<i>Registro de Nueva Factura</i>	140
Tabla 37	<i>Registro de modificar Factura</i>	141
Tabla 38	<i>Registro de agregar ingreso</i>	142
Tabla 39	<i>Registro de modificar ingreso</i>	142
Tabla 40	<i>Registro de agregar gasto</i>	143
Tabla 41	<i>Registro de modificar gasto</i>	143
Tabla 41	<i>Registro de abono de cuentas por cobrar</i>	143
Tabla 42	<i>Agregar cuentas por pagar</i>	144
Tabla 43	<i>Modificar cuentas por pagar</i>	145
Tabla 44	<i>Agregar abonos a cuentas por pagar</i>	145
Tabla 45	<i>Agregar monto Caja chica</i>	146
Tabla 46	<i>Modificar monto Caja chica</i>	146
Tabla 47	<i>Gestionar Cierre de caja</i>	147
Tabla 48	<i>Modificar Cierre de caja</i>	147
Tabla 49	<i>Gestionar presupuesto</i>	148
Tabla 50	<i>Editar presupuesto</i>	148

Figuras

Figura 1 Etapas de la metodología en cascada.....	39
Figura 2 <i>Sintaxis C-Sharp</i>	50
Figura 3 <i>MVC</i>	51
Figura 4 <i>SQL Sever Management Studio</i>	55
Figura 5 Observación de la caja chica	79
Figura 6 Registro de gastos	80
Figura 7 Creación de facturas	81
Figura 8 Registro de ingresos	82
Figura 9 Cuentas por pagar	83
Figura 10 Cuentas por cobrar	84
Figura 11 Control de presupuesto	85
Figura 12 Arquitectura del sistema.....	136
Figura 13 Arquitectura del software.....	137
Figura 14 <i>Diseño interfaz catalogo contable</i>	148
Figura 15 <i>Diseño interfaz catalogo contable editar</i>	149
Figura 16 <i>Diseño interfaz factura</i>	150
Figura 17 <i>Diseño interfaz factura cuotas</i>	150
Figura 18 <i>Diseño interfaz ingreso</i>	151
Figura 19 <i>Diseño interfaz ingreso 2</i>	151
Figura 20 <i>Diseño interfaz gastos</i>	152
Figura 21 <i>Diseño interfaz gastos nueva categoría</i>	152
Figura 22 <i>Diseño interfaz cuentas por cobrar</i>	153
Figura 23 <i>Diseño interfaz cuentas por pagar</i>	154
Figura 24 <i>Diseño interfaz cuentas por pagar abono</i>	155
Figura 25 <i>Diseño interfaz cuentas por pagar editar</i>	156
Figura 26 <i>Diseño interfaz caja chica</i>	157
Figura 27 <i>Diseño interfaz caja chica editar</i>	158
Figura 28 <i>Diseño interfaz cierre de caja</i>	159
Figura 29 <i>Diseño interfaz cierre de caja registrar un ingreso manual</i>	159
Figura 30 <i>Diseño interfaz presupuesto</i>	160
Figura 31 <i>Diseño interfaz presupuesto editar</i>	160

Figura 32	<i>Diagrama Workbench</i>	161
Figura 33	<i>Caja chica</i>	162
Figura 34	<i>Cierre de caja</i>	163
Figura 35	<i>Calcular gastos</i>	164
Figura 36	<i>Gestionar facturación</i>	165
Figura 37	<i>Gestión de cuentas por pagar</i>	166
Figura 38	<i>Gestión del presupuesto</i>	167
Figura 39	<i>Gestión de ingresos</i>	168
Figura 40	<i>Gestionar cuentas contables</i>	169
Figura 41	<i>Gestión de cuentas por cobrar</i>	170
Figura 42	<i>Catálogo contable</i>	171
Figura 43	<i>Catalogo contable Excel</i>	171
Figura 44	<i>Catalogo contable PDF</i>	171
Figura 45	<i>Factura</i>	172
Figura 46	<i>Factura Excel</i>	172
Figura 47	<i>Factura PDF</i>	172
Figura 48	<i>Ingreso</i>	173
Figura 49	<i>Ingreso Excel</i>	173
Figura 50	<i>Gastos</i>	173
Figura 51	<i>Gastos Excel</i>	174
Figura 52	<i>Gastos PDF</i>	174
Figura 53	<i>Cuentas por cobrar 1</i>	174
Figura 54	<i>Cuentas por cobrar Excel</i>	174
Figura 55	<i>Cuentas por cobrar PDF</i>	175
Figura 56	<i>Cuentas por pagar 1</i>	175
Figura 57	<i>Cuentas por pagar Excel</i>	175
Figura 58	<i>Cuentas por pagar PDF</i>	175
Figura 59	<i>Caja chica</i>	176
Figura 60	<i>Caja chica Excel</i>	176
Figura 61	<i>Caja chica PDF</i>	176
Figura 62	<i>Cierre de caja</i>	176
Figura 63	<i>Cierre de caja,detalle</i>	177

Figura 64 Cierre de caja PDF	178
Figura 65 Presupuesto.....	178
Figura 66 Presupuesto Excel	178
Figura 67 UML	179
Figura 68 Entradas contabilidad	180
Figura 69 Entradas factura	180
Figura 70 Entradas presupuesto	181
Figura 71 Entradas gastos.....	182
Figura 72 Entradas cuentas por pagar	182
Figura 73 Entradas cuentas por cobrar.....	183
Figura 74 Entradas cierre de caja	184
Figura 75 Entradas caja chica	184
Figura 76 Salidas Contabilidad.....	185
Figura 77 Salidas factura	186
Figura 78 Salidas presupuesto	186
Figura 79 Salidas gastos.....	187
Figura 80 Salidas cuentas cobrar	188
Figura 81 Salidas cuentas pagar.....	189
Figura 82 Salidas cierre de caja.....	190
Figura 83 Salidas caja chica	191
Figura 84 Procesos caja chica	191
Figura 85 Procesos factura	192
Figura 86 Procesos cierre de caja	192
Figura 87 Procesos cuentas por cobrar	193
Figura 88 Procesos catálogo contable	193
Figura 89 Procesos cuentas por pagar	194
Figura 90 Procesos presupuesto	194
Figura 91 Ingresar un ‘ tanto en el usuario como en la contraseña y tratar de ingresar	195
Figura 92 Dejar vacío el usuario y la contraseña y tratar de ingresar	196
Figura 93 Si el usuario es un correo, no poner un @ y darle ingresar	197
Figura 94 Poner la contraseña de un usuario y la clave de otro usuario distinto y darle ingresar.....	198

Figura 95 Ingresar un usuario y contraseña válida.....	199
Figura 96 Ingresar un nuevo usuario con números en el nombre y los apellidos	199
Figura 97 Ingresar un nuevo usuario con fecha de hoy.....	200
Figura 98 Ingresar un nuevo usuario con letras en la identificación.....	201
Figura 99 Crear el usuario y reiniciar la contraseña	201
Figura 100 Controller Presupuesto.....	202
Figura 101 Controller Ingresos.....	203
Figura 102 Controller Gastos.....	204
Figura 103 Controller Facturas	205
Figura 104 Controller CajaChica	206
Figura 105 Controller CierreCaja.....	207
Figura 106 Controller CuentasPagar	208
Figura 107 Controller Contabilidad	209
Figura 108 Controller CuentasCobrar	210
Figura 109 Prueba 1 Presupuesto.....	211
Figura 110 Prueba 2 Presupuesto.....	212
Figura 111 Prueba 1 Contabilidad.....	213
Figura 112 Prueba 2 Contabilidad.....	214
Figura 113 Prueba 1 Cierre de Caja	215
Figura 114 Prueba 2 Cierre de Caja	215
Figura 115 Prueba 1 Caja Chica	216
Figura 116 Prueba 2 Caja Chica	217
Figura 117 Prueba 1 Cuentas por pagar	217
Figura 118 Prueba 2 Cuentas por pagar	218
Figura 119 Prueba 1 Gastos.....	219
Figura 120 Prueba 2 Gastos.....	219
Figura 121 Prueba 1 factura	220
Figura 122 Prueba 2 factura	221
Figura 123 Prueba 1 Cuentas por cobrar	221
Figura 124 Prueba 2 Cuentas por cobrar	222
Figura 125 Prueba 1 Ingresos.....	222
Figura 126 Prueba 2 Ingresos.....	223

Anexos

Entrevista.....	167
Observación	170

Dedicatoria

Yo, Tyron López Chinchilla, le dedico mi trabajo final de graduación a mi abuela Elieth Rodríguez Sequeira, por ser mi guía, mi apoyo incondicional y una de las mayores fuentes de inspiración en mi vida. Gracias por cada consejo, por cada palabra de aliento y por creer en mí incluso en los momentos en que yo dudaba. Este logro no es solo mío, también es tuyo, porque en cada paso que di siempre estuvo tu fe de que yo lograría lo que me proponga.

A ti, que con tu esfuerzo y cariño me enseñaste el valor de la perseverancia, la humildad y el sacrificio, te entrego este trabajo como una pequeña muestra de todo el agradecimiento y amor que llevo en mi corazón. Más que una abuela eres mi madre y siempre voy a estar agradecido por la ayuda que me brindaste, desde pequeño me enseñaste valores, los cuales siempre voy a llevar en mi vida y, más importante, me enseñaste a esforzarme y a seguir adelante con mis metas.

Agradecimiento

Le agradezco profundamente a todas las personas que hicieron posible que yo llegara a estas instancias de mi carrera universitaria. Las personas más importantes en mi vida son mis familiares, sin ellos nada de esto habría sido posible, por ello les debo gran parte de este logro.

A mi papá, le quiero agradecer por su gran esfuerzo al sacarme adelante con mis estudios, para mí es un ejemplo por seguir y un apoyo del cual siempre voy a estar agradecido. A mi mamá, le agradezco por su amor, paciencia y motivación constante. A mi abuela, le quiero agradecer porque sin ella nada de esto sería posible, también por sus consejos y su cariño de madre que me ha dado durante toda mi vida. Mi hermano es una persona a la que me gustaría mencionar en este agradecimiento, ya que él me ayudó mucho en este proceso universitario.

También quiero expresar mi agradecimiento a mi tío, Ricardo López, por su apoyo y por estar presente en momentos importantes de mi formación académica. A todos ustedes, muchas gracias por ser parte fundamental de que hoy me encuentre en este proceso y por ser mi brújula cuando creo estar perdido.

CAPÍTULO I: Introducción

Planteamiento del problema

La empresa que se seleccionó para desarrollar el sistema se llama Ópticas Dr. Max; esta compañía está ubicada en la provincia de Heredia. El propósito es obtener un mejor manejo de la contabilidad, lo que genera una mayor confianza de los clientes y proveedores. Además, se ordenarán los procesos contables, de manera que la entidad cuente con información confiable y estandarizada. Con base en el estudio realizado, se llegó a la conclusión de que la empresa cuenta con algunos problemas críticos en la contabilidad debido a un mal manejo en sus procedimientos. Este mal manejo se relaciona con que no se utiliza un sistema que ayude a los empleados con los procesos contables, lo que llevó a buscar, a través de un sistema web, una solución al proceso de la contabilidad.

Algunos de los problemas que enfrenta la empresa por no contar con un sistema automatizado que ayude con la contabilidad son los siguientes: falta de un manejo adecuado de ingresos, gastos, facturas, caja chica, gestión de cierre de caja y cuentas por pagar. Estos inconvenientes ocasionan molestias, tanto en los clientes como en los trabajadores.

Objetivos

Objetivo general

Desarrollar un sistema web para la gestión de la contabilidad en la empresa Ópticas Dr. Max, utilizando las herramientas de Visual Studio con el lenguaje de C# y la base de datos SQL Server.

Objetivos específicos

Analizar los requerimientos funcionales que debe tener el sistema, junto con los no funcionales.

Construir el diseño de la base de datos mediante el uso de MySQL Workbench.

Programar los módulos mediante el uso de C# y con la ayuda de Entity Framework.

Probar los diferentes módulos de forma independiente y también de manera integral, garantizando los resultados correctos del sistema.

Justificación

El desarrollo de este sistema web se enfoca en resolver algunas de las necesidades de la empresa, debido a que los procesos contables con los que cuenta no se encuentran automatizados ni son exactos respecto a lo que se requiere para administrar adecuadamente la contabilidad empresarial.

Esta solución se planteó para la empresa Ópticas Dr. Max, con el objetivo de que obtengan un sistema que les permita realizar las gestiones contables de manera automatizada. El desarrollo del sistema contable ayuda a un mejor manejo de las facturas, lo que genera una mayor confianza en los clientes. Además, se ordenarán los procesos contables, de forma que la institución cuente con información confiable y estandarizada. Esto garantiza que el servicio se siga brindando a los clientes, también mejora la calidad de la atención ofrecida y permite al personal trabajar en un entorno más organizado, reduciendo la carga de tareas repetitivas y errores manuales, lo que ayuda a disminuir el estrés laboral.

Viabilidad técnica

La solución posee viabilidad técnica, ya que la empresa cuenta con una economía estable que favorece las diferentes etapas del sistema. Asimismo, debido al análisis previo realizado en el establecimiento ubicado en Heredia, se llegó a la conclusión de que este lugar cuenta con una buena infraestructura tecnológica para el funcionamiento correcto del sistema. Esta conclusión se debe a la composición de *hardware* y *software* que tiene la compañía, pues tiene computadoras capaces de ejecutar adecuadamente el programa. Los componentes de *hardware* que la empresa tiene para la debida ejecución del sistema

que se planteó se encuentran en las computadoras de escritorio. Estas cuentan con monitores de 24 in Xiaomi A22i IPS 1920x1080 75 Hz, un procesador Intel Core i5, una memoria RAM de 8 GB, almacenamiento SSD de 250 GB y utilizan Windows 11.

El programa está planeado para que sea escalable a largo plazo y económicamente viable, ya que se piensa usar tecnologías en la nube, lo que facilita la escalabilidad del programa. Algunas de las tecnologías en la nube que se plantearon pueden variar entre Google Cloud o Azure. El sistema utiliza licencias de Visual Studio y SQL Server, las cuales, al día de la elaboración del proyecto, se encuentran gratuitas.

Tabla 1

Viabilidad técnica

Nombre	Descripción	Versión
Visual Studio	Motor de programación que se utiliza para la elaboración del sistema	Se utiliza Visual Studio 2022 17.14
SQL Server Management Studio	Motor de la base de datos para el sistema	Se utiliza la versión 19.0
MySQL Workbench	Motor complementario de la base de datos, se diseña el diagrama de entidad de relaciones	Se usará la versión 8.0 CE
C-Sharp	Lenguaje pensado para la elaboración del sistema	La versión que se utiliza en este lenguaje es la 14.0
Entity Framework	Framework de desarrollo web que se utiliza en la creación del sistema	Versión 6.0
Razor	Motor de plantillas y sintaxis para la elaboración del <i>front-end</i> del sistema	Versión .NET 8
C++	Lenguaje orientado a objetos que ayudará en el <i>back-end</i> del sistema	Versión 17.14

Fuente: Elaboración propia

Viabilidad operativa

El sistema se planteó para que aquellos usuarios que no poseen muchos conocimientos tecnológicos puedan utilizarlo sin mayor complicación. Este planteamiento se pensó para que la persona encargada de utilizar la herramienta no requiera mucho tiempo en capacitaciones, lo que le ahorrará dinero y tiempo a la empresa. No obstante, ya que el sistema manejará temas contables, se espera que el personal que utiliza este sistema cuente con estudios o capacitaciones asociados con la contabilidad. De esta forma, se busca mejorar la eficiencia del personal a cargo, lo que de manera directa eliminará el riesgo que corre la compañía al hacer cálculos manuales, logrando una mejor organización en el ámbito económico en la empresa.

Con base en el estudio realizado de la empresa, la propuesta que se planteó no reduciría el personal encargado de la compañía, debido a que lo único que se pretende con este desarrollo web es lograr la automatización de los procesos contables, agilizando los procedimientos. Esto ayuda a la productividad de los empleados al tener procesos más rápidos y seguros, ya que se dejaría de manejar datos sensibles de forma manual, lo que reduciría errores que llevan a pérdidas.

Viabilidad económica

La Tabla 2 muestra el análisis realizado de los posibles gastos que pueden implicar el desarrollo en las diferentes etapas del sistema web.

Tabla 2

Viabilidad económica

Categoría	Descripción	Costo estimado
Análisis del sistema	Se plantea el análisis del sistema	€995.833 o \$2,000 USD
Diseño del sistema	Se plantea el diseño del sistema	\$1,000 USD
Desarrollo del sistema	Se realiza el desarrollo del <i>back-end</i> y el <i>front-end</i> del	\$3,000 USD

	sistema	
Servidores en la nube	Se aloja el sistema web mediante los servicios de la nube	657.\$00 USD al año (Azure Cloud) \$1,387.44 USD al año (Google Claude)
Pruebas del sistema	Se realiza las pruebas correspondientes del sistema	No tiene costo
Licencias de Visual Studio	Se utiliza el <i>software</i> de Visual Studio con sus respectivas librerías	Actualmente, Visual Studio se encuentra de forma gratuita, por lo que no implica ningún costo, no obstante, en el futuro puede cambiar su precio.

Fuente: Elaboración propia

Con base en la estimación de costos realizada y en el estudio de la empresa, se calcula que los costos mínimos iniciales rondan los \$6,657.00 USD anuales o los \$7,387.55 USD mensuales. El uso del *software* en el desarrollo no tiene costo, ya que al día de la presentación del proyecto este se encuentra gratuito. No se incluyen gastos en *hardware*, pues la compañía cuenta con el equipo adecuado para el uso correcto del sistema.

Con base en el precio inicial del sistema, que ronda entre los \$6657.00 USD y los \$7,387.55 USD, se obtiene un alto grado de viabilidad económica, ya que no se superan cifras abrumadoras para la empresa. Por ende, se puede realizar dicho proyecto, puesto que la compañía cuenta con el capital suficiente para la elaboración y el mantenimiento del sistema. Al tratarse de un sistema web desarrollado como proyecto de graduación, la empresa no debe incurrir en los gastos mencionados en la tabla.

Viabilidad legal

La solución considera las normativas que se plantearon en la Ley n.º 8148 del Código Penal Costarricense, que hacen referencia a los arts. 196 BIS (Violación de

comunicaciones electrónicas), 217 BIS (Fraude informático) y 229 BIS (Alteración de datos y sabotaje informático). Lo anterior tiene el fin de realizar un sistema web que se encuentre regulado por dichos artículos para una mayor transparencia y seguridad cibernética. Del mismo modo, se incorpora una ley fundamental en todo sistema web, la cual es la Ley n.º 8968 sobre la protección de la persona frente al tratamiento de sus datos personales. Esto con el objetivo de proteger los datos personales de los trabajadores y los clientes que asisten a la empresa Ópticas Dr. Max.

El desarrollo de dicho sistema utiliza tecnologías de autenticación de usuarios, lo que brinda seguridad digital al sistema. Lo anterior tiene el fin de evitar accesos indebidos al sistema contable que poseerá datos sensibles como ganancias, pérdidas, datos de cuentas por pagar e información de los clientes. Cabe destacar que el sistema cumplirá con la Ley de Derechos de Autor n.º 6683 de la Asamblea Legislativa de la República de Costa Rica de 1982, ya que se respetarán las licencias de *software* que se utilizan para la elaboración del sistema.

Del mismo modo, se cumple con la Ley n.º 4573 para reprimir y sancionar los delitos informáticos de la Asamblea Legislativa de la República de Costa Rica del año 2001. Esto se debe a que la solución incorpora métodos que harán posible cumplir con la ley que se mencionó. Estos métodos implementan protocolos de seguridad que incluyen una bitácora de registro de actividad, lo que reduce en gran medida el riesgo a la seguridad y a la mala operación del sistema contable.

Proyecciones

Alcance funcional

Se plantearon unos módulos que pueden asegurar el funcionamiento correcto de las gestiones contables de la empresa. Estos módulos se diseñaron para eliminar el mal manejo contable de ciertos procesos que la compañía realizaba de forma manual.

Tabla 3

Módulos del sistema

Nombre del módulo	Descripción del módulo
Gestionar los montos de la caja chica	Este módulo se encarga de asignar los

	montos de la caja chica y calculará cuando un monto sobrepase o sea menor de lo asignado, además de verificar que cada tipo de gasto sea definido previamente por la persona usuaria.
Gestionar cierre de caja	Este módulo se encarga de realizar la contabilización total de los ingresos por cada método de pago que posee la empresa.
Calcular los gastos	Se encarga de contabilizar los gastos que se generan en el mes, que se relacionan con los servicios públicos, licencias, mantenimiento y otros.
Gestionar facturación	Tiene como objetivo optimizar el proceso de facturación, lo que permite realizar facturas de forma electrónica vinculadas a los clientes y con su respectivo monto.
Gestión de cuentas por pagar	Lleva el control de los montos por pagar de la empresa, lo que le permite al administrador gestionar los saldos pendientes, se puede cancelar o abonar a los saldos pendientes a proveedores, servicios, entre otros.
Gestión del presupuesto	El sistema se encarga de hacer una proyección del presupuesto con base en los ingresos y gastos de la empresa dentro de un margen de tiempo

	establecido por el administrador del sistema
Gestión de ingresos	Se encarga de contabilizar los ingresos que se generan en el mes, tomando en cuenta las ventas por tipo de pago y productos vendidos.
Gestionar cuentas contables	Este módulo se encarga de la organización de las cuentas contables como los ingresos, los gastos, las cuentas por pagar y otras. A la vez, realiza un catálogo, el cual tiene el código, el nombre de la cuenta, el tipo de la cuenta y el balance de las cuentas.
Gestión de cuentas por cobrar	Se encarga de la gestión de las cuentas por cobrar de la empresa, le permite al administrador gestionar los saldos pendientes de los clientes, se puede abonar o cancelar a las cuentas de servicios prestados, entre otras.
Manteamientos	Este módulo se encarga de realizar el borrado, inserción, modificación, actualización de datos
Consultas	Este módulo se encarga de generar información proporcionada de las diferentes tablas
Reportes	Este módulo se encarga de generar información proporcionada de las diferentes tablas y procesos, pero con un formato específico, según lo solicite la persona usuaria. Podrá ser impreso o por pantalla

Seguridad	Este módulo se encarga de asignar los montos de la caja chica y calculara cuando un monto sobrepase o sea menor de lo asignado, además de verificar que cada tipo de gasto sea definido previamente por la persona usuaria.
-----------	---

Fuente: Elaboración propia

Alcance metodológico

El sistema que se planteó para el proyecto tiene como base el desarrollo web de una aplicación contable que logre automatizar los procesos de la empresa Ópticas Dr. Max. Se pensaron diversas metodologías para su debida implementación, no obstante, se decidió escoger la metodología de cascada (*waterfall*) debido a que esta se conforma en capas y cada capa se realiza cuando la anterior se termina, lo que brinda orden en el desarrollo de la solución. La metodología en cascada se puede explicar según CMC Global (2025) como:

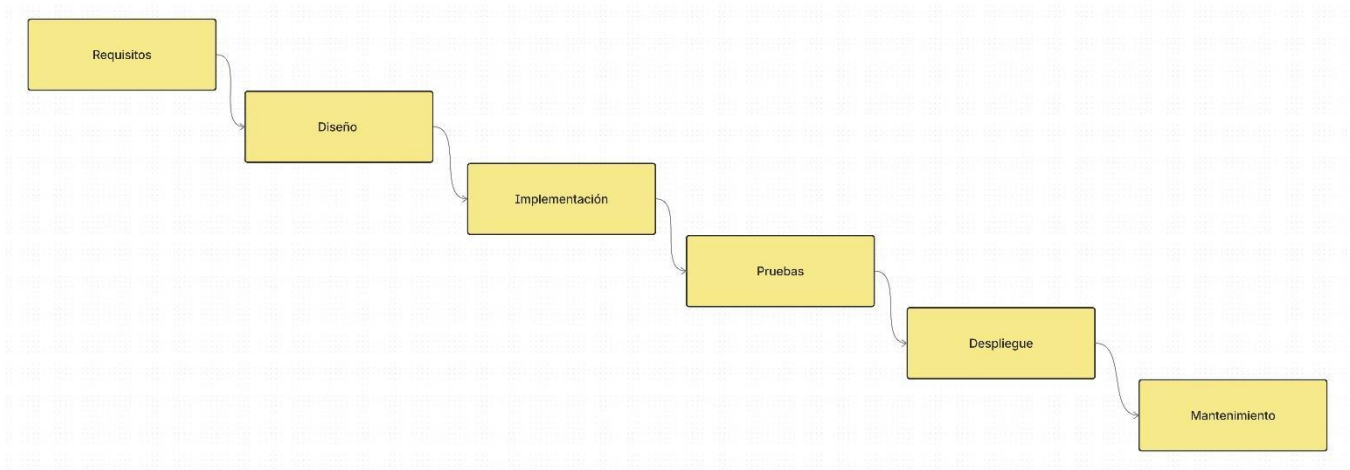
El modelo en cascada es un enfoque del ciclo de vida del desarrollo de software (SDLC) que sigue una sucesión lógica de fases para un proyecto, similar a cómo el agua se precipita al borde de un precipicio. Durante cada etapa del desarrollo, se establecen objetivos o puntos finales únicos para el proyecto. Una vez alcanzados ciertos objetivos o puntos finales, no es posible retomarlos (párr. 2).

Esta metodología permite mantener el orden del desarrollo de la propuesta y brinda confianza a la empresa, ya que se ofrece una solución a los problemas mencionados de manera ordenada y segura.

La metodología en cascada cuenta con seis etapas ya establecidas, lo cual proporciona una base para su uso. Las fases se dividen en el siguiente orden: requisitos, diseño, implementación, pruebas, despliegue y mantenimiento.

Figura 1

Etapas de la metodología en cascada



Fuente: Elaboración propia

En la Figura 1 se muestran las seis etapas del desarrollo de un sistema utilizando la metodología en cascada. El ciclo de vida de esta metodología, según Jain (2024) es:

El ciclo de vida en cascada, o modelo en cascada, es un enfoque tradicional para el desarrollo de software, donde cada fase del proyecto fluye secuencialmente, como una cascada. Esto significa que se completa un paso a la vez, como la planificación o la codificación, antes de pasar al siguiente. Es estructurado y fácil de entender, lo que lo hace ideal para proyectos con requisitos claros y definidos (párr. 1).

En la cita se describe el ciclo de vida del método en cascada y su funcionamiento, el cual se basa en la continuación de cada etapa al finalizar la anterior.

Alcance tecnológico

El alcance tecnológico requerido al plantear la propuesta de graduación es de alto grado, ya que la arquitectura que debe tener el sistema en temas de *software* suele ser la decisión más relevante al elaborar el desarrollo web. Esto se debe a que es necesario cumplir los requerimientos que se plantearon con base en los problemas que presentó la

empresa que se mencionó.

Se usa para el desarrollo web un modelo que permita al operador del sistema ejecutarlo desde cualquier navegador, lo que evita la descarga e instalación de aplicaciones externas. Esto tiene el fin de generar una operación del sistema más sencilla y rápida.

El motor que se utiliza para temas de almacenamiento y base de datos es SQL Server, el cual permite tener de manera organizada los datos contables de la empresa. No obstante, al contar con un servicio en la nube, también se asegura que el *software* sea escalable y funcional a largo plazo, al disponer de servicios de mantenimiento más rápidos.

El desarrollo del sistema en el ámbito de *back-end* y *front-end* se logra con el *software* de Visual Studio. Se usa Entity Framework y Razor para desarrollar el código necesario para el debido funcionamiento del sistema. Las herramientas de *software* mencionadas se basan en los lenguajes de programación JavaScript y C# (C-Sharp), los cuales otorgan un buen grado de compatibilidad con diversos dispositivos, lo que proporciona un alto grado de escalabilidad a largo plazo.

Capítulo II: Marco referencial

Relevancia en la automatización de los sistemas contables

Para hablar de la relevancia de la automatización de un sistema contable, primero se debe mencionar qué es un sistema contable y para qué sirven estos, así como sus beneficios en el ámbito empresarial. La automatización de un proceso es un paso fundamental que toda empresa debe contemplar, más cuando se menciona la contabilidad, ya que si esta no se encuentra presente, se pueden perder datos sensibles que comprometen a la institución, generando molestias en clientes, lo que se traduce en pérdidas monetarias. Según Conzultek (2022):

El software de contabilidad reúne bajo un mismo techo todos los sistemas y aplicaciones dedicados a la gestión y procesamiento de datos financieros. Los profesionales de contabilidad y los equipos de teneduría de libros utilizan estos programas para gobernar las cuentas y automatizar las operaciones sistemáticas, mientras que ciertos sistemas también están habilitados para registrar datos contables, medir indicadores e informar sobre la actividad financiera de la compañía (párr. 1).

Al tener conocimiento de que se trata de un sistema contable, se pueden mencionar sus beneficios en las empresas. Algunos de los beneficios más relevantes son el manejo exacto de los ingresos y pérdidas de la empresa, lo que se logra gracias a la automatización mediante códigos programables. Además, se encuentra como un gran beneficio la mejora en la eficiencia del personal, debido a que los trabajos que deben realizar los contadores se facilitan y agilizan al no ser manuales y al no contener el factor del error humano de manera tan frecuente.

La relevancia de automatizar un sistema contable en las empresas es de alto grado, lo que, sumado a los beneficios que este proceso puede traer en el ámbito empresarial, lleva a la conclusión de que un sistema contable es altamente rentable y necesario para toda empresa que pretenda mantener de manera adecuada sus finanzas.

Características de un sistema contable

Al realizar un sistema contable, se deben contemplar los componentes necesarios para un manejo óptimo enfocado en la empresa a la que se le brinda esta herramienta tecnológica. No obstante, cada sistema enfocado en la contabilidad empresarial debe tener características que ayuden al usuario final con el uso del sistema. Además, se buscan particularidades que mejoren la contabilidad mediante procesos que no se pueden llevar a cabo de forma manual. La simplicidad es una característica que se debe buscar al realizar un sistema contable, según QuickBooks (2024):

Algunos elementos como la interfaz de usuario, los tableros de control y los módulos transaccionales de un sistema de contabilidad tienen que ser amigables e intuitivos para las personas que van a operarlos. Así, cumplirán su objetivo de hacer procesos fáciles y sencillos (párr. 26).

Una característica que se busca al implementar un sistema contable y que es importante para este es la contabilización en tiempo real. La explicación que brinda QuickBooks (2024) sobre la contabilización en tiempo real es la siguiente: “El cálculo de impuestos es de las funciones obligatorias para los softwares contables. Este proceso debe ser inmediato y, sobre todo, en tiempo real, porque de eso dependerá sellar, emitir y obtener cada factura solicitada” (párr. 30). Lo anterior deja clara la importancia de conseguir un sistema contable que cuente con una contabilización que funcione, de manera que, al realizar una operación, se vea reflejada en otro movimiento en tiempo real

Otra característica fundamental que debe contener un sistema de contabilidad es la generación de reportes. Esta característica brinda a la empresa reportes de manera continua, lo que ayuda a conocer de forma exacta las finanzas. Con base en estos reportes se pueden tomar decisiones en el futuro, las cuales ayudan a la compañía a mejorar constantemente.

Conceptos de un sistema contable

El concepto principal de un sistema contable en el ámbito empresarial es la *contabilidad*. Este proceso se define por Fernando (2025), de la siguiente manera:

La contabilidad monitorea las finanzas de una empresa, incluyendo los ingresos, los egresos y su balance. Ayuda a propietarios e inversores a monitorear el rendimiento de la empresa a lo largo del tiempo, garantizando que los informes financieros cumplan con las normas legales y regulatorias (párr. 1).

Al conocer el concepto fundamental que rodea este sistema, se pueden mencionar algunos otros conceptos que realizan tareas, las cuales, en conjunto, conforman un sistema automatizado contable.

Al conocer lo que la contabilidad en un ámbito financiero implica, se puede mencionar la combinación de la contabilidad y el sistema que, en palabras de CONTPAQi (2025) es:

Un sistema de contabilidad es una herramienta digital que permite registrar, organizar y analizar todas las operaciones financieras de una empresa. Estos sistemas automatizan tareas como la emisión de facturas, la generación de pólizas contables, el cálculo de impuestos y la elaboración de reportes financieros (párr. 1).

Tabla 4

Conceptos de la contabilidad

Nombre	Descripción
Ingresos	El ingreso en contabilidad es el valor económico que una empresa genera por vender bienes o prestar servicios en un período, sin importar si ya lo cobró o no
Gastos	Según Mamian (2025): “Un gasto es una salida de dinero que una empresa o persona paga por la compra de un producto, un bien o un servicio. Por lo general, se convierte en una inversión ya sea tangible o intangible” (s. p.).

Cuentas por pagar	Las cuentas por pagar son las deudas u obligaciones que tiene un negocio a los proveedores o a los acreedores. Por lo general, el pago de estas obligaciones no es inmediato, esto se debe a que tienen un plazo acordado.
Facturas	Las facturas contables son documentos esenciales en el mundo de los negocios, actuando como registros formales de las transacciones comerciales. Desde una pequeña tienda hasta una gran corporación, la correcta emisión y gestión de las facturas es crucial para mantener la salud financiera de una empresa.
Cuentas contables	Las cuentas contables son registros estructurados que reflejan todas las transacciones económicas de una empresa. Por lo tanto, se organizan de manera cronológica y categorizada, lo que permite un mejor control financiero. Además, un registro de cuentas contables adecuado es crucial para que una compañía pueda tener un panorama claro de sus activos, pasivos y patrimonio.
Presupuesto	Según Perucontable (2022) un presupuesto contable es el documento que refleja el plan operativo de una empresa para un período determinado, usualmente 1 año. En palabras sencillas, es una previsión de los resultados, contando con los distintos flujos de dinero que se moverán en ese plazo acotado. O sea, un presupuesto en contabilidades el cálculo aproximado de los ingresos y gastos que se producirán por una actividad económica concreta.
Cierre de caja	El cierre de caja es el método que se utiliza para contabilizar, tanto la salida como la entrada de dinero de un negocio o empresa. La función principal de esta acción es comprobar que la cantidad de dinero en efectivo o las ganancias obtenidas correspondan al valor de las ventas que se han reportado en la caja.
Caja chica	La caja chica es una pequeña cantidad de dinero que una empresa reserva para gastos menores e imprevistos. Estos pagos menores incluyen suministros de oficina, papelería, comidas, almuerzos para

	clientes, sellos, etc. La cantidad de fondos de caja chica puede variar según el tamaño de la organización.
--	---

Fuente: Elaboración propia

Los conceptos que rodean este proyecto de graduación están definidos en la anterior Tabla 3. Estos términos son fundamentales para conocer el debido funcionamiento de los módulos que se plantearon, ya que sin ellos no se logra la debida implantación programable del sistema contable.

Aspectos legales de un sistema contable en Costa Rica

Costa Rica, al igual que muchos países, cuenta con un marco legal que funciona como base para desarrollar un sistema contable. Este marco legal que se debe acatar, nuestro país lo adoptó conforme a la normativa internacional en relación con la contabilidad. Cabe destacar que la normativa contable en Costa Rica está regulada por el Colegio de Abogados y el Consejo Nacional de Supervisión del Sistema Financiero. A la vez, es necesario tomar en cuenta el art. 33-38 del Código de Comercio (Ley n.º 3284, 1964). Estos artículos se basan en el marco legal costarricense para las empresas del sector privado. El art. 33-38 del Código de Comercio establece que se debe llevar una correcta contabilidad mediante diversos métodos autorizados, entre los cuales se encuentra el sistema contable. Este sistema debe poder usarse, con el fin de presentar los estados financieros correctamente realizados y ordenados.

Las Normas Internacionales de Información Financiera (NIIF) son un conjunto de estándares técnicos y principios contables definidos por la International Accounting Standards Board (IASB) para armonizar la normativa contable en el mundo. Según Borges (2023): “Estas reglas deben aplicarse de manera uniforme en las empresas de todos los países para que sus estados financieros sean comparables entre sí y resulten más comprensibles para todos los encargados de tomar decisiones” (párr. 3). La NIIF 9 introdujo un apartado sobre la automatización del control de cuentas por cobrar, integrando análisis predictivos dentro de

la planificación de recursos empresariales, o por sus siglas en inglés, *ERP*. Esto último beneficia en gran medida el desarrollo web del sistema contable.

Las normas internacionales de contabilidad (NIC) se implementan como regímenes en las empresas, debido a que establecen cómo se deben presentar: “Los estados financieros, desde lo que debe incluirse hasta el formato adecuado, todo sujeto a las normas internacionales de contabilidad emitidas por el International Accounting Standards Committee (IASC)” (QuickBooks en español, 2023, s. p.). Esta normativa se debe tener en cuenta al crear un sistema contable, ya que le brinda a la empresa un sistema que esté respaldado de manera correcta según las normativas emitidas por el IASC.

Las leyes que abarcan el marco legal para el desarrollo de un sistema web basado en la contabilidad de una empresa del sector privado costarricense se fundamentan en las Normas Internacionales de Información Financiera (NIIF/NIC). Algunas de las leyes que conforman esta normativa son: Ley n.º 1269 (Ley Orgánica del Colegio de Contadores Privados de Costa Rica), Ley n.º 9635 (Fortalecimiento de las Finanzas Públicas) y las regulaciones de la Sugef. La Sugef (s. f.):

Es una entidad supervisora y fiscalizadora que vela por la estabilidad, solidez y eficiente funcionamiento del sistema financiero nacional. La Ley Orgánica del Banco Central de Costa Rica (n.º, 7558), vigente desde el 27 de noviembre de 1995, declara de interés público la fiscalización de las entidades financieras y crea la Superintendencia General de Entidades Financieras (SUGEF), bajo la figura jurídica de la desconcentración máxima. Asimismo, la función de la SUGEF, según se define en la Ley Orgánica del Banco Central de Costa Rica (Ley n.º, 7558), es velar por la estabilidad del Sistema Financiero Nacional como un todo y ésta tiene competencias primordialmente en materia financiero-contable, no así en asuntos relativos a prestación de servicios. Asimismo, la SUGEF no es una instancia judicial y como ente supervisor carece de potestad para coadministrar los entes que supervisa (s. p.).

Al conocer algunas de las leyes que establece el gobierno de Costa Rica, se puede llegar a la conclusión de que se cuenta con material suficiente para el correcto manejo legal en materia contable de empresas privadas. Esto beneficia de gran manera al desarrollo web

de una aplicación que gestione las finanzas de una institución o empresa, lo que a la vez brinda seguridad legal y puede generar satisfacción al cumplir con el marco legal estipulado por el gobierno de nuestro país.

Metodología cascada (waterfall)

En palabras de Taylor (2024):

El modelo de cascada es un enfoque lineal y secuencial para el desarrollo de software, donde cada fase debe completarse antes de que comience la siguiente. Este modelo es una de las metodologías más antiguas y tradicionales en ingeniería de software. Se suele representar como un flujo descendente similar a una cascada, donde el proyecto pasa por fases de forma estructurada y paso a paso (párr. 1).

Con base en lo mencionado por Taylor (2024), se puede concluir que la metodología en cascada brinda orden en el desarrollo de las etapas que están involucradas en proyectos de desarrollo de *software*, pues cuenta con una estructura ya establecida que se basa en ir terminando cada etapa en orden, de manera que si una etapa aún se encuentra en desarrollo, no se puede continuar con la siguiente hasta la finalización de la actual.

Software en un sistema contable

El *software* es un punto fundamental en cada sistema web que se desee realizar, ya que sin este no existiría tal sistema. El *software* de un sistema cuenta con diversos componentes, como el motor para la programación del *front-end* y *back-end*, el motor de la base de datos que contiene la información de la empresa a la cual se le desea llevar a cabo el sistema, el lenguaje por utilizar para la elaboración del código y las diversas herramientas empleadas al elaborar el código del sistema.

El software, según Inesdi (2025) es:

Un software podría definirse como un conjunto de instrucciones que hacen posible la comunicación con un ordenador. En otras palabras, son los programas informáticos que permiten que estos últimos, así como otros dispositivos, tengan

una utilidad. Mientras que el hardware es la parte física, el software es intangible y esencial para el funcionamiento de cualquier sistema informático. Sin embargo, no hay una única modalidad que sirva para todo, sino que existen distintos tipos de software con categorías que varían según su función y propósito. Por ejemplo, el software de sistema, el de aplicación y el de programación (s. p.).

Al conocer la importancia y qué es el *software*, se puede concluir que sin este no existirían los sistemas que ayudan en la vida cotidiana de cada persona. Enfocándose en un sistema contable, se debe tomar en cuenta qué tipo de *software* se utiliza, ya que al tratarse de un punto crítico como la economía de la empresa, es necesario analizar y escoger cuál beneficia más el desarrollo y la escalabilidad del programa. Sin un estudio o conocimiento previo del *software* por elegir, se puede dificultar la elaboración del sistema, así como la eficiencia de este mismo.

Visual Studio como motor de desarrollo

Visual Studio se puede definir, según Arimetrics (s. f.), como:

Visual Studio es un entorno de desarrollo integrado (IDE) desarrollado por Microsoft que permite a los programadores crear aplicaciones para distintas plataformas (incluyendo Windows, Android, iOS y Linux). Este software ofrece diversas herramientas para facilitar el proceso de desarrollo, como depuración de código, integración con sistemas de control de versiones, pruebas automatizadas, y asistencia en la escritura de código (párr. 1).

Al saber que es Visual Studio, se puede mencionar que este brinda altas capacidades en lo que a temas de programación se refiere, ya que ofrece una interfaz intuitiva. Esto junto con el modelo que se piensa utilizar para la creación del sistema contable, permite obtener un motor robusto con capacidades de alta escalabilidad del código en el futuro. Lo anterior se debe a que Visual Studio es uno de los motores que más se utilizan en el ámbito mundial en lo que respecta a la programación en entornos web, lo que ayuda a futuros programadores a dar mantenimiento constante al sistema que se desee implementar en una empresa.

El motor de desarrollo mencionado cuenta con algunas características que lo hacen ideal para el desarrollo de sistemas web. Ciertas de estas particularidades son: cuenta con un entorno integrado que permite a los programadores escribir, depurar y probar el código en un mismo lugar, lo que facilita el proceso de desarrollo y da la posibilidad de ahorrar tiempo. Además, cuenta con la característica de la integración de herramientas, que les permite a los desarrolladores utilizar herramientas adicionales para complementar el proceso de desarrollo. Otra ventaja que posee este motor es la integración de diversos lenguajes de programación, lo que lo vuelve flexible y adaptable a diferentes necesidades. A la vez, cuenta con la función de depuración de código, lo que ayuda a los desarrolladores a detectar y solucionar errores en el código de manera rápida y eficiente (Arimetrics, s. f.).

Al conocer las características, los beneficios y qué es Visual Studio, se puede llegar a la conclusión de que este motor es bastante completo y recomendable en el momento de elaborar un sistema contable web. Según Arimetrics (s. f.), se concluye que:

Visual Studio no solo es una herramienta poderosa por sus características técnicas, sino que también se beneficia de una amplia comunidad de usuarios y un robusto sistema de soporte. Esta comunidad y soporte son vitales para los desarrolladores que buscan aprender, resolver problemas y mejorar sus habilidades (s. p.).

Componentes por utilizar en Visual Studio

Las herramientas que ofrece el motor de desarrollo Visual Studio son competentes e ideales cuando se refiere al desarrollo web. El motor de desarrollo mencionado cuenta con una amplia variedad de componentes que hacen el desarrollo más sencillo y práctico. Entre estos elementos se encuentran los siguientes: lenguaje de programación (C-Sharp), modelo de desarrollo web (MVC), componentes del *front-end* (Razor, CSS, HTML) y el uso del *framework* llamado Entity Framework. Los elementos y herramientas mencionados conforman un desarrollo sostenible y organizado, lo cual, anexado al motor de desarrollo, conforma una base sólida y confiable que ayuda al desarrollo web.

Lenguaje de programación C-Sharp

Sharma (2026) menciona que:

C# (C Sharp) es un lenguaje de programación robusto y versátil que se utiliza para crear una amplia gama de aplicaciones que funcionan en múltiples plataformas. Desde aplicaciones web y de escritorio hasta servicios en la nube y juegos, es un lenguaje de confianza para desarrolladores de todo el mundo. Se sitúa constantemente entre los cinco mejores lenguajes de programación del Índice TIOBE , lo que destaca su gran demanda y relevancia en el sector. Con una sintaxis clara y potentes funciones integradas, C Sharp es fácil de aprender, lo que lo convierte en una excelente opción para principiantes (párr. 1).

Con base en lo mencionado sobre el lenguaje C# por Sharma (2026), se puede afirmar que este lenguaje cuenta con las capacidades adecuadas cuando se plantea realizar un sistema web enfocado en la contabilidad.

El lenguaje mencionado cuenta con una sintaxis clara que lo convierte en un lenguaje fácil de entender. Esto beneficia tanto al programador como al programa, ya que, debido a esta situación, se puede brindar mantenimiento y escalabilidad al código de forma sencilla, lo que resuelve en gran medida los problemas a largo plazo. Un ejemplo de esta sintaxis se muestra en la siguiente imagen. La base de mostrar esta figura es ejemplificar la sintaxis que se debe usar al programar con este lenguaje. En la figura se presenta un mensaje de *Hello World* o por su traducción al español, *Hola mundo*.

Figura 2

Sintaxis C-Sharp

```
using System;

class Program
{
    static void Main()
    {
        Console.WriteLine("Hello, World!");
    }
}
```

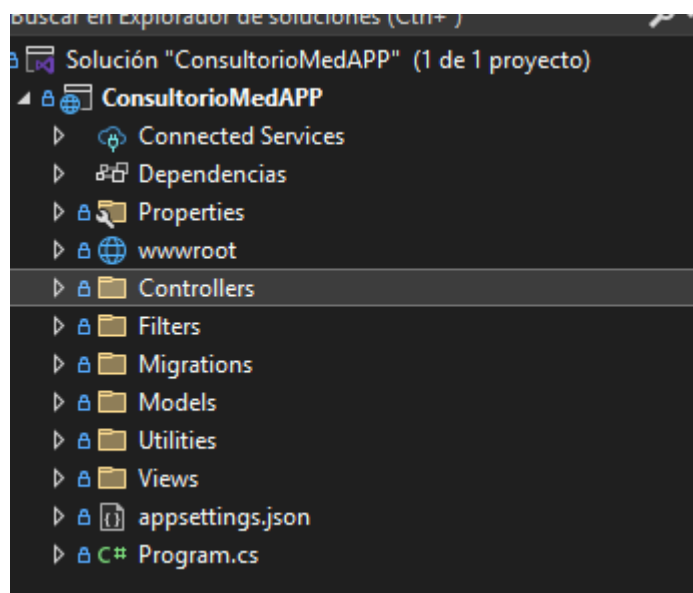
Fuente: Sharma (2026).

Modelo MVC (modelo-vista-controlador)

El modelo MVC ayuda al desarrollo al generar orden en el momento de construir las bases del sistema web. Esto se debe a que cuenta con una estructura clara que se enfoca en tres etapas del código por desarrollar para un sistema web. Estas fases son las mencionadas a continuación: modelo, vistas y controladores. La fase llamada modelo trata del esquema que se relaciona con la base de datos; en esta fase solo es necesario realizar estrictamente lo que involucra a la base de datos. La fase de vista se enfoca en el desarrollo del *front-end* del sistema, ayudada de componentes como Razor, CSS y HTML. Por último, cabe destacar la fase de controlador, ya que en esta se debe colocar el código que contiene las funciones que harán del sistema una aplicación funcional con sus respectivos módulos y funcionalidades. En la figura denominada como MVC se muestra un proyecto programable que utiliza el modelo mencionado para estructurar las diversas fases del desarrollo de este mismo.

Figura 3

MVC



Fuente: Elaboración propia

Componentes del front-end

Al realizar un desarrollo web, se debe plantear cuáles componentes pueden ser los óptimos al desarrollar dicho sistema. Una parte fundamental de este desarrollo es el *front-end*, el cual, según Ionos (2024) es:

La parte de una página web o aplicación que los usuarios ven y con la que interactúan. Se centra en presentar la información de manera atractiva y facilitar las interacciones del usuario. El diseño del frontend se adapta a las necesidades y expectativas de los usuarios, priorizando una experiencia de usuario positiva (párr. 2).

Al conocer qué es el *front-end* y para qué sirve en el desarrollo de sistemas web, se tiene la posibilidad de mencionar algunos elementos que ayudan a realizar el código al elaborar una página web.

Razor es una sintaxis basada en .NET que permite agregar código del lado del servidor directamente en archivos HTML, lo que facilita la creación de páginas web. Cabe mencionar la importancia de HTML en el uso de Razor: en HTML se crean las etiquetas, las cuales sirven para dar la estructura y el contenido de las páginas web. Por último, se encuentra el uso de CSS, que se encarga del diseño de botones, etiquetas de texto, menús, submenús, logos, colores, tamaño de las letras, fuentes de las letras y el espaciado entre los diversos componentes que se deben incluir en el *front-end*.

Entity Framework

De acuerdo con NetMentor (s. f.):

Entity framework Core es una librería creada por Microsoft que permite acceder a las bases de datos, principalmente relacionales, de una forma muy simple. La gran ventaja de Entity Framework Core es que es muy simple y amigable, ya que permite utilizar código en vez de SQL directamente (s. p.).

Al conocer qué es Entity Framework, se puede concluir que es una herramienta útil al desarrollar un sistema web, pues facilita la conexión de una base de datos con el desarrollo de la página web. Esto sumado a las herramientas y componentes que se mencionan en este proyecto de graduación, da la posibilidad de obtener un desarrollo bien estructurado en el futuro en temas de mantenimiento y escalabilidad.

Base de datos relacional y su utilidad en un sistema contable

La información que brinda Google Cloud (2026) sobre una base de datos relacional es la siguiente:

Una base de datos relacional es una colección de información que organiza datos en relaciones predefinidas, en la que los datos se almacenan en una o más tablas (o relaciones) de columnas y filas, lo que facilita la visualización y la comprensión de cómo se relacionan las diferentes estructuras de datos entre sí. Las relaciones son conexiones lógicas entre las diferentes tablas y se establecen a partir de la interacción entre ellas (párr. 1).

Con base en lo mencionado por Google Cloud (2026), se puede señalar que una base de datos bien elaborada es de gran beneficio al desarrollar un sistema web. Sin embargo, si la base de datos se estructura mal, puede llevar al colapso de las funcionalidades que debe tener el sistema web que se desarrolla. Al crear una base de datos, se deben seguir las buenas prácticas de programación y mantener un orden en cada proceso que implique este desarrollo. De igual forma, al realizar el estudio de una base de datos, es necesario tener en cuenta el uso de un motor que sea lo suficientemente potente, ya que se deben elaborar diversas tablas, procedimientos almacenados y funcionalidades que ayuden con el *back-end* y el *front-end* en el desarrollo web.

La utilidad de una base de datos relacional aplicada a un sistema contable es importante, ya que funciona como base para los diferentes métodos que emplean los sistemas que se relacionan con la contabilidad empresarial. Algunos procesos que puede crear una base de datos en relación con un sistema contable son: manejo de facturas, creación de funciones de suma y resta para el cálculo de ganancias y pérdidas, elaboración de libros

contables para su inclusión adecuada en un sistema, gestión de usuarios, controles de auditoría y gestión de porcentajes mediante operaciones complejas.

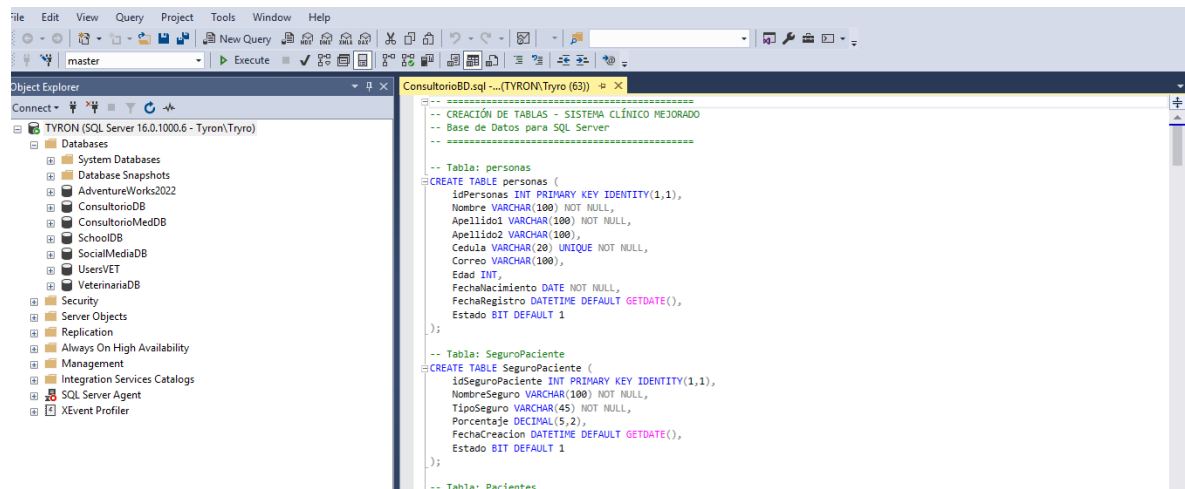
SQL Management Studio

SQL Server Management Studio (SSMS) es un motor de desarrollo: “Que proporciona un entorno integrado para administrar cualquier infraestructura relacionada con SQL Server. Con SSMS, los programadores pueden configurar, gestionar y administrar las bases de datos relacionales” (Revolution Soft, s. f., s. p.). En palabras de Blanco (2023): “SQL Server Management Studio (SSMS) ha sido desde hace mucho tiempo una herramienta esencial en el arsenal de todo administrador de bases de datos, desarrollador y profesional IT que trabaja con Microsoft SQL Server” (párr. 3).

SQL Server, al ser un motor de desarrollo enfocado en bases de datos, cuenta con algunas ventajas que sobresalen y ejemplifican por qué es uno de los motores que más se utilizan en entornos de desarrollo web. Ciertas ventajas que este posee pueden variar entre la interfaz intuitiva, que ayuda a los programadores familiarizados con otros motores de desarrollo y las herramientas que van desde la depuración de código hasta la administración de datos. Una de sus grandes ventajas se encuentra en la compatibilidad que este motor tiene, ya que se puede usar una base de datos creada en versiones anteriores y que resulta perfectamente funcional en versiones más modernas.

Lo anterior resalta un motor de desarrollo versátil, potente y funcional para la creación de una base de datos, la cual es una de las fases más relevantes en el desarrollo de un sistema web. Aunado a lo anterior, Blanco (2023) indica que: “Esta herramienta es una elección insuperable para profesionales que buscan administrar y optimizar sus bases de datos de manera eficiente” (párr., 18).

En la Figura 4 se presenta cómo se observa el motor de desarrollo SQL Server Management Studio y sus respectivos botones. Además, se logran apreciar tablas que contienen datos sobre una base de datos llamada consultorioDB.

Figura 4*SQL Server Management Studio*

Fuente: Elaboración propia

Capítulo III: Marco metodológico

Enfoques de investigación

Un enfoque de investigación claro y bien desarrollado debe contener cómo se debe tratar un problema por solucionar. Para realizar el enfoque investigativo, se toma en cuenta el estudio previo de la metodología que se desea escoger, ya que cada una se enfoca en diferentes aspectos que quizás no se adapten al tipo de proyecto o solución que se desee brindar a una institución o empresa. Algunos de estos enfoques pueden abarcar el enfoque cuantitativo, cualitativo y el mixto que, como la palabra lo menciona, es una combinación del enfoque cualitativo y cuantitativo. Lo anterior sobre las diversas metodologías por elegir busca ofrecer una alternativa para que cada solución contenga maneras de obtener información que pueda beneficiar a un proyecto, con el fin de obtener el mejor resultado posible.

Enfoque cuantitativo

Según Concepto.de (2021a):

El método cuantitativo es un conjunto de estrategias científicas que se usan en investigación para obtener información expresada en datos numéricos. De esta forma, se puede analizar un tema o un objeto de estudio teniendo en cuenta sus características medibles, es decir, aquellas que se pueden expresar mediante números (párr. 2).

Con base en lo mencionado por Concepto.de (2021), se puede afirmar que este enfoque se basa en el estudio de la información obtenida sobre un problema, el cual se transforma en datos numéricos, lo que suele dar un panorama más estadístico sobre el problema que se desea plantear.

A partir de lo anterior, se puede usar este enfoque para el desarrollo del sistema que se planteó, ya que se puede medir con datos numéricos la cantidad de veces que se cometen errores al calcular los beneficios o pérdidas de la empresa. De acuerdo con estos resultados, se lograría llegar a una solución con base en los datos más elevados y brindar prioridad a estos, lo que facilita la investigación del proyecto al ofrecer posibles soluciones para las estadísticas con los rangos más altos del proyecto investigativo realizado con el enfoque cuantitativo.

Cabe mencionar que este enfoque se rige por la veracidad de los instrumentos que se utilizan al medir los datos que se recolectan en la fase investigativa del proyecto. Si estos instrumentos no se emplean de manera correcta o no cuentan con la veracidad suficiente, este enfoque no contaría con el respaldo y la certeza de un método de investigación que fue usado de forma correcta. Lo anterior perjudica el estudio, por lo que se busca de forma constante obtener datos que se puedan convertir en soluciones correctas a un problema.

Enfoque cualitativo

Según Hecker y Kalpokas (2025):

La investigación cualitativa es un enfoque esencial en diversas disciplinas académicas y campos profesionales, ya que trata de comprender e interpretar los significados, las experiencias y las realidades sociales de las personas en sus entornos naturales. Este tipo de investigación emplea una serie de métodos cualitativos para recopilar y analizar datos no numéricos, como palabras, imágenes y comportamientos, y pretende generar percepciones profundas y contextualizadas de los fenómenos objeto de estudio (párr. 1).

Con base en lo mencionado por Hecker y Kalpokas (2025), se puede afirmar, a modo de resumen, que este enfoque consiste en recolectar datos de forma no numérica y obtenerlos

mediante diferentes métodos, lo que representa una clara diferencia con el enfoque cuantitativo, que se basa más en datos estadísticos.

Al conocer qué es el enfoque cualitativo, se puede mencionar para qué se diseñó. Este enfoque se basa en abordar cuestiones de investigación que se centran en comprender el comportamiento, las experiencias y las interacciones humanas, en lugar de solo los datos cuantitativos que suelen tratar los métodos cuantitativos. Cabe destacar que Hecker y Kalpokas (2025) plantean el objetivo principal de la investigación cualitativa, el cual es: “Obtener datos de la perspectivas, emociones, creencias y motivaciones de las personas en relación con cuestiones, situaciones o fenómenos específicos” (párr. 2).

El enfoque cualitativo se puede aplicar al proyecto que se planteó, ya que este enfoque se puede catalogar como el más importante al centrarse en los datos que se obtienen mediante el estudio del comportamiento del personal o de los clientes. Al desarrollar un sistema web, por lo general se debe evaluar al personal, pues es este quien utiliza dicho sistema. Al conocer los problemas que tiene dicho personal, se puede llegar a un contexto que logre satisfacerlo, lo que resuelve en gran medida la implantación de la página web.

Enfoque mixto

El enfoque mixto combina dos enfoques diferentes: el cualitativo y el cuantitativo en un estudio o investigación. Este método busca: “Aprovechar las fortalezas de ambos enfoques para obtener una comprensión más completa y profunda de un problema de investigación” (Cueva Luza *et al.*, 2023 p. 77). Por otra parte, la investigación cualitativa se centra en comprender y explorar los significados, experiencias y perspectivas a través de técnicas como entrevistas, observación o análisis de contenido y la investigación cuantitativa se basa en recolectar y analizar datos numéricos para identificar patrones, establecer relaciones y realizar inferencias estadísticas.

Seguido de lo mencionado, se puede afirmar que el enfoque mixto combina estos dos enfoques en una sola investigación. Esto puede ayudar a recopilar y analizar datos cuantitativos y cualitativos de forma separada y luego juntarlos, lo que da un enfoque completo que mezcla lo mejor de ambos enfoques.

Todo lo mencionado sobre el enfoque mixto deja un método de investigación completo que, al combinar lo mejor de dos enfoques, permite obtener datos más precisos y completos para el desarrollo web del sistema contable. Esto mejora el desarrollo y la implantación de dicho sistema logrado, ya que se puede conocer la perspectiva del futuro operador y las inquietudes de los clientes.

Enfoque de investigación seleccionado

Al conocer los beneficios de cada enfoque y sus desventajas, se puede seleccionar un enfoque que beneficie el desarrollo de dicho sistema, ya que este debe ocupar un método que sea robusto y confiable, con el fin de obtener los mejores resultados en temas de información, lo que se transforma en un desarrollo web óptimo y estable a largo plazo.

El enfoque seleccionado para este proyecto de graduación es el cualitativo, el cual se basa en obtener información detallada de los problemas mediante diferentes prácticas. Al ser un enfoque más centrado en el comportamiento y la perspectiva del trabajador y el cliente, se puede llegar de manera fácil y completa al problema en cuestión y desarrollar una solución

que brinde comodidad y funcionalidad al personal, lo que a la vez genera satisfacción en el cliente.

Una de las razones que se tomó en cuenta al escoger este enfoque fue el método para obtener la información. Al enfocarse más en experiencias humanas, se puede llegar a una conclusión que ayude a todos los sectores que rodean un desarrollo web. Esto se debe al conocimiento obtenido por medio de entrevistas o diversos métodos de extracción de información que generan un alto grado de conocimiento de la situación actual de la empresa Ópticas DR. MAX.

Tipos de investigación

Al referirse a un tipo de investigación, se alude a la estrategia que se debe escoger, con el fin de obtener los datos que se utilizan en una propuesta o documento pensado para solucionar un problema; en este caso, orientado por resolver los problemas contables de la empresa Ópticas Dr. Max. Cuando se busca el tipo de investigación, es necesario realizar un estudio previo que valide esta selección, ya que suele ser de suma importancia al validar los datos que se obtienen.

Existen varios tipos de investigación, no obstante, en esta solución solo se mencionan tres, los cuales se consideran los más adecuados con base en el tipo de desarrollo web que se planteó y son: investigación descriptiva, exploratoria y explicativa. Enseguida se detalla cada tipo de investigación mencionado, con la finalidad de obtener un resultado que se adapte de mejor manera a la propuesta que se planteó, esto con el fin de garantizar el trabajo de recolección de datos, validándolos de forma correcta.

Investigación exploratoria

Según Appinio Research (2023):

La investigación exploratoria es un método de investigación utilizado en las primeras etapas de un proyecto para profundizar en un tema cuando se dispone de poco o ningún conocimiento o información. Es un enfoque dinámico y flexible

que busca obtener perspectivas, descubrir tendencias y generar hipótesis iniciales (párr. 2).

Con base en lo mencionado por Appinio Research (2023), se puede interpretar que la investigación exploratoria se utiliza para estudiar un problema que no está definido. Esta clase de investigación se lleva a cabo para comprender mejor dicho problema. Aunque esta técnica es muy flexible en comparación con otros tipos de métodos, en ocasiones, suele requerir que la persona sea paciente y receptiva.

Debido a lo mencionado, se puede afirmar que este método de investigación puede ser de utilidad al querer realizar la propuesta llamada sistema web para Óticas Dr. Max. Esto se basa en el fundamento principal en el que se enfoca dicho método, el cual consiste en descubrir información cuando se posee poco conocimiento de esta. Además, es de utilidad al tratarse de un método flexible.

Este método de investigación cuenta con varios puntos que definen su importancia. Uno de los más relevantes es la reducción de riesgos y la adaptabilidad. Estos puntos combinados hacen de este tipo de investigación una gran alternativa al querer iniciar la propuesta que se desarrolla en un informe o, como en este caso, en un proyecto de graduación.

Investigación descriptiva

Al referirse a la investigación descriptiva, Stewart (2025) menciona que esta es, en esencia: “Un enfoque sistemático utilizado por los investigadores para recopilar, analizar y presentar datos sobre fenómenos de la vida real con el fin de describirlos en su contexto natural” (párr. 4). Según lo mencionado por la autora, se puede señalar que el objetivo de este tipo de investigación es obtener una descripción detallada de una situación en concreto, lo que, anexo a la propuesta, funciona de manera que permite recolectar datos detallados de los puntos más relevantes de la empresa.

Seleccionar este método puede ser vital, ya que ayuda a conocer los procesos que ejecuta la empresa, con el fin de arreglar o mejorar dichos procedimientos. Obtener estos datos con base en esta metodología ayuda a fundamentar de manera correcta y completa los módulos o características que debe tener el desarrollo web, lo que beneficia en parte la investigación y las diferentes etapas que se deben cruzar al querer ejecutar el sistema de la mejor forma.

Investigación explicativa

Según Pérez (2023): “La investigación explicativa tiene por objetivo dar respuesta a por qué unos fenómenos particulares funcionan en el modo en que lo hacen, es decir, busca relaciones causa-efecto y su cuantificación” (párr. 2). Conforme a lo mencionado por Pérez (2023), se puede afirmar que la investigación explicativa es una técnica que se debe utilizar para obtener una comprensión más profunda del estudio que se desea realizar, así como las causas detrás de un problema que aún no se ha estudiado ampliamente.

Al utilizar esta metodología, se puede explicar el problema al plantear preguntas sobre el funcionamiento de las cosas. Al referirse a la puesta en marcha de la solución con este tipo de investigación, se profundizan los diversos errores con los que cuenta la empresa. Esto se logra al estudiar con detalle el actual funcionamiento de la contabilidad empresarial. Como beneficio extra, esta metodología brinda un apoyo al determinar los contratiempos que pueden surgir cuando se implemente dicha solución de desarrollo web. Esto se logra al haber validado datos siguiendo el fundamento de la investigación explicativa.

Tipo de investigación seleccionado

El tipo de investigación es fundamental al querer desarrollar un sistema web. Al realizar la explicación de los tres tipos mencionados, se puede obtener un fundamento sólido al querer escoger un tipo de investigación. Cada uno cuenta con su enfoque, que diferencia el rumbo por tomar cuando se plantean las preguntas iniciales y el desarrollo del proyecto.

El tipo de investigación seleccionado fue el explicativo, ya que al plantear el fenómeno o por qué suceden los problemas en la empresa, se puede llegar a una conclusión más elaborada, la cual se puede definir como un mejor planteamiento de los módulos, objetivos y alcance que puede tener el proyecto. La razón principal que motivó para seguir esta metodología de investigación se basa en el hecho de plantear una pregunta y explicar de las variables que pueden influir en dicha respuesta. Esto combinado con el enfoque cualitativo permite obtener datos de manera confiable y, a la vez, estos datos se pueden entrelazar entre sí, logrando resolver más de un problema.

Fuentes de información

Al realizar una investigación siempre se deben tener fuentes que sustenten el trabajo realizado. Esto se hace con el fin de garantizar un trabajo de excelente calidad, que es lo que se espera de la persona investigadora en cuestión. Los recursos que se utilizan brindan conocimientos que ayudan con el estudio y el entendimiento de un tema, lo que genera un amplio grado de exactitud al exponer un tema o un problema. Lo anterior se relaciona con el planteamiento de este trabajo final, que se enfoca en un sistema contable. Esto ya que se debe llevar a cabo una investigación previa que confirme los problemas y soluciones expuestas con anterioridad, lo cual se logra con la ayuda de las fuentes de información primarias, secundarias y terciarias.

Fuentes primarias

Según la University of Minnesota Crookston (2026), las fuentes primarias son:

Registros de eventos o evidencia tal como se describen inicialmente o sucedieron, sin interpretación ni comentario alguno. Se trata de información que se presenta por primera vez o de materiales originales en los que se basan otras investigaciones. Las fuentes primarias muestran ideas originales, informan sobre nuevos descubrimientos o comparten información novedosa (párr. 2).

Al guiarse por lo mencionado por la University of Minnesota Crookston (2026), se puede interpretar que las fuentes primarias tienen como objetivo brindar conocimiento de un tema de forma que esta información sea de primera mano. Con esto último se plantea que es información que se obtiene sin comentarios que cambien el tema o den un punto extra y que, por lo tanto, se obtiene de materiales originales. Lo anterior es importante, ya que brinda información precisa y actualizada a los tiempos modernos, lo que puede dar un mejor enfoque a la investigación del presente trabajo.

Fuentes secundarias

Las fuentes secundarias ofrecen un análisis o reformulación de fuentes primarias y a menudo intentan describirlas o explicarlas. Suelen resumir, interpretar, reorganizar y aportar valor a una fuente primaria. FJ Intelligence (2023) menciona que las fuentes secundarias: “No producen datos originales, sino que transforman información existente en conocimiento organizado y comprensible, facilitando el análisis de fenómenos complejos, especialmente en procesos de inteligencia competitiva y análisis estratégico” (párr. 1).

Al describir qué son las fuentes secundarias y cuál es su funcionamiento, se puede mencionar la gran importancia de esta fuente en cada investigación por realizar. Su relevancia radica en el hecho de que expande la investigación, lo que da más contexto a las fuentes primarias obtenidas. Esto adaptado a la propuesta de graduación, se puede notar en el momento en el que se describen las herramientas que se desean usar y sus complementos correspondientes, con el fin de llevar a cabo un sistema contable de manera correcta.

Fuentes terciarias

Según University of Minnesota Crookston (2026), las fuentes terciarias son aquellas que:

Indexan, resumen, organizan, compilan o resumen otras fuentes. Algunos materiales de referencia y libros de texto se consideran fuentes terciarias cuando su propósito principal es enumerar, resumir o simplemente reorganizar ideas u otra información. Las fuentes terciarias no suelen atribuirse a un autor en particular (párr. 6).

Al conocer qué son las fuentes terciarias, se puede llegar a la conclusión de que estas ayudan a generar un documento más completo y detallado al expandir el tema principal. En el trabajo de graduación elaborado se presentan varias fuentes terciarias que fueron utilizadas, con el fin de compilar ideas o resumirlas, lo que benefició en gran medida la comprensión de algunos temas y el desarrollo de estos. Como tal, este tipo de fuente siempre debe incluirse, ya que nutre de gran manera cualquier clase de investigación que se desee realizar.

Variables

De acuerdo con Stewart (2025):

Las variables son componentes fundamentales de la investigación que permiten medir y analizar datos. Pueden definirse como características o propiedades que pueden adoptar distintos valores. En diseño de la investigación, comprender los tipos de variables y sus funciones es crucial para elaborar hipótesis, diseño de métodos e interpretación de resultados (s. p.).

Con lo mencionado, se puede llegar a la idea de que las variables son de suma importancia al desarrollar una investigación, ya que estas definen términos, instrumentos por utilizar y las herramientas más adecuadas para dichos instrumentos. Lo anterior se puede ver reflejado enseguida con la definición de tres variables, las cuales se usan como medidas en busca de un buen desarrollo de la propuesta de graduación.

Variables conceptuales

Este tipo de variable se define, según Explorable.com. (s. f.), de tal forma que explica que las: “Variables conceptuales son generalmente expresadas en términos generales, teóricos, cualitativos o subjetivos. Asimismo, son importantes en el proceso de construcción de hipótesis” (párr. 3).

Con la explicación que se mencionó, se puede destacar la gran importancia de usar las variables conceptuales en una investigación, ya que estas exploran la información de manera general, lo que ayuda a crear una solución. Esto se debe a que contribuye a dar una contextualización más amplia de un tema o punto en específico.

Variables instrumentales

Las variables instrumentales son las herramientas que se utilizan con el fin de obtener información detallada de un problema o, a la vez, también funcionan como medio por el cual se explica un funcionamiento o se prueba este mismo. Estas herramientas suelen variar, ya

que no existe una sola forma de conseguir la respuesta correcta o el método correcto de probar o analizar un problema que se planteó.

Al utilizar lo anterior, se planteó para este tipo de proyecto de graduación el método de entrevistas, guías y diagramas, los cuales funcionan como base para solucionar, plantear o recolectar los datos necesarios para la debida elaboración de un sistema web que se enfoque en la contabilidad empresarial de Ópticas Dr. Max.

Variables operacionales

Según la página SalusPlay (2023), las variables operacionales: “Constituyen el conjunto de procedimientos que describe las actividades que un observador debe realizar para recibir las impresiones sensoriales, las cuales indican la existencia de un concepto teórico en mayor o menor grado” (párr. 14). Con lo mencionado, se puede afirmar que las variables operacionales especifican cuáles tareas u operaciones deben realizarse para medir una variable.

Con base en lo anterior, se puede mencionar una de las características que posee esta variable, la cual se considera la principal, ya que intenta descifrar aquellos conceptos de alto grado de entendimiento en elementos que puedan ser primero observados y medidos posteriormente.

Tabla 5

Variables

Objetivo específico	Variable	Variable conceptual	Variable instrumental	Variable operacional
Analizar los requerimientos funcionales que debe tener el sistema junto	Requerimientos del sistema	Según Science Direct (2026): “Los requisitos del sistema se definen como las condiciones o capacidades que un sistema	Entrevistas	Guía de entrevistas

con los no funcionales		o un componente del sistema deben cumplir o poseer para satisfacer un contrato, estándar, especificación u otros documentos impuestos formalmente, así como las condiciones o capacidades que necesita un usuario para resolver un problema o lograr un objetivo” (párr. 4).		
Construir el diseño de la base de datos mediante el uso de MySQL Workbench.	Diseño del sistema Diseño de la base de datos	Según Harith Abeyasinghe (2025): “El diseño de una base de datos consiste en definir la estructura de los datos que debe tener un sistema de información determinado. Para ello se suelen seguir por regla general unas fases en el proceso de diseño, definiendo para ello el modelo conceptual, el lógico y el físico” (párr. 6), De igual forma, Weibel (2026) menciona que: “El diseño de un sistema consiste en crear experiencias digitales consistentes, escalables y	Casos de uso y la documentación realizada	MySQL server Workbench

		fáciles de usar. Es un conjunto estructurado de directrices, herramientas y componentes reutilizables que mantienen a los equipos coordinados y los productos perfeccionados” (párr. 3).		
Programar los módulos mediante el uso de C# y con la ayuda de Entity Framework	Desarrollo del sistema	De acuerdo con el autor SentinelOne (2025): “El desarrollo de sistemas, también conocido como ciclo de vida del desarrollo de sistemas o SDLC, es un proceso formal para el desarrollo y mantenimiento de sistemas de información. El proceso es una actividad muy estructurada con etapas claramente definidas que abarcan el recorrido del desarrollo desde el concepto hasta la implementación final, de tal manera que el producto resultante se ajuste tanto a los objetivos empresariales como a los de los usuarios” (s. p.).	Diagramas de tipo UML y E/R	Entorno de desarrollo Visual Studio con el lenguaje C# y herramientas de .NET junto con la base de datos realizada en MySQL server y con el motor de la base de datos SQL Server Management Studio

Probar los diferentes módulos de forma independiente y también de manera integral garantizando los resultados correctos del sistema.	Pruebas del sistema	De acuerdo con Das (2025): “Las pruebas del sistema verifican el correcto funcionamiento de toda la aplicación en un entorno real. Ayudan a identificar problemas de integración o funcionalidad antes del lanzamiento del producto” (s. p.).	Pruebas al sistema	Casos de prueba
--	---------------------	---	--------------------	-----------------

Fuente: Elaboración propia

Instrumentos de recolección de datos

Los instrumentos de recolección de datos cumplen un rol fundamental al realizar este proceso del proyecto de graduación. Esto último se debe a que se utilizan diversas herramientas al intentar recolectar la mayor parte de información de un lugar, en este caso la Óptica Dr. Max. Se recomienda usar las herramientas adecuadas, ya que no todas funcionan de esta forma ni tienen la misma efectividad al llevar a cabo el estudio, con el fin de extraer información beneficiosa para el debido desarrollo de la solución que se planteó.

La observación

Según Concepto.de (2021b): “La observación es el proceso mediante el cual se obtiene información sobre un fenómeno de la realidad a través de los sentidos” (párr. 10). Por otro lado:

Se trata de una de las primeras etapas de cualquier investigación, especialmente de las que se rigen por el método científico. En este sentido, consiste en la recopilación directa de datos a partir de la naturaleza mediante trabajos de campo o de laboratorio, en los que se utilizan diferentes instrumentos de medición (Concepto.de, 2021b, párr. 10).

La observación es un método que puede beneficiar la elaboración del sistema. Esto se debe a que se utiliza a menudo, con el fin de obtener respuestas de manera rápida al solo observar procesos que deben funcionar de forma eficiente. Lo mencionado convierte a este método en uno de los más usados, sin embargo, no basta con la sola utilización de este al querer profundizar en los problemas que se presentan.

La entrevista

El propósito fundamental de una entrevista como método de recolección de datos consiste en obtener respuestas verbales, con el fin de recopilar datos de manera exhaustiva. Los objetivos de la entrevista pueden incluir el intercambio de ideas y experiencias, así como la extracción de datos sobre una amplia variedad de temas. Se realizan diversos tipos de entrevistas para conseguir datos de investigación.

Según la página Unacademy (2022), existen diversos tipos de entrevistas con las que se puede obtener información relevante para extraer datos que benefician la investigación que la persona desee plantear. Algunos tipos de entrevistas que se pueden usar son los siguientes: entrevista directa estructurada, entrevista directa no estructurada, entrevista indirecta estructurada, entrevista indirecta no estructurada, entrevistas semiestructuradas, entrevista telefónica y entrevista personal.

Proceso para la recolección y el análisis de datos

Al conocer cuáles instrumentos se piensan usar en el momento de recolectar los datos necesarios para la debida propuesta de graduación, también se debe evaluar qué proceso es necesario seguir y en qué orden. El proceso de recolección de datos suele ser uno de los puntos más importantes al iniciar una investigación. Una vez que se obtienen estos datos, se debe seguir un proceso de análisis exhaustivo de dicha información. Esto es importante, ya que determina los problemas que tiene la empresa en su contabilidad y sus respectivas soluciones.

Para la entrevista se planea realizar un conjunto de diez preguntas que se le harán al gerente de Ópticas Dr. Max, con el fin de obtener los datos que posteriormente se analizan para plantear los módulos que debe llevar el sistema y cómo estos resolverán los problemas que tiene la empresa. El proceso de análisis de los datos que se obtienen se realiza conforme a la necesidad principal del negocio respecto a la contabilidad empresarial. Lo mencionado se plantea de esta forma, ya que al enfocarse en el problema principal se puede llegar a una conclusión que beneficie el desarrollo del sistema.

El proceso por seguir para la debida recolección de datos en el método de observación consiste en visitar la empresa que se ubica en Heredia. Esto último es fundamental, ya que este método consiste en realizar un estudio de los datos mediante la observación de problemas o puntos de mejora. Posterior a la recopilación de datos, se deben estudiar los hallazgos por medio de un análisis de la situación actual de la compañía. Lo anterior logra un entendimiento sobre cómo funciona la empresa y cuáles puntos es necesario mejorar con respecto a la contabilidad que esta posee.

Capítulo IV: Análisis de los resultados

En este capítulo se analizan los resultados del uso de los instrumentos de recolección de datos. Para esto se utilizó el instrumento de la entrevista, en el cual se realizó una cantidad de preguntas al gerente de la empresa Ópticas Dr. Max, con el fin de obtener los problemas reales que tiene la compañía y cómo se pueden solucionar mediante el sistema web. Como segundo instrumento se usó la observación detallada de los procesos que se emplean en la empresa, con la finalidad de comprender los métodos que se utilizan y poder aplicarlos en el sistema.

Análisis del instrumento de la entrevista

Pregunta n.º 1:

¿Cómo verifica usted que el dinero recibido durante el día coincide con las ventas realizadas?

Objetivo: el objetivo de esta pregunta es conocer el proceso de cierre de caja que utiliza la empresa. Lo anterior tiene el fin de obtener información que valide la operación que se usa y cómo se puede implementar en un sistema contable.

Respuesta: se hace un cierre de caja en el que se dividen los ingresos dependiendo si fueron en efectivo o tarjeta, después se hace una suma para obtener el montón total del día.

La respuesta dada por el gerente indica el proceso exacto que utilizan en la empresa para realizar el cierre de caja. Además, se logró obtener información de que reciben pagos en tarjeta y efectivo, lo cual es útil, ya que brinda una respuesta a las interrogantes que se plantearon en el objetivo.

Pregunta n.º 2:

¿Cuáles son los principales gastos operativos y cómo llevan el control de esos gastos?

Objetivo: el objetivo de esta pregunta es conocer algunos de los gastos de la empresa y saber cómo llevan el control de estos.

Respuesta: los pagos de facturas a los distintos proveedores que tenemos y gastos públicos (agua, luz, teléfono, Internet).

La respuesta dada por el gerente contestó cuáles son los gastos que tiene la empresa. No obstante, no se logró obtener una respuesta sobre el control que utiliza la compañía. La pregunta es importante, ya que al conocer los tipos de gastos se puede realizar un control en el sistema. El control que llevan de estos se pudo ver con el instrumento de la observación.

Pregunta n.º 3:

Cuando se realiza una venta ¿qué información colocan en la factura?

Objetivo: conocer los datos que se agregan a una factura realizada por la empresa.

Respuesta: tiene un numero de factura, en ciertos casos el nombre del cliente, tipo de pago, monto, servicio o elemento vendido, la fecha y hora de la emisión de la factura. Si el cliente no realiza el pago de a contado y opta por el beneficio que tenemos de pagos a plazos en la factura le agregamos el plazo y la cantidad a pagar por plazo.

El gerente responde la pregunta y proporciona los datos más importantes que se incluyen en la factura que realiza la empresa y que se entrega al cliente. Estos datos ayudan a realizar un apartado del sistema que contenga las facturas. Además, incluye a los clientes que optaron por el beneficio que brinda la compañía.

Pregunta n.º 4:

¿Cómo registran y controlan los pagos pendientes que tiene la empresa?

Objetivo: el objetivo de esta pregunta es conocer la forma en la que la empresa registra y controla las cuentas por pagar. Lo anterior tiene el fin de identificar mejoras que puedan incorporarse en el sistema contable.

Respuesta: se manejan de manera manual en un Excel, en una hoja se agregan los gastos y pagos pendientes del mes.

La respuesta del gerente permitió conocer que el control de los pagos pendientes se realiza de forma manual mediante hojas de Microsoft Excel. Esto evidencia la necesidad de un sistema más automatizado que reduzca errores y facilite el seguimiento de las obligaciones pendientes de la empresa.

Pregunta n.º 5:

¿Cuáles tipos de gastos pequeños se pagan usualmente con el dinero disponible en la caja chica y cómo se registra actualmente cada uno de esos movimientos?

Objetivo: el objetivo de esta pregunta es identificar los gastos menores que se cubren con caja chica y conocer cómo se registran actualmente, para mejorar su control en el sistema contable.

Respuesta: objetos de oficina (papel, folder, lapiceros, etc.), limpieza (bolsas de basura, desinfectantes, etc.) y de uso del personal en el local (papel higiénico, etc.), actualmente tenemos esos gastos y segmentados de esta forma.

Con base en la pregunta, se identificaron los principales gastos que se cubren mediante caja chica y la forma en la que se clasifican actualmente. Esta información es útil para diseñar un mejor control en el sistema, lo que permite registrar cada movimiento de manera ordenada y precisa.

Pregunta n.º 6:

¿Existen otras fuentes de ingreso, como servicios de exámenes de la vista o reparaciones y, si existen, cómo registran estos ingresos?

Objetivo: el objetivo de esta pregunta es conocer si la empresa cuenta con otras fuentes de ingreso además de la venta de productos y determinar cómo se registran entre su control financiero.

Respuesta: sí tenemos lo que son los exámenes y reparaciones. Todos los servicios brindados en la óptica se facturan, el cambio se da en la factura porque se especifica el servicio o el elemento vendido.

Al conocer los ingresos que genera la empresa mediante servicios como exámenes de la vista y reparaciones, se puede realizar una sección del sistema que registre estos ingresos, de mejor manera. Además, se confirmó que todos estos ingresos se registran por medio de facturación, lo cual facilita integrar estos servicios en el sistema contable.

Pregunta n.º 7:

¿Cómo organizan actualmente la información financiera de la óptica para saber cuánto dinero entra y cuánto sale?

Objetivo: el objetivo de esta pregunta es conocer la forma en la que la empresa organiza su información financiera. Lo anterior tiene el fin de evaluar su método actual y proponer mejoras mediante el sistema contable.

Respuesta: al final del día se cuenta el dinero de la caja, además de los pagos con tarjetas, el valor obtenido se guarda en un Excel.

La respuesta muestra que la empresa lleva un control financiero básico mediante el conteo diario de ingresos y su posterior registro en Microsoft Excel. Esto permite identificar una oportunidad de mejora, ya que un sistema contable facilitaría un control más preciso de las cuentas que tiene la compañía, organizándolas en un libro contable que incluya el nombre de la cuenta, un código y el monto. Esto ayuda a una detección rápida de los montos que se obtienen en cada cuenta.

Pregunta n.º 8

¿Qué tipo de información financiera le gustaría poder consultar fácilmente para tomar decisiones sobre el negocio?

Objetivo: el objetivo de esta pregunta es identificar qué información financiera considera importante la empresa para la toma de decisiones. Lo anterior tiene el fin de incluirla en el sistema contable propuesto.

Respuesta: ver los gastos exactos de la caja chica, el dinero disponible en la caja chica, las facturas pendientes y un historial de facturas pagas, además de los ingresos obtenidos, ya sean diarios o mensuales.

Según lo mencionado por el gerente, se permite conocer cuáles son las principales necesidades de información financiera de la empresa. Estos datos servirán como guía para el desarrollo del sistema contable, asegurando que cumpla con las expectativas y requerimientos reales del negocio.

Pregunta n.º 9:

¿Cómo funciona el método que ustedes utilizan para controlar cuando un cliente queda debiendo dinero de algún servicio que se brinda?

Objetivo: se planteó esta pregunta, con la finalidad de conocer el método que se utiliza para controlar las cuentas por cobrar y cómo se administra este proceso en la empresa.

Respuesta: por políticas no damos fiado, entonces eso no es un problema, sin embargo, nosotros tenemos lo que viene siendo un beneficio que si el cliente no puede o no quiere pagar de contado le damos a un plazo 3 meses, el cliente con este beneficio puede cancelar un producto o un servicio de manera más cómoda.

Se obtuvieron datos importantes que permiten conocer que la empresa no trabaja con fiado, pero ofrece pagos a plazos como una facilidad para sus clientes. Estos datos ayudan a la planificación de un módulo de cuentas por cobrar que permita al gerente verificar de manera rápida los clientes y los plazos que estos tienen.

Pregunta n.º 10:

¿La empresa establece algún presupuesto mensual o anual para controlar los gastos o ingresos del negocio y cómo revisan este presupuesto para tomar acciones en caso de que lo requiera?

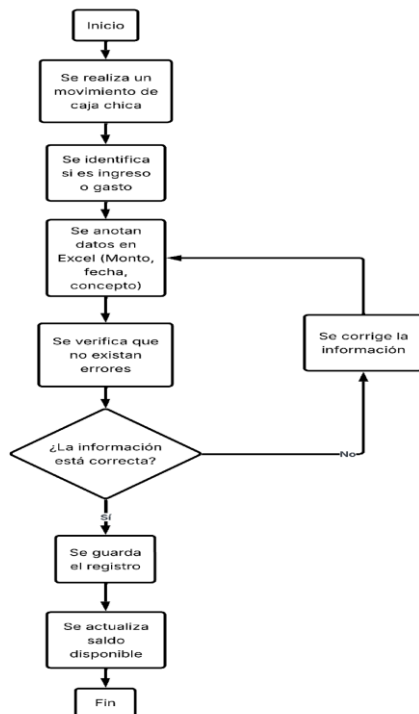
Objetivo: tiene como objetivo principal conocer los plazos que utiliza la empresa al realizar el presupuesto que posee.

Respuesta: el presupuesto se da mensualmente y nosotros con base a lo que tenemos nos organizamos para los gastos que se den, pero es necesario tener un orden o porque a veces por apuntar las cosas de forma manual hay ciertos errores de cálculo que afectan al presupuesto mensual estimado.

Con base en lo mencionado por el gerente, la empresa trabaja con un presupuesto mensual, pero su control actual genera errores que afectan la planificación financiera. Esto refuerza la necesidad de implementar un sistema contable que permita un registro más preciso, lo que ayuda en gran medida a la toma de decisiones de la compañía.

Análisis del instrumento: la observación

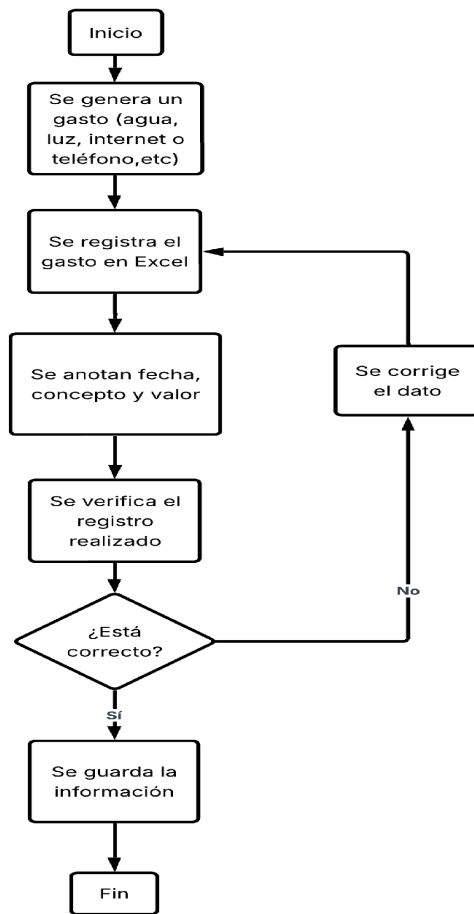
Para el debido análisis de los datos requeridos, se efectuó un instrumento de recolección que consiste en observar y analizar los procesos que ejecuta la empresa, con el fin de comprender y brindar puntos de mejora. Lo anterior ayuda con la elaboración del sistema contable que facilita el método actual que utilizan en Ópticas Dr. Max. Enseguida, se brinda un diagrama de flujo que muestra el proceso actual por cada observación realizada, junto con el punto de mejora recomendado.

Figura 5**Observación de la caja chica**

Fuente: Elaboración propia

Objetivo: se busca observar qué método utiliza la empresa al registrar los movimientos de la caja chica. Esto con el fin de obtener información útil para el futuro sistema.

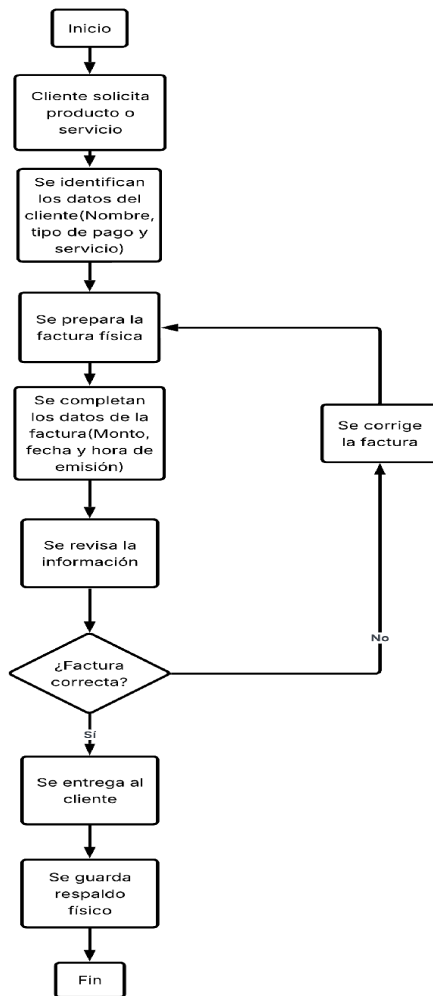
Como punto de mejora, se puede realizar un sistema que contenga todos los datos que se deseen registrar sobre los movimientos de la caja chica. Lo anterior tiene el fin de reducir errores que se puedan cometer al llevar a cabo los cálculos correspondientes. Asimismo, ayuda a mantener los datos seguros y de rápido acceso, lo que facilita el trabajo.

Figura 6**Registro de gastos**

Fuente: Elaboración propia

Objetivo: se busca observar cómo la empresa registra los diferentes gastos operativos. Lo anterior tiene el fin de identificar los métodos que se utilizan y obtener información útil para el diseño del sistema.

Se puede realizar un sistema que ayude con este proceso, en donde se puedan obtener estos datos de manera segura y rápida. Esta observación ayudó a recolectar los datos y el procedimiento que se utiliza al registrar los gastos, lo que beneficia en gran medida el desarrollo del sistema contable, ya que se tienen los puntos por mejorar del proceso actual.

Figura 7**Creación de facturas**

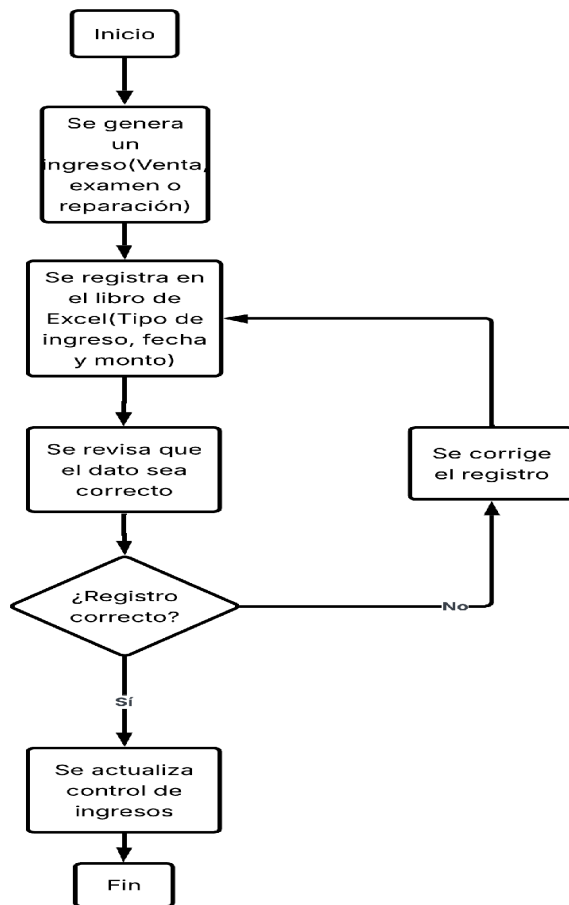
Fuente: Elaboración propia

Objetivo: se busca observar el procedimiento que utiliza la empresa en el momento de emitir facturas. Lo anterior tiene el fin de conocer los datos que se registran y utilizarlos como referencia para el sistema.

El método de recolección de datos empleado en esta prueba fue satisfactorio, ya que cumplió con su objetivo de verificar y conocer el proceso de facturación que utiliza la empresa. Como punto de mejora, se puede elaborar un sistema que almacene las facturas y permita crear nuevas, teniendo un historial que ayude a organizar las cuentas de la compañía.

Figura 8

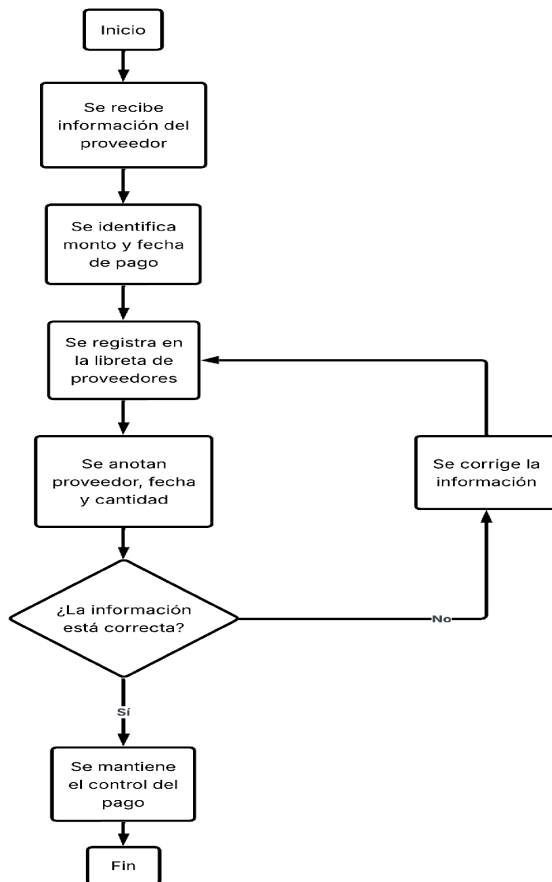
Registro de ingresos



Fuente: Elaboración propia

Objetivo: se busca observar la forma en la que la empresa registra los ingresos generados por las ventas o servicios ofrecidos.

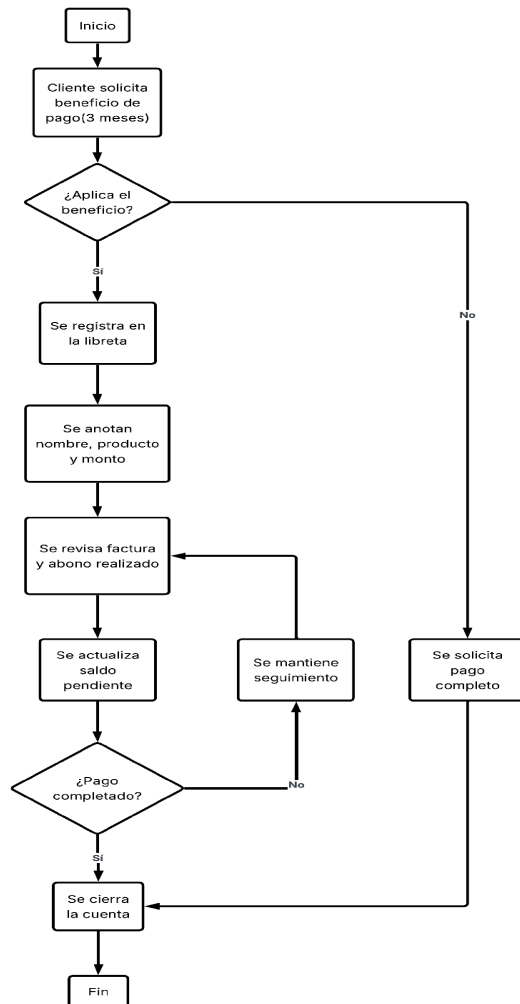
La observación de este proceso determina que el método que se utiliza en el registro de ingresos es el que se emplea en su contraparte, los gastos. Como punto de mejora, al realizar un sistema contable se puede acceder a estos datos de manera rápida, lo que aporta un alto grado de confianza al saber que estos datos se encuentran seguros.

Figura 9**Cuentas por pagar**

Fuente: Elaboración propia

Objetivo: se busca observar cómo la empresa lleva el control de los pagos que debe realizar a los proveedores. Lo anterior tiene el fin de identificar el método que se utiliza para registrar y dar seguimiento a estas obligaciones.

Como punto de mejora que se puede realizar en el sistema contable gracias a los datos que se obtienen mediante la observación de este proceso, se puede añadir un apartado en el sistema que contenga las cuentas por pagar de la empresa con los proveedores. Esto mejora la obtención de estos datos y facilita el monitoreo de dichas cuentas.

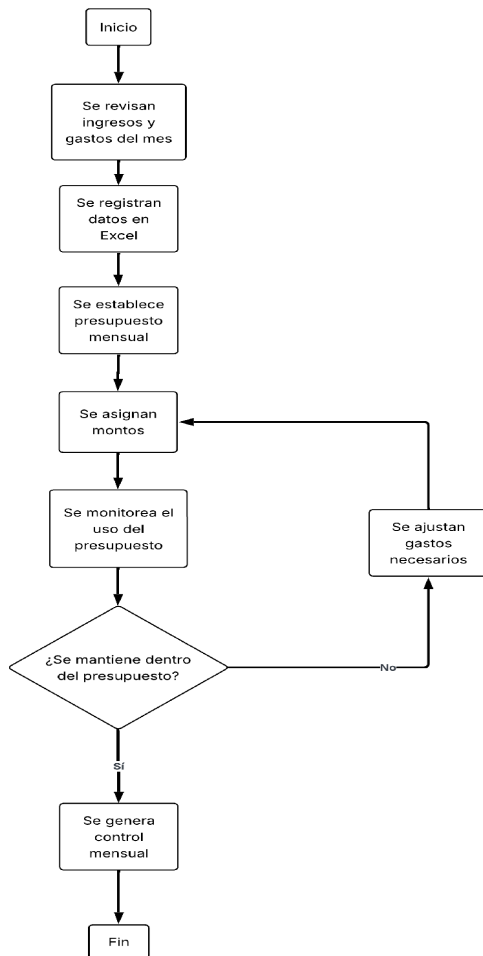
Figura 10**Cuentas por cobrar**

Fuente: Elaboración propia

Objetivo: se busca observar la forma en la que la empresa registra y da seguimiento a los montos pendientes de pago por parte de los clientes.

Al realizar la observación correspondiente del proceso que tiene la empresa sobre las cuentas por cobrar a los clientes, se obtuvieron datos beneficiosos que ayudan al desarrollo del sistema contable. Al implementar un sistema, se puede tener un mejor registro de los clientes que tienen una cuenta por pagar con la compañía, lo que beneficia en gran medida a la empresa, ya que se asegura mantener los datos al día y correctamente guardados.

Figura 11
Control de presupuesto



Fuente: Elaboración propia

Objetivo: se busca observar si la empresa establece algún control o planificación sobre sus gastos e ingresos mediante un presupuesto. Lo anterior tiene el fin de conocer el método que se utiliza y considerar esta funcionalidad en el sistema.

Con base en el análisis de los datos que se obtienen de la observación realizada sobre el método que utiliza la empresa para elaborar un presupuesto, se puede brindar un punto de mejora, el cual consiste en implementar en el sistema un apartado que ayude a realizar el

presupuesto mensual de la compañía. De esta manera, se puede ver la cantidad de dinero que se gastó de ese presupuesto y el total que sobró.

Capítulo V: Propuesta

En este capítulo se desarrolla la propuesta del trabajo final de graduación, la cual incluye los módulos con su respectivo funcionamiento y el problema que ayudó a planificarlos. De igual forma, se realizan los diseños necesarios para la debida documentación del sistema contable que lleva como título: *Sistema web para la gestión contable de la empresa Ópticas Dr. Max, ubicada en Heredia.*

Requerimientos del sistema

Cada desarrollo que se lleve a cabo debe cumplir con los requisitos que se establecen según la necesidad de la empresa a la cual se le realiza el sistema. La compañía llamada Ópticas Dr. Max cuenta con problemas en el sector contable, ya que realizan procesos de forma manual, lo que genera errores contables que desencadenan molestias en los clientes y proveedores. Se planificaron nueve módulos principales que resolverán los inconvenientes financieros de la empresa que se mencionó.

Gestionar cuentas contables

REQF-001. Permitir registrar cuentas contables con código (por ej., 1.1.01), nombre, tipo y naturaleza (deudora o acreedora).

REQF-002. Permitir editar y desactivar cuentas contables conservando su historial de movimientos.

REQF-003. Permitir reactivar cuentas desactivadas, validando previamente que no exista otra cuenta activa con el mismo código.

REQF-004. Permitir consultar el libro contable de cualquier cuenta filtrando por fechas.

REQF-005. Mostrar los movimientos del libro contable con las columnas *Entra* y *Sale* según la naturaleza de la cuenta.

REQF-006. Registrar automáticamente entradas de reversa en el libro mayor al anular facturas o eliminar movimientos.

REQF-007. Recalcular el balance actual de cada cuenta según el movimiento de caja, las facturas pagadas y los asientos contables.

REQF-008. Permitir gestionar los tipos de cuenta contable (activo, pasivo, capital, ingreso y gasto) con validación de duplicados.

REQF-009. Permitir gestionar los tipos de movimiento de caja con marcadores por módulo (caja chica, gastos, CxP, CxC e ingresos).

REQF-010. Permitir gestionar los métodos de pago con el marcador *Es efectivo* que se utiliza en el cierre de caja.

REQF-011. Validar antes de reactivar cualquier catálogo que no exista otro registro activo con el mismo nombre.

Gestionar facturación

REQF-012. Permitir crear facturas tipo *contado* o *crédito* con un número electrónico único e irrepetible.

REQF-013. Permitir agregar múltiples líneas de detalle a una factura con producto, cantidad, precio e IVA.

REQF-014. Calcular automáticamente el subtotal, el IVA y el total de cada factura.

REQF-015. Permitir definir plazos de pago de 1 a 3 meses para facturas a crédito.

REQF-016. Permitir anular facturas en cascada (restaurar *stock*, anular CxC asociada) y bloquear la anulación si la factura tiene abonos.

REQF-017. Permitir reactivar facturas anuladas revirtiendo la cascada de anulación.

REQF-018. Mostrar el detalle de la factura al identificar visualmente si es de contado o crédito.

Gestión de cuentas por cobrar

REQF-019. Crear automáticamente una cuenta por cobrar (CxC) al emitir una factura a crédito.

REQF-020. Generar los plazos numerados de la CxC según el período acordado en la factura.

REQF-021. Permitir registrar abonos parciales o totales a una CxC seleccionando el plazo por cancelar.

REQF-022. Generar un recibo imprimible al registrar cada abono.

REQF-023. Bloquear la anulación de una CxC que tenga abonos registrados.

REQF-024. Bloquear la reactivación de una CxC si su factura asociada está anulada.

Gestión de cuentas por pagar

REQF-025. Permitir registrar manualmente cuentas por pagar (CxP) con proveedor, monto y plazos.

REQF-026. Permitir registrar abonos a una CxP.

REQF-027. Permitir editar solo las CxP de origen manual, no las que se generan automáticamente por compras.

REQF-028. Bloquear la anulación de una CxP con abonos registrados.

REQF-029. Generar el asiento contable de egreso automático al pagar cada cuota.

REQF-030. Permitir reactivar CxP anuladas.

Gestionar los fondos de la caja chica

REQF-031. Permitir registrar movimientos de caja chica (ingreso o egreso) con validación del límite configurado.

REQF-032. Permitir editar y eliminar movimientos de caja chica, lo que genera la reversa contable correspondiente a la cuenta afectada.

REQF-033. Permitir configurar el límite máximo de caja chica.

Gestionar cierre de caja

REQF-034. Realizar el cierre de caja diario, lo que consolida automáticamente las facturas del día y los demás movimientos.

REQF-035. Mostrar el resumen del cierre (Ingresos/Gastos/Neto).

REQF-036. Identificar dinámicamente si el cierre ocasiona ganancia, pérdida o margen ajustado y mostrarlo de manera visual con colores.

REQF-037. Bloquear la creación de más de un cierre por usuario en la misma fecha.

REQF-038. Permitir al administrador eliminar cierres antiguos.

Calcular los gastos

REQF-039. Permitir registrar gastos con categoría, monto, fecha y método de pago.

REQF-040. Permitir administrar el catálogo de categorías de gasto.

REQF-041. Permitir eliminar gastos, lo que genera la reversa contable correspondiente.

REQF-042. Mostrar el resumen de gastos por categoría y por período.

Gestión del presupuesto

REQF-043. Permitir crear períodos de presupuesto mensuales con nombre único y monto global.

REQF-044. Generar la proyección automática del presupuesto, lo que consolida movimientos de caja, facturas pagadas y asientos contables.

REQF-045. Permitir agregar y eliminar detalles de presupuesto por cuenta contable.

REQF-046. Permitir eliminar un presupuesto completo con sus detalles asociados en cascada.

Gestión de ingresos

REQF-047. Permitir visualizar los ingresos de facturas realizadas a clientes.

REQF-048. Permitir filtrar las fechas para ver los ingresos en un período.

REQF-049. Permitir ver los movimientos de cada cuenta de ingresos.

Seguridad

REQF-050. Asignar roles a los usuarios y restringir el acceso a los módulos según el rol asignado.

Consultas

REQF-051. Permitir realizar consultas en los diversos módulos mediante filtros.

Reportes

REQF-052. Permitir generar reportes en Microsoft Excel y PDF en los diversos módulos.

Mantenimientos

REQF-053. Debe permitir crear, editar y eliminar los datos correspondientes en los módulos.

Matriz de requerimientos

Al obtener los requerimientos principales que debe cumplir el sistema contable, junto con los problemas actuales que llevaron a la planificación de estos, como siguiente paso se debe realizar una matriz de requisitos que compacte la información obtenida mediante los métodos de investigación y el estudio realizado de los datos que se obtienen. Esto beneficia al desarrollo del sistema al tener de manera clara los problemas y su respectiva solución que se planteó.

Tabla 6

Matriz de requerimientos

Indicador	Nombre del módulo	Requerimientos
1	Gestionar los montos de la caja chica	REQF-033, REQF-032, REQF-031
2	Gestionar cierre de caja	REQF-034, REQF-035, REQF-036, REQF-037, REQF-038
3	Calcular los gastos	REQF-039, REQF-040, REQF-041, REQF-042
4	Gestionar facturación	REQF-012, REQF-013, REQF-014, REQF-015, REQF-016, REQF-017, REQF-018
5	Gestión de cuentas por pagar	REQF-025, REQF-026, REQF-027, REQF-028, REQF-029, REQF-030
6	Gestión del presupuesto	REQF-043, REQF-044, REQF-045, REQF-046
7	Gestión de ingresos	REQF-047, REQF-048, REQF-049
8	Gestionar cuentas contables	REQF-001, REQF-002, REQF-003, REQF-004, REQF-005, REQF-006, REQF-007, REQF-008, REQF-009, REQF-010, REQF-011.
9	Gestión de cuentas por cobrar	REQF-019, REQF-020., REQF-021, REQF-022, REQF-023, REQF-024
10	MANTENIMIENTOS	REQF-053
11	CONSULTAS	REQF-051
12	REPORTES	REQF-052
13	SEGURIDAD	REQF-050

Fuente: Elaboración propia

Análisis detallado del software

El *software* es una parte crucial al desarrollar todo sistema, ya que en este se encuentran las partes más importantes en temas de programación y desarrollo de bases de datos. Para elaborar el sistema contable, se utilizan distintas herramientas que ayuden a crear un sistema que cumpla con los requerimientos que se plantearon.

Según Arimetrics (s. f.): “Visual Studio es un entorno de desarrollo integrado (IDE) desarrollado por Microsoft que permite a los programadores crear aplicaciones para distintas plataformas (incluyendo Windows, Android, iOS y Linux)” (s. p.). Este es el motor en el cual se programa la parte lógica (*back-end*) y la parte visual (*front-end*). Para desarrollar el *back-end* se utiliza el lenguaje C# (C Sharp) junto con la herramienta Entity Framework, la cual facilita la conexión con la base de datos. Por otro lado, para la parte visual, se trabaja con JavaScript, C++ y la herramienta Razor.

SQL Management Studio es el motor que se utiliza para la administración de la base de datos, lo que permite gestionar consultas, procedimientos almacenados y demás componentes necesarios para el funcionamiento correcto del sistema contable. SQL Server es el gestor encargado de la creación y almacenamiento de la base de datos, lo que incluye las tablas, relaciones y procesos almacenados, garantizando un manejo seguro y eficiente de la información contable. Además, se usa MySQL Workbench como herramienta de apoyo para la diagramación y modelado de las tablas de la base de datos, lo que da la posibilidad de una mejor planificación, organización y visualización de la estructura de los datos antes de implementarla.

Análisis detallado del hardware

Los componentes de *hardware* para el debido funcionamiento del sistema que se planteó son los siguientes: computadoras de escritorio, monitores de 24 in Xiaomi A22i IPS 1920x1080 75 Hz, procesador Intel Core i5, memoria RAM de 8 GB, almacenamiento SSD de 250 GB, teclado y ratón.

Las computadoras de escritorio son el equipo principal para la ejecución del sistema contable, ya que permiten realizar las operaciones administrativas, contables y de atención al cliente de manera eficiente. Los monitores de 24 in Xiaomi A22i IPS con resolución Full HD permiten una mejor visualización de la información financiera, reportes, facturación y demás procesos contables. Su tamaño y calidad de imagen facilitan el trabajo prolongado del personal administrativo, lo que reduce la fatiga visual.

El procesador Intel Core i5 proporciona la capacidad de procesamiento necesaria para ejecutar el sistema de manera fluida. Esto permite el uso simultáneo de varias aplicaciones sin afectar el rendimiento general del equipo ni del sistema. La memoria RAM de 8 GB da la posibilidad de mantener un rendimiento estable durante la ejecución del sistema y de otras aplicaciones que utiliza la empresa.

Análisis detallado de los elementos que se relacionan con las telecomunicaciones

El programa está planeado para que sea escalable a largo plazo, ya que se piensa usar tecnologías en la nube, lo que facilita la escalabilidad del programa. Algunas de las tecnologías en la nube que se plantearon pueden variar entre Google Cloud o Azure. Para los servicios en la nube se tiene que contratar al proveedor en la nube, esto incluye temas de base de datos y respaldos de documentación.

La red de área local que se usa se llama LAN; esta es la red principal del sistema y corresponde a la conexión de los equipos necesarios, como las computadoras. Se utiliza la red Ethernet cableada, que se conectará directamente a los equipos de trabajo. Para este sistema, se emplea una variedad de equipos requeridos para utilizar el sistema de manera correcta y segura.

Se utilizan *routers* de doble banda (2.4 GHz y 5 GHz), *switches* de red de 8 puertos o de 24 puertos PoE para la conexión de los equipos que requieran conexión por medio de cable de red, repetidores wifi y *access points* que ayudan a extender la señal wifi en lugares donde esta no llegue. Además, se usan unidades de suministro llamadas (UPS) o generadores, los cuales son esenciales para garantizar que todos los equipos críticos (servidores, *firewalls*, *switches* centrales) sigan funcionando durante fallos eléctricos.

Descripción detallada de las herramientas técnicas que se utilizan

Las herramientas que se utilizan en este trabajo final de graduación son las siguientes: Visual Studio, SQL Server Management Studio y MySQL Workbench. Cada una de las herramientas mencionadas cumple un rol fundamental al desarrollar el sistema contable y se seleccionaron por su gran versatilidad y seguridad.

En Visual Studio se realiza el *back-end* y *front-end* del sistema, esto gracias a la ayuda de Entity Framework, Razor y el lenguaje de programación C#. No obstante, antes de desarrollar lo mencionado, se debe crear la base de datos en SQL Server Management Studio. En esta herramienta se crean las tablas necesarias para un funcionamiento correcto al desarrollar la parte del *back-end*.

Por último, cabe mencionar la herramienta MySQL Workbench, en la cual se realiza el diagrama de la base de datos. Este diagrama sirve como guía de desarrollo, facilita el proceso de creación de la base de datos y minimiza posibles errores que puedan ocurrir

Casos de uso

Tabla 7

Caja chica 1

Número del caso de uso:	CU-CAJ-01
Fecha de elaboración:	25/04/2026
Descripción del caso de uso:	El administrador registra un movimiento de ingreso o egreso en la caja chica. El sistema valida que el tipo de gasto esté definido y que el monto no supere el límite asignado, actualizando el saldo disponible en tiempo real.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión en el sistema.

	El tipo de gasto debe estar definido en el catálogo. La caja chica debe tener un monto límite asignado.
Flujo básico del caso de uso	
	1. El administrador selecciona la opción <i>Registrar movimiento de caja chica</i>. 2. El sistema muestra el formulario de registro con los campos: tipo de movimiento, clase de gasto, monto, fecha y descripción. 3. El administrador completa los datos y confirma el registro. 4. El sistema valida que el tipo de gasto exista en el catálogo. 5. El sistema verifica que el monto no supere el límite asignado. 6. El sistema guarda el movimiento y actualiza el saldo disponible. 7. El sistema muestra el resumen del saldo actualizado al administrador.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Ver historial	El administrador selecciona <i>Ver historial de movimientos</i>. El sistema muestra la lista de movimientos del período actual con fecha, tipo y monto. El administrador puede filtrar por fecha o tipo de gasto.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Tipo de gasto no definido	Si el tipo de gasto no existe en el catálogo, el sistema notifica al usuario. El sistema sugiere crear la categoría antes de continuar. El flujo regresa al paso 3.
FA_02 Monto supera el límite	Si el monto registrado supera el límite asignado, el sistema genera una alerta visual.

	El administrador decide si corrige el monto o confirma el sobrepaso con justificación.
Requerimientos especiales:	El saldo debe actualizarse en tiempo real sin necesidad de recargar la página.
Poscondiciones	El saldo de la caja chica refleja el nuevo valor actualizado.

Fuente: Elaboración propia

Tabla 8

Caja chica 2

Número del caso de uso:	CU-CAJ-02
Fecha de elaboración:	25/04/2026
Descripción del caso de uso:	El administrador define o modifica el monto límite asignado a la caja chica. El sistema almacena el valor y lo utiliza como referencia para generar alertas cuando se exceda en futuros movimientos.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión con perfil de administrador. El módulo de caja chica debe estar activo en el sistema.
Flujo básico del caso de uso	
	1. El administrador accede a la configuración de caja chica. 2. El sistema muestra el monto límite actual y los campos para modificarlo. 3. El administrador ingresa el nuevo monto límite. 4. El sistema valida que el monto sea mayor que cero. 5. El sistema guarda el nuevo límite.

Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Ver historial de límites	El administrador selecciona <i>Historial de cambios de límite</i>. El sistema muestra los cambios anteriores con fecha, usuario y monto previo. El administrador puede comparar valores históricos.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Monto inválido	Si el monto ingresado es cero o negativo, el sistema muestra un mensaje de error. El administrador debe corregir el valor antes de continuar.
Requerimientos especiales:	El sistema debe guardar un registro de auditoría de cada cambio de límite.
Poscondiciones	El nuevo monto límite queda almacenado y activo para las validaciones de futuros movimientos de caja chica.

Fuente: Elaboración propia

Tabla 9
Caja chica 3

Número del caso de uso:	CU-CAJ-03
Fecha de elaboración:	25/04/2026
Descripción del caso de uso:	El administrador consulta el saldo disponible actual de la caja chica. El sistema muestra el monto asignado, el total de gastos y el saldo restante, con una alerta si el saldo es bajo.

Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión en el sistema. Debe existir al menos un período de caja chica activo.
Flujo básico del caso de uso	
	1. El administrador accede al módulo de caja chica. 2.El sistema calcula automáticamente el saldo disponible (monto asignado menos total de gastos). 3.El sistema muestra el panel de resumen con monto asignado, total gastos, total ingresos y saldo actual. 4.Si el saldo es menor al 20 % del límite, el sistema muestra un indicador de alerta. 5.El administrador puede seleccionar un período específico para ver el saldo histórico.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Desglose por categoría	El administrador selecciona <i>Ver desglose por categoría</i>. El sistema muestra cuánto se ha gastado por cada tipo de gasto definido.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Sin movimientos registrados	Si no hay movimientos en el período, el sistema muestra el saldo igual al monto asignado. El sistema indica que no hay gastos registrados en el período seleccionado.
Requerimientos especiales:	El saldo debe calcularse en tiempo real con cada consulta.

Poscondiciones	El administrador obtiene la información actualizada del saldo de caja chica.
-----------------------	--

Fuente: Elaboración propia

Tabla 10

Cierre de caja 1

Número del caso de uso:	CU-CIE-01
Fecha de elaboración:	25/04/2026
Descripción del caso de uso:	El cajero ejecuta el cierre de caja al finalizar el día. El sistema totaliza automáticamente los ingresos agrupados por método de pago (efectivo y tarjeta) y genera el resumen contable del período.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados :	Cajero, administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión con perfil de cajero o administrador. No debe existir un cierre de caja previo para el mismo día.
Flujo básico del caso de uso	
	1. El cajero selecciona la opción <i>Realizar cierre de caja</i>. 2. El sistema muestra el resumen de transacciones del día agrupadas por método de pago. 3. El cajero verifica los totales mostrados por el sistema. 4. El cajero ingresa el monto físico contado en la caja para efectivo. 5. El sistema compara el monto físico con el total calculado y muestra la diferencia. 6. El cajero confirma el cierre de caja.
Subflujos	

Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Ver detalle por transacción	El cajero selecciona <i>Ver detalle de transacciones</i>. El sistema lista cada transacción del día con monto y método de pago. El cajero puede identificar transacciones específicas antes de confirmar el cierre.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Sin transacciones	Si no hay transacciones en el día, el sistema notifica que no hay movimientos para cerrar. El sistema permite registrar un cierre en cero.
Requerimientos especiales:	El comprobante generado debe poder imprimirse o exportarse a PDF.
Poscondiciones	El período queda cerrado, los ingresos del día quedan contabilizados por método de pago y no se pueden agregar nuevas transacciones al período cerrado.

Fuente: Elaboración propia

Tabla 11
Cierre de caja 2

Número del caso de uso:	CU-CIE-02
Fecha de elaboración:	25/04/2026
Descripción del caso de uso:	El cajero registra cada transacción de ingreso indicando el monto y el método de pago que se utiliza. El sistema acumula los valores separados por categoría de pago para facilitar el cierre.

Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Cajero, sistema
Precondiciones:	El usuario debe haber iniciado sesión con perfil de cajero. No debe existir un cierre de caja para el período actual. Debe existir al menos una factura emitida vinculada con el ingreso.
Flujo básico del caso de uso	
	1. El cajero selecciona la opción <i>Registrar ingreso por pago</i>. 2. El sistema muestra el formulario con los campos: monto, método de pago, factura. 3. El cajero selecciona el método de pago (efectivo y tarjeta). 4. El cajero ingresa el monto y la referencia de la factura correspondiente. 5. El sistema valida que la factura exista. 6. El sistema guarda el ingreso y lo acumula en el total del método de pago seleccionado. 7. El sistema confirma el registro y actualiza el resumen del día.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Pago con diferentes métodos	El sistema permite ingresar montos por cada método de pago. El sistema valida que la suma sea igual al total de la factura.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Factura no encontrada	Si la factura no existe, el sistema muestra un mensaje de error. El cajero debe verificar el número de factura e intentarlo nuevamente.

Requerimientos especiales:	El sistema debe aceptar pagos en efectivo, tarjeta. Cada transacción debe quedar vinculada a una factura existente.
Poscondiciones	El ingreso queda registrado y acumulado por método de pago, disponible para el proceso de cierre de caja del día.

Fuente: Elaboración propia

Tabla 12

Cierre de caja 3

Número del caso de uso:	CU-CIE-03
Fecha de elaboración:	25/04/2026
Descripción del caso de uso:	El administrador consulta los cierres de caja realizados en períodos anteriores. El sistema muestra el detalle de ingresos por método de pago y el total del día seleccionado.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión con perfil de administrador. Deben existir cierres de caja registrados en el sistema.
Flujo básico del caso de uso	
	1. El administrador accede al módulo de cierres de caja. 2. El sistema muestra la lista de cierres. 3. El administrador selecciona o filtra por rango de fechas. 4. El administrador selecciona un cierre específico. 5. El sistema muestra el detalle del cierre: total efectivo, total tarjeta y gran total. 6. El administrador puede ver el listado de transacciones incluidas en ese cierre.
Subflujos	

Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Exportar cierre	El administrador selecciona un cierre y el sistema genera un archivo PDF o Microsoft Excel con el detalle.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Sin cierres en el rango	Si no hay cierres en el rango de fechas seleccionado, el sistema muestra un mensaje informativo.
Requerimientos especiales:	La consulta debe permitir filtrar por fecha, cajero o monto total. El detalle de cada cierre debe poder exportarse en formato PDF o Microsoft Excel.
Poscondiciones	El administrador obtiene la información de cierres sin modificar ningún dato del sistema.

Fuente: Elaboración propia

Tabla 13

Calcular los gastos 1

Número del caso de uso:	CU-GAS-01
Fecha de elaboración:	26/04/2026
Descripción del caso de uso:	El administrador registra un gasto operativo de la empresa (servicios públicos, licencias, mantenimiento, entre otros). El sistema lo clasifica por categoría, valida la existencia de la categoría y lo acumula al total mensual.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema

Precondiciones:	El usuario debe haber iniciado sesión en el sistema. La categoría de gasto debe estar definida en el catálogo.
Flujo básico del caso de uso	
	1. El administrador selecciona la opción <i>Registrar gasto operativo</i>. 2. El sistema muestra el formulario con los campos: categoría, descripción, monto, fecha. 3. El administrador selecciona la categoría y completa los datos del gasto. 4. El sistema valida que la categoría exista en el catálogo. 5. El sistema verifica si el gasto supera el presupuesto asignado a esa categoría. 6. El sistema guarda el gasto y lo acumula al total mensual de la categoría. 7. El sistema confirma el registro y muestra el total actualizado del mes.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Exportar gasto	El administrador selecciona un gasto y el sistema genera un archivo PDF o Microsoft Excel.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Categoría no existe	Si la categoría no está definida, el sistema notifica al administrador. El flujo regresa al paso 2.
FA_02 Gasto supera presupuesto	Si el monto supera el presupuesto de la categoría, el sistema genera una alerta. El administrador decide si continúa el registro.

Requerimientos especiales:	El sistema debe acumular los gastos por categoría.
Poscondiciones	El gasto queda registrado, clasificado por categoría y acumulado en el total mensual de gastos de la empresa.

Fuente: Elaboración propia

Tabla 14

Calcular los gastos 2

Número del caso de uso:	CU-GAS-02
Fecha de elaboración:	25/04/2026
Descripción del caso de uso:	El administrador consulta el resumen de gastos agrupados por categoría. El sistema muestra subtotales por categoría y el total general del mes.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión en el sistema. Deben existir gastos registrados en el período consultado.
Flujo básico del caso de uso	
	1. El administrador accede al módulo de gastos y selecciona <i>Ver resumen mensual</i>. 2. El sistema calcula y muestra los gastos agrupados por categoría. 3. El sistema muestra el subtotal de cada categoría y el total general del mes. 4. El administrador puede seleccionar un mes o rango de fechas diferente. 5. El sistema actualiza el resumen según el filtro aplicado.
Subflujos	

Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Ver gasto individual	El administrador selecciona un gasto específico del listado. El sistema muestra el detalle completo: fecha, descripción, monto.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Sin gastos en el período	Si no hay gastos registrados en el período seleccionado, el sistema muestra los totales en cero. El sistema sugiere verificar el rango de fechas seleccionado.
Requerimientos especiales:	El resumen debe poder exportarse a Microsoft Excel o PDF.
Poscondiciones	El administrador obtiene el resumen de gastos por categoría

Fuente: Elaboración propia

Tabla 15

Calcular los gastos 3

Número del caso de uso:	CU-GAS-03
Fecha de elaboración:	25/04/2026
Descripción del caso de uso:	El administrador crea, edita o elimina categorías de gasto del catálogo. El sistema valida que no existan categorías duplicadas antes de guardar cualquier cambio.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema

Precondiciones:	El usuario debe haber iniciado sesión con perfil de administrador. El módulo de gestión de gastos debe estar activo.
Flujo básico del caso de uso	
	1. El administrador accede a la opción <i>Gestionar categorías de gasto</i>. 2. El sistema muestra el listado de categorías existentes. 3. El administrador selecciona la acción por realizar: crear, editar o eliminar. 4. Para crear: ingresa el nombre y descripción de la nueva categoría. 5. El sistema valida que no exista otra categoría con el mismo nombre. 6. El sistema guarda el cambio y actualiza el catálogo de categorías.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Asignar presupuesto a categoría	Al crear o editar una categoría, el administrador puede asignar un monto de presupuesto mensual. El sistema usa ese monto para validar alertas al registrar gastos de esa categoría.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Categoría duplicada	Si el nombre ya existe en el catálogo, el sistema muestra un mensaje de error. El administrador debe ingresar un nombre diferente para continuar.
FA_02 Categoría con gastos asociados	Si se intenta eliminar una categoría que tiene gastos registrados, el sistema bloquea la eliminación.

	El sistema sugiere desactivar la categoría en lugar de eliminarla.
Requerimientos especiales:	No se deben permitir categorías con nombres duplicados. Las categorías con gastos asociados no pueden eliminarse, solo desactivarse.
Poscondiciones	El catálogo de categorías queda actualizado y disponible para usarse en el registro de nuevos gastos.

Fuente: Elaboración propia

Tabla 16

Gestionar facturación 1

Número del caso de uso:	CU-FAC-01
Fecha de elaboración:	26/04/2026
Descripción del caso de uso:	El cajero genera una factura electrónica vinculando al cliente, el servicio o producto y el monto. El sistema calcula automáticamente los totales e impuestos y emite la factura.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Cajero, sistema
Precondiciones:	El usuario debe haber iniciado sesión con perfil de cajero. El cliente debe estar registrado en el sistema. Los productos o servicios deben estar definidos en el catálogo.
Flujo básico del caso de uso	
	1. El cajero selecciona la opción <i>Crear nueva factura</i>. 2. El sistema muestra el formulario de factura con los campos de cliente y líneas de detalle. 3. El cajero busca y selecciona el cliente al que se emitirá la factura. 4. El cajero agrega los productos o servicios a la factura. 5. El sistema calcula automáticamente el subtotal, impuesto y total de cada línea y el total general.

	6. El cajero selecciona el método de pago. 7. El sistema registra la factura como emitida. 8. El sistema muestra la factura.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Aplicar descuento	El cajero selecciona la opción <i>Aplicar descuento</i> . El sistema permite ingresar un porcentaje o monto fijo de descuento. El sistema recalcula el total con el descuento aplicado.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Cliente no encontrado	Si el cliente no está registrado, el sistema ofrece la opción de registrarlo en el momento. Una vez registrado, el flujo regresa al paso 3.
Requerimientos especiales:	El cálculo de impuestos debe aplicarse según el IVA.
Poscondiciones	La factura queda emitida y vinculada con el cliente.

Fuente: Elaboración propia

Tabla 17

Gestionar facturación 2

Número del caso de uso:	CU-FAC-02
Fecha de elaboración:	26/04/2026
Descripción del caso de uso:	El administrador busca facturas por cliente. El sistema muestra el estado de cada factura (pagada o pendiente), el monto y el detalle completo.
Autor del caso de uso:	Tyron López Chinchilla

Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión en el sistema. Deben existir facturas registradas en el sistema.
Flujo básico del caso de uso	
	1. El administrador accede al módulo de facturación y selecciona <i>Consultar historial</i>. 2. El sistema muestra el listado de facturas ordenadas por fecha descendente. 3. El administrador aplica filtros por cliente o estado. 4. El administrador selecciona una factura. 5. El sistema muestra el detalle completo: cliente, producto, subtotal, impuesto, total y estado.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Reimprimir factura	El administrador selecciona la opción <i>Reimprimir</i>. El sistema genera nuevamente el comprobante de la factura que se seleccionó. El administrador puede imprimirlo o descargarlo en PDF.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Sin resultados en la búsqueda	Si ninguna factura coincide con los filtros aplicados, el sistema muestra un mensaje informativo. El administrador puede limpiar los filtros y realizar una nueva búsqueda.
Requerimientos especiales:	El historial debe permitir filtrar por múltiples criterios simultáneamente. Debe mostrar el estado de cada factura
Poscondiciones	El administrador obtiene la información de las facturas consultadas sin modificar ningún dato del sistema.

Fuente: Elaboración propia

Tabla 18
Gestionar facturación 3

Número del caso de uso:	CU-FAC-03
Fecha de elaboración:	26/04/2026
Descripción del caso de uso:	El administrador solicita la anulación de una factura emitida por error. El sistema registra el motivo, cambia el estado de la factura a anulada.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión con perfil de administrador. La factura debe existir y estar en estado emitida o pagada. La factura no debe tener más de 30 días de emitida.
Flujo básico del caso de uso	
	1. El administrador busca y selecciona la factura por anular. 2. El sistema muestra el detalle de la factura y la opción <i>Anular factura</i>. 3. El sistema solicita confirmación de la acción. 4. El administrador confirma la anulación. 5. El sistema cambia el estado de la factura a <i>Anulada</i>. 6. El sistema revierte el impacto de la factura en los totales de ingresos del período. 7. El sistema notifica al administrador que la anulación se procesó exitosamente.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Ver motivo de anulación	El administrador selecciona una factura anulada y el sistema muestra el motivo y la fecha en la que fue anulada.
Flujos alternos	

Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Factura fuera del plazo	Si la factura supera los 30 días de emitida, el sistema bloquea la anulación.
Requerimientos especiales:	Las facturas anuladas no deben eliminarse del sistema, solo cambiar su estado.
Poscondiciones	La factura queda con estado <i>Anulada</i> , los totales de ingresos del período se ajustan.

Fuente: Elaboración propia

Tabla 19

Gestión de cuentas por pagar 1

Número del caso de uso:	CU-CXP-01
Fecha de elaboración:	26/04/2026
Descripción del caso de uso:	El administrador registra una obligación pendiente de pago indicando el proveedor, el monto y la fecha de vencimiento. El sistema almacena el registro y lo incluye en la agenda de pagos pendientes.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión con perfil de administrador. El proveedor debe estar registrado en el sistema.
Flujo básico del caso de uso	
	1. El administrador selecciona la opción <i>Registrar cuenta por pagar</i>. 2. El sistema muestra el formulario con los campos: proveedor, descripción, monto, fecha de emisión y fecha de vencimiento.

	3. El administrador completa todos los campos requeridos. 4. El sistema valida que la fecha de vencimiento sea después de la fecha actual. 5. El sistema calcula automáticamente los días restantes hasta el vencimiento. 6. El sistema guarda la cuenta y la incluye en el listado de cuentas pendientes. 7. El sistema confirma el registro y muestra el saldo total de cuentas por pagar actualizado.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Ver detalle de cuenta	El administrador selecciona una cuenta por pagar del listado y el sistema muestra el detalle completo: proveedor, monto, fecha de vencimiento y días restantes.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Proveedor no registrado	Si el proveedor no existe en el sistema, el sistema ofrece la opción de registrarlo. Una vez registrado el proveedor, el flujo regresa al paso 2.
FA_02 Fecha de vencimiento inválida	Si la fecha de vencimiento es anterior a la actual, el sistema muestra un error. El administrador debe corregir la fecha antes de guardar.
Requerimientos especiales:	El sistema debe calcular automáticamente los días restantes al vencimiento.
Poscondiciones	La cuenta por pagar queda registrada en el sistema y disponible en la agenda de pagos pendientes ordenada por fecha de vencimiento.

Fuente: Elaboración propia

Tabla 20*Gestión de cuentas por pagar 2*

Número del caso de uso:	CU-CXP-02
Fecha de elaboración:	26/04/2026
Descripción del caso de uso:	El administrador registra un pago parcial o total a una cuenta pendiente. El sistema actualiza el saldo adeudado y marca la cuenta como cancelada si el saldo llega a cero.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión en el sistema. La cuenta por pagar debe existir y tener saldo pendiente mayor que cero.
Flujo básico del caso de uso	
	1. El administrador busca y selecciona la cuenta por pagar a abonar. 2.El sistema muestra el detalle de la cuenta: proveedor, monto original, abonos anteriores y saldo pendiente. 3.El administrador ingresa el monto del abono o pago y el método de pago que se utiliza. 4.El sistema valida que el monto ingresado no sea mayor al saldo pendiente. 5.El sistema registra el abono y recalcula el saldo pendiente. 6.Si el saldo llega a cero, el sistema cambia automáticamente el estado de la cuenta a <i>Cancelada</i>. 7.El sistema muestra la confirmación del pago y el saldo actualizado.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Generar comprobante de pago	El administrador selecciona <i>Generar comprobante</i> . El sistema genera un comprobante del abono realizado con fecha, monto y saldo restante.

	El comprobante puede imprimirse o exportarse a PDF.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Monto mayor al saldo	Si el monto del abono supera el saldo pendiente, el sistema muestra un error. El administrador debe corregir el monto para que no exceda el saldo adeudado.
Requerimientos especiales:	El cambio de estado a cancelada debe realizarse de forma automática.
Poscondiciones	El abono queda registrado, el saldo pendiente se actualiza y si la deuda queda saldada la cuenta cambia automáticamente a estado <i>Cancelada</i> .

Fuente: Elaboración propia

Tabla 21

Gestión de cuentas por pagar 3

Número del caso de uso:	CU-CXP-03
Fecha de elaboración:	26/04/2026
Descripción del caso de uso:	El administrador consulta las cuentas por pagar con fecha de vencimiento próxima. El sistema muestra la lista ordenada por el monto pendiente de cada cuenta.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión en el sistema. Deben existir cuentas por pagar pendientes en el sistema.
Flujo básico del caso de uso	

	1. El administrador accede al módulo de cuentas por pagar y selecciona <i>Ver próximas a vencer</i>. 2.El sistema calcula los días restantes de cada cuenta activa. 3.El sistema muestra el listado ordenado de menor a mayor (días restantes). 4.El administrador selecciona una cuenta para ver su detalle completo. 5.El administrador puede iniciar el proceso de pago desde esta pantalla.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Filtrar por proveedor	El administrador aplica el filtro por proveedor específico. El sistema muestra solo las cuentas pendientes del proveedor seleccionado ordenadas por vencimiento.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Sin cuentas próximas a vencer	Si no hay cuentas con vencimiento en los próximos 30 días, el sistema lo notifica. El sistema muestra de todas formas el listado completo de cuentas pendientes.
Requerimientos especiales:	La consulta debe actualizarse automáticamente cada vez que se accede al módulo.
Poscondiciones	El administrador obtiene la información de cuentas próximas a vencer sin modificar ningún dato del sistema.

Fuente: Elaboración propia

Tabla 22

Gestión del presupuesto 1

Número del caso de uso:	CU-PRE-01
--------------------------------	-----------

Fecha de elaboración:	27/04/2026
Descripción del caso de uso:	El administrador define el período y el sistema genera una proyección de presupuesto basada en el historial de ingresos y gastos, esto muestra el estimado de saldo disponible al cierre del período.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión con perfil de administrador. Deben existir datos históricos de ingresos y gastos de al menos un período anterior.
Flujo básico del caso de uso	
	1. El administrador accede al módulo de presupuesto y selecciona <i>Generar proyección</i>. 2. El sistema solicita el rango de fechas del período por proyectar. 3. El administrador ingresa las fechas de inicio y fin del período. 4. El sistema analiza el historial de ingresos y gastos de los períodos anteriores. 5. El sistema calcula los promedios por categoría y genera la proyección estimada. 6. El sistema muestra la proyección con ingresos estimados, gastos estimados y saldo proyectado. 7. El administrador puede ajustar manualmente los valores proyectados antes de confirmar. 8. El administrador confirma el presupuesto y el sistema lo guarda como presupuesto activo del período.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Ajuste manual de proyección	El administrador selecciona una categoría y modifica el valor proyectado. El sistema recalcula el saldo estimado con el valor ajustado.

Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Historial insuficiente	Si no hay suficiente historial para proyectar, el sistema notifica al administrador. El sistema permite ingresar el presupuesto de forma manual sin proyección automática.
Requerimientos especiales:	La proyección debe basarse en el promedio de al menos los últimos 3 meses. El administrador debe ajustar los valores proyectados antes de confirmar.
Poscondiciones	El presupuesto queda definido para el período y disponible para compararse contra los valores reales conforme avance el período.

Fuente: Elaboración propia

Tabla 23

Gestión del presupuesto 2

Número del caso de uso:	CU-PRE-02
Fecha de elaboración:	27/04/2026
Descripción del caso de uso:	El administrador compara lo presupuestado con los valores del período activo. El sistema muestra las variaciones por categoría e indica si hay sobrepaso o superávit.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión en el sistema. Debe existir un presupuesto activo para el período consultado. Deben existir ingresos y gastos registrados en ese período.
Flujo básico del caso de uso	

	1. El administrador accede al módulo de presupuesto y selecciona <i>Comparar</i>. 2.El sistema selecciona automáticamente el período activo. 3.El sistema obtiene los valores presupuestados y los valores reales registrados. 4.El sistema calcula la variación por categoría (real menos presupuestado). 5.El sistema muestra la tabla comparativa con categoría, valor presupuestado, valor real, variación en monto y variación en porcentaje. 6.El sistema muestra el resultado global del período: déficit o superávit total.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Ver detalle por categoría	El administrador selecciona una categoría específica. El sistema muestra el desglose de todos los movimientos reales que conforman el total de esa categoría. El administrador puede identificar cuáles movimientos causaron el sobrepaso.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Sin datos reales en el período	Si no hay movimientos reales registrados en el período, el sistema muestra todos los valores reales en cero. El sistema indica que aún no hay datos para comparar.
Requerimientos especiales:	La comparación debe actualizarse automáticamente al ingresar nuevos movimientos. El resultado debe poder exportarse a Microsoft Excel o PDF.
Poscondiciones	El administrador obtiene el análisis comparativo del presupuesto sin modificar ningún dato del sistema.

Fuente: Elaboración propia

Tabla 24*Gestión del presupuesto 3*

Número del caso de uso:	CU-PRE-03
Fecha de elaboración:	27/04/2026
Descripción del caso de uso:	El administrador establece el rango de fechas para el cálculo del presupuesto. El sistema valida que el período no choque con uno existente y guarda la configuración.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión con perfil de administrador. El módulo de presupuesto debe estar activo en el sistema.
Flujo básico del caso de uso	
	1. El administrador accede a la configuración de períodos de presupuesto. 2. El sistema muestra los períodos anteriores y el formulario para crear un nuevo período. 3. El administrador ingresa la fecha de inicio y fin del nuevo período. 4. El sistema valida que las fechas sean válidas y que el período no choque con otros existentes. 5. El administrador confirma la creación del período. 6. El sistema guarda el período y lo deja disponible para generar el presupuesto. 7. El sistema muestra la confirmación y el período creado en el listado.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Editar período existente	El administrador selecciona un período existente y elige <i>Editar</i> . El sistema permite modificar las fechas siempre que no haya movimientos reales registrados.

	El sistema valida nuevamente la ausencia de solapamiento antes de guardar.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Período con choques	Si las fechas chocan con un período existente, el sistema indica las fechas en conflicto. El administrador debe ajustar el rango de fechas para continuar.
FA_02 Fecha de inicio posterior al fin	Si la fecha de inicio es mayor o igual a la fecha de fin, el sistema muestra un error de validación. El administrador debe corregir las fechas ingresadas.
Requerimientos especiales:	No se deben permitir períodos con fechas iguales.
Poscondiciones	El período queda configurado en el sistema y disponible para asociar un presupuesto y registrar movimientos reales.

Fuente: Elaboración propia

Tabla 25

Gestión de ingresos I

Número del caso de uso:	CU-ING-01
Fecha de elaboración:	27/04/2026
Descripción del caso de uso:	El cajero registra un ingreso generado por una venta o servicio indicando el tipo de pago, el producto y el monto. El sistema lo clasifica automáticamente y actualiza el total de ingresos del mes.
Autor del caso de uso:	Equipo 2
Actores relacionados:	Cajero, sistema
Precondiciones:	El usuario debe haber iniciado sesión con perfil de cajero. No debe existir un cierre de caja para el período actual.

	La factura asociada al ingreso debe existir en el sistema.
Flujo básico del caso de uso	
	1. El cajero selecciona la opción <i>Registrar ingreso por venta</i>. 2. El sistema muestra el formulario con los campos: tipo de ingreso, producto o servicio, monto, tipo de pago y referencia de factura. 3. El cajero selecciona el tipo de ingreso (venta de producto, examen o reparación). 4. El cajero ingresa el monto y el método de pago. 5. El sistema guarda el ingreso y lo clasifica por tipo y método de pago. 6. El sistema actualiza automáticamente el total acumulado de ingresos del mes. 7. El sistema confirma el registro al cajero.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Ingreso sin factura previa	Si el cajero no tiene una factura previa selecciona <i>Crear factura y registrar ingreso</i>. El sistema redirige al módulo de facturación para emitir la factura. Una vez emitida, el ingreso queda automáticamente registrado.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Monto no coincide con factura	Si el monto ingresado difiere del total de la factura, el sistema muestra una alerta. El cajero debe verificar y corregir el monto antes de guardar.
Requerimientos especiales:	Cada ingreso debe estar vinculado obligatoriamente a una factura. El total mensual debe actualizarse en tiempo real con cada nuevo registro.
Poscondiciones	El ingreso queda registrado, clasificado por tipo y método de pago y el total acumulado del mes se actualiza automáticamente.

Fuente: Elaboración propia

Tabla 26
Gestión de ingresos 2

Número del caso de uso:	CU-ING-02
Fecha de elaboración:	27/04/2026
Descripción del caso de uso:	El administrador consulta los ingresos del período filtrados por método de pago (efectivo y tarjeta). El sistema muestra el subtotal de cada categoría y el total combinado.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión en el sistema. Deben existir ingresos registrados en el período consultado.
Flujo básico del caso de uso	
	1. El administrador accede al módulo de ingresos y selecciona <i>Consultar por tipo de pago</i>. 2. El sistema muestra el formulario de filtros con las opciones de método de pago y rango de fechas. 3. El administrador selecciona uno o varios métodos de pago y el período por consultar. 4. El sistema agrupa los ingresos por método de pago y calcula los subtotales. 5. El sistema muestra la tabla con método de pago, cantidad de transacciones, subtotal y porcentaje del total. 6. El sistema muestra el total general combinado de todos los métodos seleccionados.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Ver transacciones de un método de pago	El administrador selecciona un método de pago específico de la tabla. El sistema muestra el listado de todas las transacciones de ese método con fecha, monto y factura asociada.

Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Sin ingresos en el período	Si no hay ingresos registrados para los filtros seleccionados, el sistema muestra los totales en cero. El sistema sugiere ampliar el rango de fechas o cambiar los filtros.
Requerimientos especiales:	El filtro debe permitir seleccionar múltiples métodos de pago simultáneamente. Los resultados deben poder exportarse en formato Microsoft Excel o PDF.
Poscondiciones	El administrador obtiene el desglose de ingresos por método de pago sin modificar ningún dato del sistema.

Fuente: Elaboración propia

Tabla 27

Gestión de ingresos 3

Número del caso de uso:	CU-ING-03
Fecha de elaboración:	27/04/2026
Descripción del caso de uso:	El administrador visualiza un desglose de ingresos agrupado por producto o servicio vendido. El sistema muestra las unidades y el monto total generado por cada producto en el período.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión en el sistema. Deben existir ingresos con tipo de producto registrados en el período.
Flujo básico del caso de uso	

	1. El administrador accede al módulo de ingresos y selecciona <i>Consultar por producto</i>. 2. El sistema muestra el formulario de filtros por período y categoría de producto. 3. El administrador selecciona el rango de fechas y confirma la consulta. 4. El sistema agrupa los ingresos por producto o servicio y calcula los totales. 5. El sistema muestra la tabla con producto, unidades vendidas, monto total y porcentaje sobre el total de ingresos. 6. El sistema ordena los productos de mayor a menor ingreso generado. 7. El administrador puede identificar los productos más rentables del período.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Ver detalle de ventas de un producto	El administrador selecciona un producto específico. El sistema muestra cada venta individual del producto: fecha, cantidad, precio unitario y total.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Sin ventas de productos en el período	Si no hay ventas registradas en el período, el sistema muestra la tabla vacía. El sistema indica que no se encontraron registros para los filtros aplicados.
Requerimientos especiales:	El desglose debe mostrar tanto el monto total como las unidades vendidas por producto.
Poscondiciones	El administrador obtiene el análisis de ingresos por producto sin modificar ningún dato del sistema.

Fuente: Elaboración propia

Tabla 28*Gestionar cuentas contables 1*

Número del caso de uso:	CU-CON-01
Fecha de elaboración:	27/04/2026
Descripción del caso de uso:	El administrador registra una nueva cuenta en el catálogo contable indicando código, nombre, tipo. El sistema valida que el código no esté duplicado y guarda la cuenta.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión con perfil de administrador. El código de la nueva cuenta no debe existir en el catálogo.
Flujo básico del caso de uso	
	1. El administrador accede al módulo de cuentas contables y selecciona <i>Crear nueva cuenta</i>. 2. El sistema muestra el formulario con los campos: código, nombre, tipo de cuenta. 3. El administrador ingresa el código único de la cuenta siguiendo el formato establecido. 4. El administrador selecciona el tipo de cuenta (ingreso, gasto, activo, pasivo y patrimonio). 5. El administrador ingresa el nombre descriptivo. 6. El sistema valida que el código no exista en el catálogo. 7. El sistema guarda la nueva cuenta y la agrega al catálogo contable. 8. El sistema confirma la creación y muestra la cuenta en el listado.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Ver cuentas por tipo	El administrador filtra el catálogo por tipo de cuenta y el sistema muestra solo las cuentas que corresponden al tipo seleccionado.

Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Código duplicado	Si el código ya existe en el catálogo, el sistema muestra un error de validación. El administrador debe ingresar un código diferente antes de continuar.
FA_02 Formato de código inválido	Si el código no cumple el formato requerido, el sistema muestra el formato correcto. El administrador debe corregir el código e intentar nuevamente.
Requerimientos especiales:	El código de cada cuenta debe ser único en todo el catálogo. El tipo de cuenta determina si los movimientos son débito o crédito.
Poscondiciones	La nueva cuenta queda registrada en el catálogo contable y disponible para recibir movimientos desde los demás módulos del sistema.

Fuente: Elaboración propia

Tabla 29

Gestionar cuentas contables 2

Número del caso de uso:	CU-CON-02
Fecha de elaboración:	27/04/2026
Descripción del caso de uso:	El administrador consulta el catálogo completo de cuentas contables con su código, nombre, tipo y balance actualizado. Puede filtrar por tipo de cuenta o buscar por nombre o código.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión en el sistema. Deben existir cuentas registradas en el catálogo contable.

Flujo básico del caso de uso	
	1. El administrador accede al módulo de cuentas contables y selecciona <i>Ver catálogo</i>. 2. El sistema muestra el listado completo de cuentas con código, nombre, tipo y balance actual. 3. El administrador puede filtrar por tipo de cuenta (ingresos, gastos, activos y pasivos). 4. El administrador puede buscar una cuenta específica por nombre o código. 5. El sistema retorna las cuentas que coincidan con el filtro o búsqueda aplicados. 6. El administrador selecciona una cuenta para ver su detalle y el historial de movimientos asociados.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Ver historial de movimientos de una cuenta	El administrador selecciona la opción <i>Ver movimientos</i> de una cuenta. El sistema muestra todos los movimientos que han afectado esa cuenta con fecha, tipo, monto y módulo origen. El administrador puede filtrar por rango de fechas.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Sin cuentas que coincidan con el filtro	Si ninguna cuenta coincide con el criterio de búsqueda, el sistema muestra un mensaje informativo. El administrador puede limpiar los filtros y ver el catálogo completo.
Requerimientos especiales:	El balance de cada cuenta debe reflejar el valor actualizado con todos los movimientos registrados. El catálogo debe poder exportarse en formato Microsoft Excel o PDF.
Poscondiciones	El administrador obtiene la información del catálogo sin modificar ningún dato del sistema.

Fuente: Elaboración propia

Tabla 30*Gestionar cuentas contables 3*

Número del caso de uso:	CU-CON-03
Fecha de elaboración:	27/04/2026
Descripción del caso de uso:	El sistema actualiza automáticamente el balance de las cuentas contables al registrar ingresos, gastos u otros movimientos desde cualquier módulo. El administrador puede revisar el historial de movimientos de cada cuenta.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Sistema, administrador
Precondiciones:	Las cuentas contables deben estar configuradas en el catálogo. El movimiento que origina la actualización debe estar correctamente registrado en su módulo origen.
Flujo básico del caso de uso	
	1. Un movimiento se registra en cualquier módulo del sistema (ingreso, gasto, factura, etc.). 2. El sistema identifica las cuentas contables que deben afectarse según el tipo de movimiento. 3. El sistema determina si el movimiento es débito o crédito para cada cuenta afectada. 4. El sistema actualiza el balance de cada cuenta afectada de forma automática. 5. El sistema registra el movimiento en el historial de la cuenta con fecha, tipo, monto y módulo origen. 6. El administrador puede consultar el historial de movimientos de cualquier cuenta en cualquier momento.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	

SF_01 Revisión manual del balance	El administrador accede al detalle de una cuenta. El sistema muestra el balance calculado y el listado de movimientos que lo conforman. El administrador puede verificar que cada movimiento esté correctamente registrado.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Cuenta no configurada para el tipo de movimiento	Si el tipo de movimiento no tiene una cuenta contable configurada, el sistema registra el movimiento. El administrador debe configurar la cuenta contable correspondiente para futuros movimientos del mismo tipo.
Requerimientos especiales:	La actualización de balances debe ser automática y en tiempo real. Cada movimiento debe quedar trazado con el módulo y registro de origen.
Poscondiciones	Los balances de las cuentas afectadas quedan actualizados y el movimiento queda registrado en el historial contable del sistema.

Fuente: Elaboración propia

Tabla 31

Gestión de cuentas por cobrar 1

Número del caso de uso:	CU-CXC-01
Fecha de elaboración:	28/04/2026
Descripción del caso de uso:	El administrador genera una cuenta por cobrar al vincular al cliente con el servicio prestado y el monto pendiente. El sistema registra la deuda con fecha de inicio y condiciones de pago.
Autor del caso de uso:	Tyron López Chinchilla

Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión en el sistema. El cliente debe estar registrado en el sistema. Debe existir una factura emitida al cliente con saldo pendiente.
Flujo básico del caso de uso	
	1. El administrador selecciona la opción <i>Registrar cuenta por cobrar</i>. 2. El sistema muestra el formulario con los campos: cliente, factura asociada, monto total, fecha de inicio y condiciones de pago. 3. El administrador busca y selecciona el cliente. 4. El administrador asocia la cuenta a una factura emitida. 5. El sistema autocompleta el monto con el total de la factura que se seleccionó. 6. El administrador define las condiciones de pago (plazo, número de cuotas). 8. El sistema valida los datos y guarda la cuenta por cobrar con estado <i>Activa</i>. 9. El sistema confirma el registro y muestra el saldo pendiente inicial.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Cuenta por cobrar en cuotas	El administrador selecciona la modalidad de pago en cuotas. El sistema permite definir el número de cuotas y la fecha de cada una.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Cliente sin factura pendiente	Si el cliente seleccionado no tiene facturas pendientes, el sistema lo notifica.
Requerimientos especiales:	Cada cuenta por cobrar debe estar vinculada obligatoriamente a una factura.

	El sistema debe soportar cuentas con pago único y cuentas en cuotas.
Poscondiciones	La cuenta por cobrar queda registrada con estado <i>Activa</i> y el saldo pendiente igual al monto total de la factura vinculada.

Fuente: Elaboración propia

Tabla 32

Gestión de cuentas por cobrar 2

Número del caso de uso:	CU-CXC-02
Fecha de elaboración:	28/04/2026
Descripción del caso de uso:	El administrador registra el pago parcial o total de un cliente. El sistema actualiza el saldo pendiente, si el pago es completo, cierra la cuenta y cambia su estado.
Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión en el sistema. La cuenta por cobrar debe existir con estado <i>Activa</i> y saldo mayor que cero.
Flujo básico del caso de uso	
	1. El administrador busca y selecciona la cuenta por cobrar del cliente. 2. El sistema muestra el detalle: monto original, abonos anteriores, saldo pendiente y condiciones de pago. 3. El administrador ingresa el monto del abono y el método de pago. 4. El sistema valida que el monto del abono no sea mayor al saldo pendiente. 5. El sistema registra el abono en el historial de la cuenta. 6. El sistema recalcula el saldo pendiente descontando el abono. 7. Si el saldo llega a cero, el sistema cambia automáticamente el estado de la cuenta a <i>Cancelada</i>.

Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Ver historial de abonos	El administrador selecciona la cuenta y el sistema muestra el listado de todos los abonos realizados con fecha, monto y método de pago
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Abono mayor al saldo	Si el monto del abono supera el saldo pendiente, el sistema muestra un error. El administrador debe corregir el monto para que sea igual o menor al saldo adeudado.
Requerimientos especiales:	El cierre automático de la cuenta debe ocurrir en el mismo proceso de registro del último abono.
Poscondiciones	El abono queda registrado, el saldo se actualiza y si la cuenta queda saldada, el estado cambia automáticamente a <i>Cancelada</i> .

Fuente: Elaboración propia

Tabla 33

Gestión de cuentas por cobrar 3

Número del caso de uso:	CU-CXC-03
Fecha de elaboración:	28/04/2026
Descripción del caso de uso:	El administrador busca a un cliente y el sistema muestra todas sus cuentas activas con el monto original, los abonos realizados y el saldo pendiente actualizado.

Autor del caso de uso:	Tyron López Chinchilla
Actores relacionados:	Administrador, sistema
Precondiciones:	El usuario debe haber iniciado sesión en el sistema. El cliente debe estar registrado y tener al menos una cuenta por cobrar activa.
Flujo básico del caso de uso	
	1. El administrador accede al módulo de cuentas por cobrar y selecciona <i>Consultar saldo por cliente</i>. 2. El sistema muestra el campo de búsqueda de clientes. 3. El administrador ingresa el nombre o código del cliente. 4. El sistema retorna el listado de clientes que coincidan con la búsqueda. 5. El administrador selecciona el cliente. 6. El sistema muestra todas las cuentas activas del cliente con monto original, total abonado y saldo pendiente. 7. El administrador puede seleccionar una cuenta para ver el detalle de abonos. 8. El administrador puede iniciar un nuevo abono desde esta pantalla.
Subflujos	
Corresponde a las diferentes opciones (alternativas funcionales) que un actor tiene al iniciar con el flujo básico.	
SF_01 Ver total de la deuda del cliente	El sistema muestra el monto total que el cliente tiene pendiente sumando todas sus cuentas activas.
Flujos alternos	
Corresponde a lo que debe realizar el sistema ante posibles errores:	
FA_01 Cliente sin cuentas activas	Si el cliente no tiene cuentas activas, el sistema lo notifica.
Requerimientos especiales:	La consulta debe mostrar claramente el saldo pendiente actualizado.

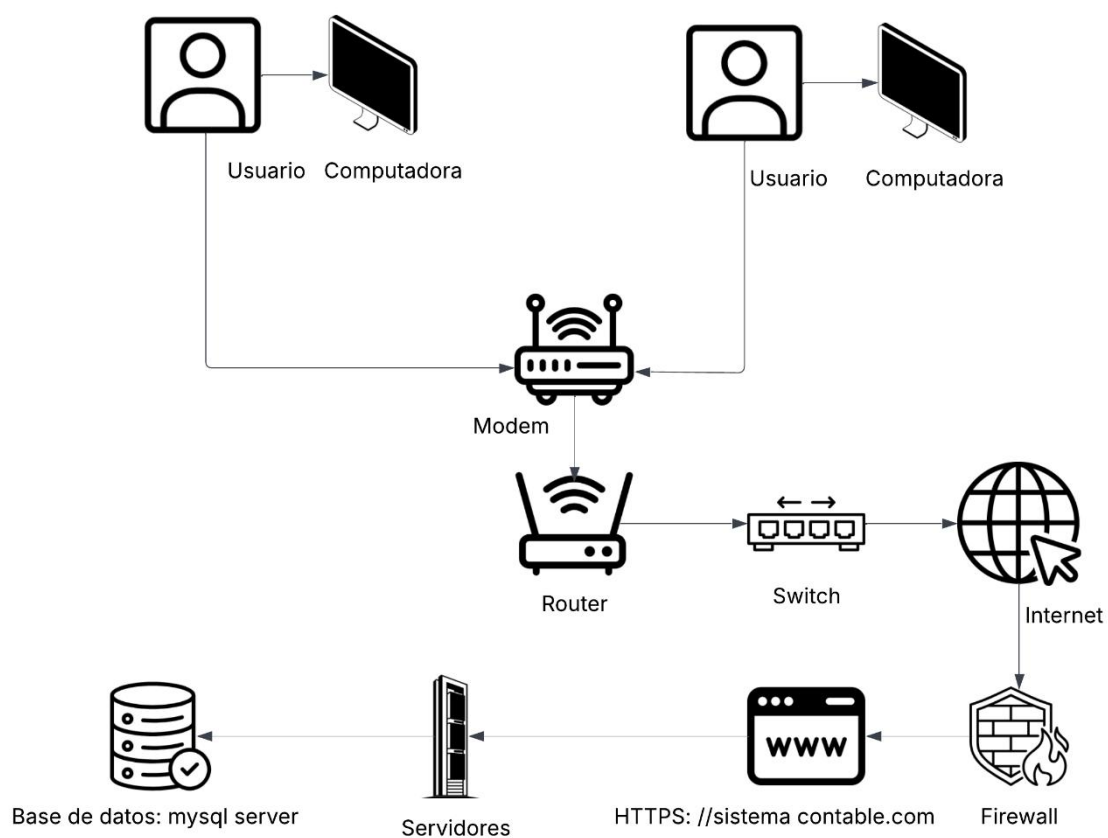
	Debe permitir identificar clientes con saldos vencidos o próximos a vencer.
Poscondiciones	El administrador obtiene el estado actualizado de las cuentas del cliente sin modificar ningún dato del sistema.

Fuente: Elaboración propia

Arquitectura del sistema

Figura 12

Arquitectura del sistema

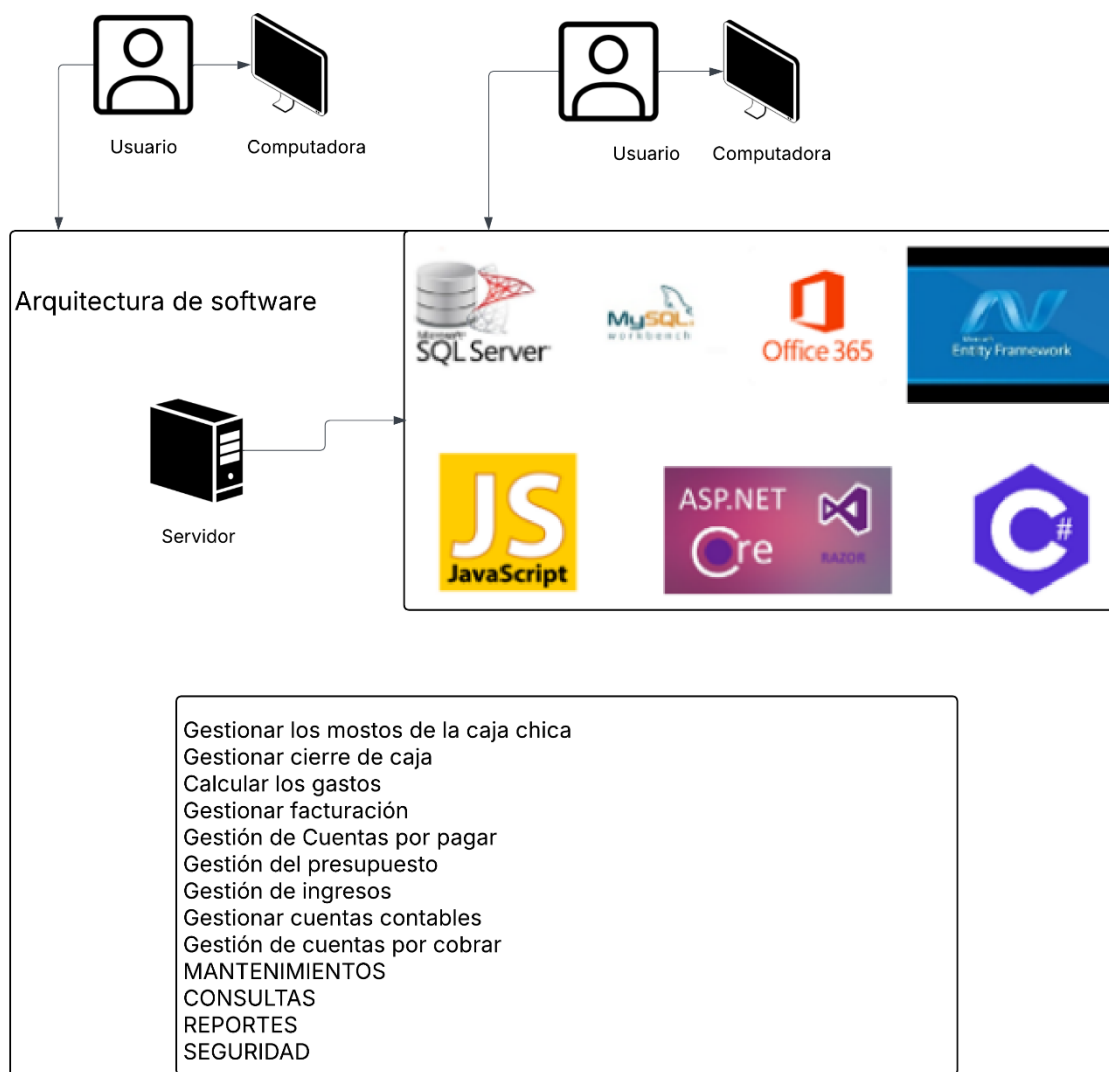


Fuente: Elaboración propia

Arquitectura del software

Figura 13

Arquitectura del software



Fuente: Elaboración propia

Diseño de entradas

Tabla 34

Catálogo contable agregar

Campo	Dato en BD	Dato a digitar	Restricción
Código	VARCHAR(25)	Texto	NOT NULL, UNIQUE, formato X-XX-XX
Nombre	VARCHAR(100)	Texto	NOT NULL
IdTipo_Cuenta	INT (FK)	Selección	Debe existir en Tipo_Cuenta_Contable
Naturaleza	VARCHAR(20)	Radio button	Deudora o Acreedora, NOT NULL
Balance_actual	DECIMAL(18,2)	Numérico	NOT NULL, DEFAULT 0.00

Fuente: Elaboración propia

Tabla 35

Catálogo contable modificar

Campo	Tipo BD	Dato a digitar	Restricción
Código	VARCHAR(25)	Texto	NOT NULL, UNIQUE, formato X-XX-XX
Nombre	VARCHAR(100)	Texto	NOT NULL
IdTipo_Cuenta	INT (FK)	Selección	Debe existir en Tipo_Cuenta_Contable
Naturaleza	VARCHAR(20)	Radio button	Deudora o Acreedora, NOT NULL
Balance_actual	DECIMAL(18,2)	Numérico	NOT NULL, DEFAULT 0.00
Activo	BIT	Radio button	Activo (1) o Inactivo (0)

Fuente: Elaboración propia

Tabla 36

Registro de nueva factura

Campo	Dato en BD	Dato a digitar	Restricción
Num_Factura	VARCHAR(50)	Texto	NOT NULL, UNIQUE
Fecha	DATETIME	Fecha/Hora	NOT NULL, DEFAULT GETDATE()
IdCliente	INT (FK)	Lista/Búsqueda	Debe existir en Cliente, NOT NULL
Descripción (línea)	VARCHAR(255)	Texto	NOT NULL por línea de detalle
Cantidad	INT	Numérico	NOT NULL, > 0
Precio_Unitario	DECIMAL(18,2)	Numérico	NOT NULL, >= 0
IdCuenta (línea)	INT (FK)	Lista	Debe existir en Catalogo_Cuenta
Monto_IVA	DECIMAL(18,2)	Numérico	NOT NULL, calculado automático
Subtotal_Linea	DECIMAL(18,2)	Numérico	NOT NULL, calculado automático
IdEstado_Factura	INT (FK)	Radio button	Emitida/Pagada/Anulada, NOT NULL
Subtotal	DECIMAL(18,2)	Numérico	NOT NULL, calculado automático
Impuesto_IVA	DECIMAL(18,2)	Numérico	NOT NULL, calculado automático

Campo	Dato en BD	Dato a digitar	Restricción
Total_Colones	DECIMAL(18,2)	Numérico	NOT NULL, calculado automático

Fuente: Elaboración propia

Tabla 37

Registro de modificar factura

Campo	Dato en BD	Dato a digitar	Restricción
Num_Factura	VARCHAR(50)	Texto	NOT NULL, UNIQUE (no editable)
Fecha	DATETIME	Fecha/Hora	NOT NULL
IdCliente	INT (FK)	Texto (lectura)	Nombre del cliente, NOT NULL
Descripción (línea)	VARCHAR(255)	Texto	NOT NULL por línea de detalle
Cantidad	INT	Numérico	NOT NULL, > 0
Precio_Unitario	DECIMAL(18,2)	Numérico	NOT NULL, >= 0
IdCuenta (línea)	INT (FK)	Lista	Debe existir en Catalogo_Cuenta
Monto_IVA	DECIMAL(18,2)	Numérico	NOT NULL, calculado automático
Subtotal_Linea	DECIMAL(18,2)	Numérico	NOT NULL, calculado automático
IdEstado_Factura	INT (FK)	Radio button	Emitida/Pagada/Anulada, NOT NULL
Subtotal	DECIMAL(18,2)	Numérico	NOT NULL, calculado automático

Campo	Dato en BD	Dato a digitar	Restricción
Impuesto_IVA	DECIMAL(18,2)	Numérico	NOT NULL, calculado automático
Total_Colones	DECIMAL(18,2)	Numérico	NOT NULL, calculado automático

Fuente: Elaboración propia

Tabla 38

Registro de agregar ingreso

Campo	Dato en BD	Dato a digitar	Restricción
Monto	DECIMAL(18,2)	Numérico	NOT NULL, > 0
IdCuenta	INT (FK)	Lista/Búsqueda	Debe existir en Catalogo_Cuenta, NOT NULL
IdMetodo_Pago	INT (FK)	Radio button	Efectivo o Tarjeta, NOT NULL
Descripcion	VARCHAR(MAX))	Texto largo	NOT NULL
Tipo_Transaccion	VARCHAR(20)	Oculto	Valor fijo: 'Ingreso'

Fuente: Elaboración propia

Tabla 39

Registro de modificar ingreso

Campo	Dato en BD	Dato a digitar	Restricción
Monto	DECIMAL(18,2)	Numérico	NOT NULL, > 0

Campo	Dato en BD	Dato a digitar	Restricción
IdCuenta	INT (FK)	Lista/Búsqueda	Debe existir en Catalogo_Cuenta, NOT NULL
IdMetodo_Pago	INT (FK)	Radio button	Efectivo o Tarjeta, NOT NULL
Descripcion	VARCHAR(MAX)	Texto largo	NOT NULL
Tipo_Transaccion	VARCHAR(20)	Oculto	Valor fijo: 'Ingreso'

Fuente: Elaboración propia

Tabla 40

Registro de agregar gasto

Campo	Dato en BD	Dato a digitar	Restricción
IdCuenta	INT (FK)	Lista/Búsqueda	Debe existir en Catalogo_Cuenta, NOT NULL
Monto	DECIMAL(18,2)	Numérico	NOT NULL, > 0
Descripcion	VARCHAR(MAX)	Texto largo	NOT NULL (Detalle del gasto)
Tipo_Transaccion	VARCHAR(20)	Oculto	Valor fijo: 'Egreso'

Fuente: Elaboración propia

Tabla 41*Registro de modificar gasto*

Campo	Dato en BD	Dato a digitar	Restricción
IdCuenta	INT (FK)	Lista/Búsqueda	Debe existir en Catalogo_Cuenta, NOT NULL
Monto	DECIMAL(18,2)	Numérico	NOT NULL, > 0
Descripcion	VARCHAR(MAX)	Texto largo	NOT NULL (Detalle del gasto)
Tipo_Transaccion	VARCHAR(20)	Oculto	Valor fijo: 'Egreso'

Fuente: Elaboración propia**Tabla 41***Registro de abono de cuentas por cobrar*

Campo	Dato en BD	Dato a digitar	Restricción
IdCliente	INT (FK)	Búsqueda/Texto	Buscar cliente, debe existir en Cliente
Saldo_Pendiente	DECIMAL(18,2)	Numérico (lectura)	Solo lectura, calculado del sistema
IdFactura	INT (FK)	Texto/Lista	Factura relacionada, debe existir en Factura
Monto a Abonar	DECIMAL(18,2)	Numérico	NOT NULL, > 0 y <= Saldo_Pendiente
IdMetodo_Pago	INT (FK)	Texto/Lista	Efectivo, debe existir en Metodo_Pago
Descripcion	VARCHAR(MAX)	Texto largo	Opcional (Notas/Observaciones)

Fuente: Elaboración propia

Tabla 42*Agregar cuentas por pagar*

Campo	Dato en BD	Dato a digitar	Restricción
IdProveedor	INT (FK)	Búsqueda/Lista	Debe existir en Proveedor, NOT NULL
Monto_Total	DECIMAL(18,2)	Numérico	NOT NULL, > 0 (Monto de la Deuda)
Fecha_Vencimiento	DATE	Fecha	NOT NULL
Descripcion	VARCHAR(255)	Texto	Referencia de factura/Concepto

Fuente: Elaboración propia**Tabla 43***Modificar cuentas por pagar*

Campo	Dato en BD	Dato a digitar	Restricción
IdProveedor	INT (FK)	Texto (lectura)	Nombre del proveedor, NOT NULL
Monto_Total	DECIMAL(18,2)	Numérico	NOT NULL, > 0 (Monto de la Deuda)
Fecha_Vencimiento	DATE	Fecha	NOT NULL
Descripcion	VARCHAR(255)	Texto	Referencia de factura/Concepto, opcional
Activo	BIT	Selección	Estado activo/inactivo de la cuenta

Fuente: Elaboración propia

Tabla 44*Agregar abonos a cuentas por pagar*

Campo	Dato en BD	Dato a digitar	Restricción
IdProveedor	INT (FK)	Búsqueda/Texto	Buscar proveedor, debe existir en Proveedor
Saldo_Pendiente	DECIMAL(18,2)	Númerico (lectura)	Solo lectura, calculado del sistema
IdFactura	INT (FK)	Texto/Lista	Factura relacionada, debe existir en Factura
Monto a Abonar	DECIMAL(18,2)	Númerico	NOT NULL, > 0 y <= Saldo_Pendiente
Descripcion	VARCHAR(MAX)	Texto largo	Opcional (Notas/Observaciones)

Fuente: Elaboración propia**Tabla 45***Agregar monto caja chica*

Campo	Dato en BD	Dato a digitar	Restricción
Saldo_Actual	DECIMAL(18,2)	Númerico	NOT NULL, Monto de Apertura >= 0
Monto_Limite	DECIMAL(18,2)	Númerico	NOT NULL, Monto Máximo de Gasto > 0
IdCedula_Responsable	INT (FK)	Lista	Usuario Responsable, debe existir en Usuario
Estado	VARCHAR(20)	Solo lectura	Valor fijo: 'Abierta'

Fuente: Elaboración propia

Tabla 46*Modificar monto Caja chica*

Campo	Dato en BD	Dato a digitar	Restricción
Saldo_Actual	DECIMAL(18,2)	Numérico	NOT NULL, Monto de Apertura ≥ 0
Monto_Limite	DECIMAL(18,2)	Numérico	NOT NULL, Monto Máximo de Gasto > 0
IdCedula_Responsable	INT (FK)	Lista	Usuario Responsable, debe existir en Usuario
Estado	VARCHAR(20)	Solo lectura	Valor fijo: 'Abierta'

Fuente: Elaboración propia**Tabla 47***Gestionar Cierre de caja*

Campo	Dato en BD	Dato a digitar	Restricción
Monto_Total_Sistema	DECIMAL(18,2)	Numérico (lectura)	Ingresos totales del día, calculado
Monto_Egresos	DECIMAL(18,2)	Numérico (lectura)	Egresos totales del día, calculado
Saldo_Esperado	DECIMAL(18,2)	Numérico (lectura)	Saldo esperado en la caja, calculado

Campo	Dato en BD	Dato a digitar	Restricción
Monto_Total_Fisico	DECIMAL(18,2)	Numérico	NOT NULL, efectivo físico contado
Monto_Diferencia	DECIMAL(18,2)	Numérico (lectura)	Diferencia calculada automático
Descripcion	VARCHAR(MAX))	Texto largo	Justificación/Notas, opcional

Fuente: Elaboración propia

Tabla 48

Modificar Cierre de caja

Campo	Dato en BD	Dato a digitar	Restricción
Monto_Total_Sistema	DECIMAL(18,2)	Numérico (lectura)	Ingresos totales, solo lectura
Monto_Egresos	DECIMAL(18,2)	Numérico (lectura)	Egresos totales, solo lectura
Saldo_Esperado	DECIMAL(18,2)	Numérico (lectura)	Saldo esperado, solo lectura
Monto_Total_Fisico	DECIMAL(18,2)	Numérico	Editable solo por Admin, NOT NULL
Descripcion	VARCHAR(MAX)	Texto largo	Justificación/Notas de modificación, NOT NULL

Fuente: Elaboración propia

Tabla 49*Gestionar presupuesto*

Campo	Dato en BD	Dato a digitar	Restricción
Mes (Nombre_Periodo)	VARCHAR(100)	Lista/Texto	NOT NULL, ej.: enero, febrero
Año	INT	Numérico	NOT NULL, año del período
IdCuenta	INT (FK)	Lista/Búsqueda	Cuenta Contable, debe existir en Catalogo_Cuenta
Monto_Presupuestado	DECIMAL(18,2)	Numérico	NOT NULL, Monto Proyectado ≥ 0

Fuente: Elaboración propia**Tabla 50***Editar presupuesto*

Campo	Dato en BD	Dato a digitar	Restricción
Mes (Nombre_Periodo)	VARCHAR(100)	Texto (lectura)	NOT NULL, no editable
Año	INT	Numérico (lectura)	NOT NULL, no editable
IdCuenta	INT (FK)	Texto (lectura)	Cuenta Contable, no editable
Monto_Presupuestado	DECIMAL(18,2)	Numérico	NOT NULL, Monto Proyectado a modificar ≥ 0

Fuente: Elaboración propia

Diseño físico de entradas

Figura 14

Diseño interfaz catalogo contable

+ Nueva Cuenta Contable ✕

CÓDIGO

Ej: 1.1.01

Solo números y puntos. Ej: 1.1.01

NOMBRE

Nombre de la cuenta

Solo letras y espacios.

TIPO DE CUENTA

Seleccione... ▼

NATURALEZA

☐ Deudora ☐ Acreedora

¿EN QUÉ MÓDULOS APLICA ESTA CUENTA?

☐ Caja Chica

☐ Cuentas por Pagar (origen del dinero)

☐ Cuentas por Cobrar (destino del dinero)

Cancelar **Guardar**

Fuente: Elaboración propia

Figura 15*Diseño interfaz catalogo contable editar*

Editar Cuenta Contable

CÓDIGO
1.1.0
Solo números y puntos. Ej: 1.1.01

NOMBRE
papel higiénico
Solo letras y espacios.

TIPO DE CUENTA
Gasto caja chica

NATURALEZA
☒ Deudora ☐ Acreedora

¿EN QUÉ MÓDULOS APLICA ESTA CUENTA?

☒ Caja Chica

☐ Cuentas por Pagar

☐ Cuentas por Cobrar

TIPO DE MOVIMIENTO POR DEFECTO
Gasto Caja Chica
Se auto-seleccionará al usar esta cuenta en Caja Chica.

Fuente: Elaboración propia

Figura 16
Diseño interfaz factura

CLIENTE Y MODALIDAD DE PAGO

CLIENTE * ▼

MÉTODO DE PAGO * ▼

MODALIDAD DE PAGO

☐ Pago de Contado
La factura se marca como "Pagada". No genera CXC.

DETALLE DE LA FACTURA Agregar producto/servicio

#	CÓDIGO	DESCRIPCIÓN	CANTIDAD	PRECIO UNIT.	IVA 13%	TOTAL
 Aún no has agregado productos. Haz clic en "Agregar producto/servicio".						

Subtotal (sin IVA): ₡ 0.00

IVA (13%): ₡ 0.00

TOTAL A PAGAR: ₡ 0.00

Cancelar Crear Factura

Fuente: Elaboración propia

Figura 17
Diseño interfaz factura cuotas

CLIENTE Y MODALIDAD DE PAGO

CLIENTE * ▼

MÉTODO DE PAGO * ▼

MODALIDAD DE PAGO

☒ Pago a Crédito
Se genera una Cuenta por Cobrar con cuotas mensuales.

¿EN CUÁNTOS MESES?

1 mes - Pago único

2 meses - 2 cuotas

3 meses - 3 cuotas

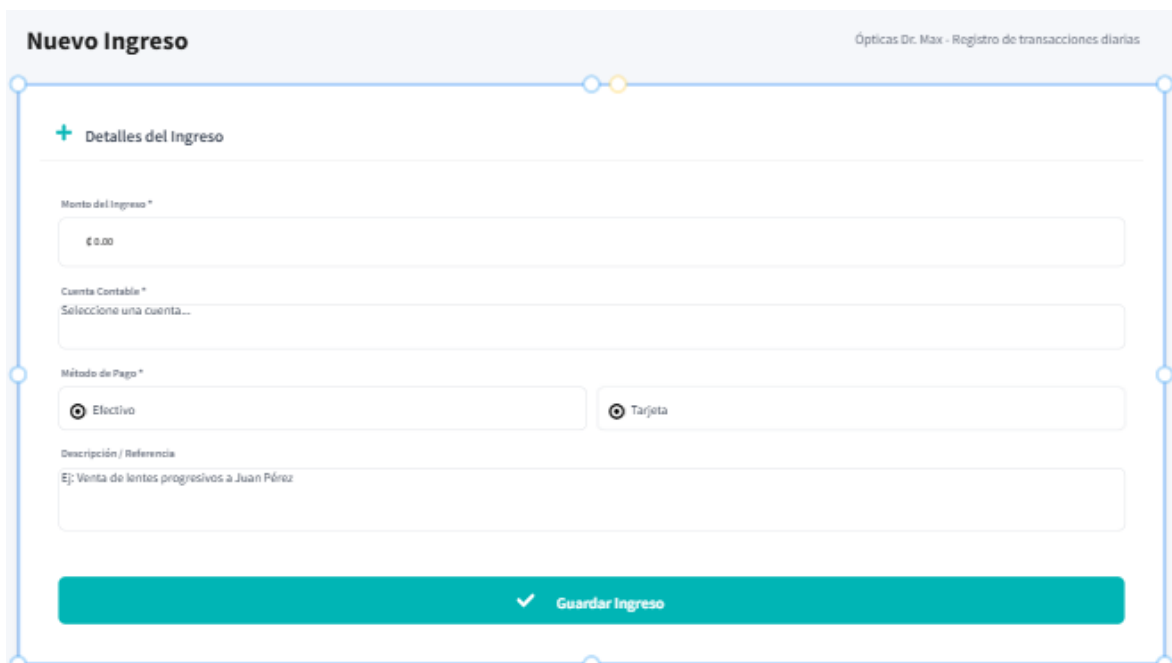
● Máximo permitido por la empresa: 3 meses. Las cuotas se generan automáticamente cada mes desde hoy.

Plan de cuotas

— agregó líneas para ver el plan —

Fuente: Elaboración propia

Figura 18
Diseño interfaz ingreso



Nuevo Ingreso Ópticas Dr. Max - Registro de transacciones diarias

+ Detalles del Ingreso

Monto del Ingreso *

₡ 0.00

Cuenta Contable *

Seleccione una cuenta...

Método de Pago *

☒ Electivo ☐ Tarjeta

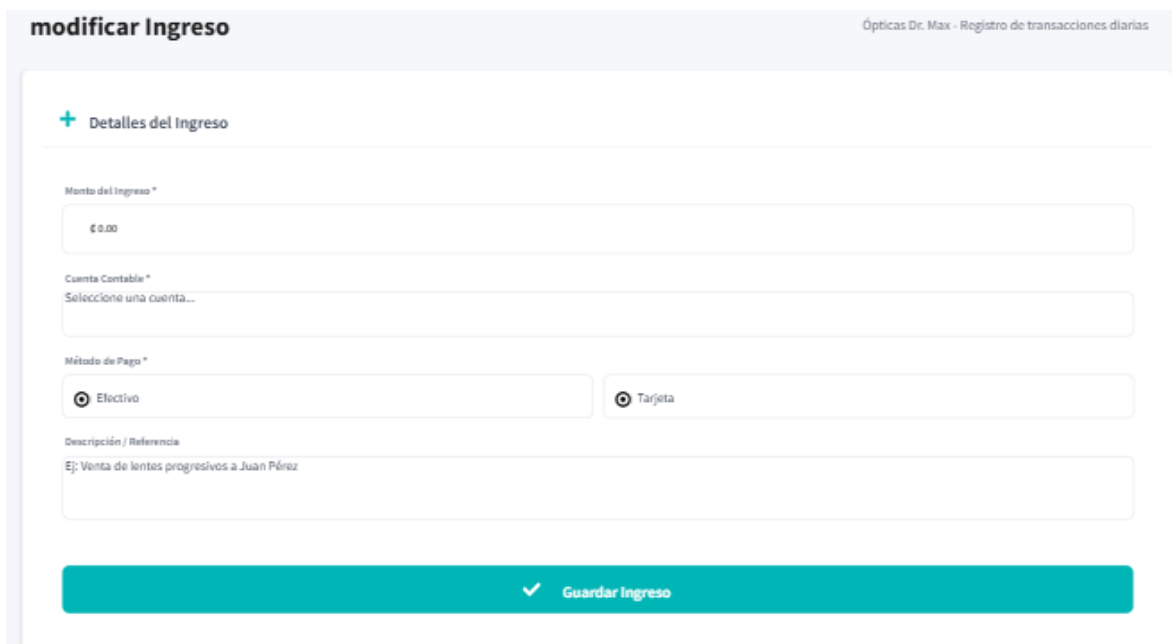
Descripción / Referencia

Ej: Venta de lentes progresivos a Juan Pérez

✓ Guardar Ingreso

Fuente: Elaboración propia

Figura 19
Diseño interfaz ingreso 2



modificar Ingreso Ópticas Dr. Max - Registro de transacciones diarias

+ Detalles del Ingreso

Monto del Ingreso *

₡ 0.00

Cuenta Contable *

Seleccione una cuenta...

Método de Pago *

☒ Electivo ☐ Tarjeta

Descripción / Referencia

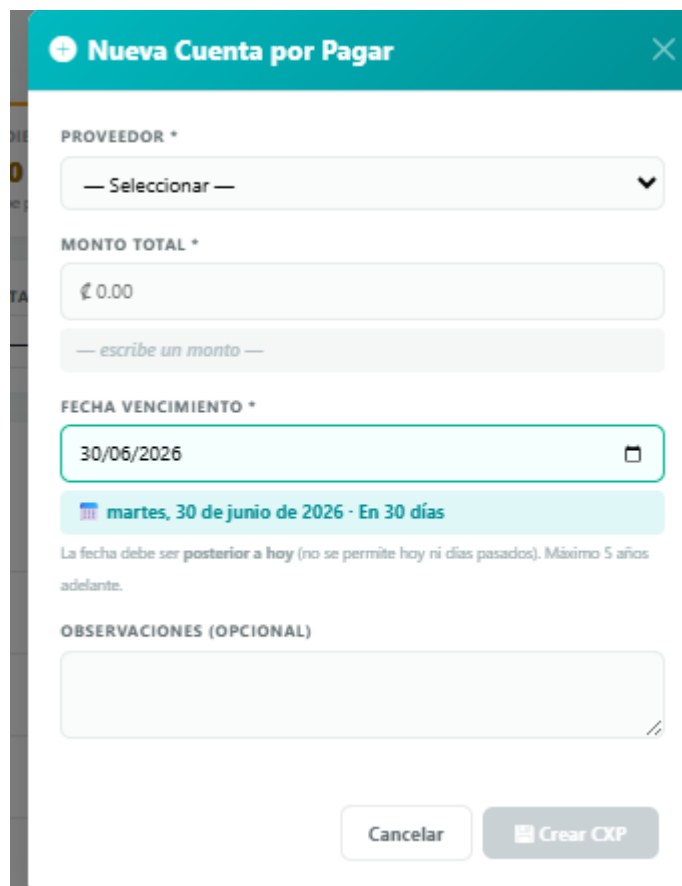
Ej: Venta de lentes progresivos a Juan Pérez

✓ Guardar Ingreso

Fuente: Elaboración propia

Figura 20*Diseño interfaz gastos*
*Fuente: Elaboración propia***Figura 21***Diseño interfaz gastos nueva categoría*
Fuente: Elaboración propia

Fuente: Elaboración propia

Figura 23*Diseño interfaz cuentas por pagar*

El formulario 'Nueva Cuenta por Pagar' presenta los siguientes campos y elementos:

- PROVEEDOR ***: Un menú desplegable con la opción '— Seleccionar —' y un ícono de flecha hacia abajo.
- MONTO TOTAL ***: Un campo de texto que muestra '₡ 0.00' y un placeholder '— escribe un monto —'.
- FECHA VENCIMIENTO ***: Un campo de fecha que muestra '30/06/2026' con un ícono de calendario. Debajo, una barra de información indica 'martes, 30 de junio de 2026 · En 30 días'.
- Observación de fecha**: Un texto informativo que dice: 'La fecha debe ser posterior a hoy (no se permite hoy ni días pasados). Máximo 5 años adelante.'
- OBSERVACIONES (OPCIONAL)**: Un área de texto grande para agregar comentarios.
- Botones de acción**: 'Cancelar' y 'Crear CXP' (con un ícono de documento).

Fuente: Elaboración propia

Figura 24

Diseño interfaz cuentas por pagar abono

Pagar a Proveedor · CxP #7

Volver al detalle

Estás registrando un egreso: sale dinero de tu caja/banco hacia el proveedor. El balance de la cuenta destino se va a reducir por el monto del abono.

Datos del Pago

MONTO DEL ABONO *

€ 0,00

— escribe un monto —

El monto no puede superar el saldo pendiente con el proveedor.

MÉTODO DE PAGO *

— Seleccionar —

CUENTA DE LA QUE SALE EL DINERO *

3.0.1 - Pago a proveedores

Típicamente Efectivo, Caja o Banco — la cuenta de donde sale el dinero hacia el proveedor.

TIPO DE MOVIMIENTO *

Pago a proveedor

Ej: Pago a proveedor; Abono parcial, Cancelación de deuda.

DESCRIPCIÓN (OPCIONAL)

Ej: Pago a lentes S.AAAA del 31/05/2026

Cancelar Registrar Pago

Datos de la CxP

PROVEEDOR
lentes S.AAAA
VENCIAMIENTO
22/05/2026

EXP ID:
#7

MONTO TOTAL
€ 1 000,00

YA PAGADO
€ 0,00

SALDO PENDIENTE
€ 1 000,00

Fuente: Elaboración propia

Figura 25*Diseño interfaz cuentas por pagar editar*

 **Editar CXP · lentes S.AAAA** 

1 Estás modificando la DEUDA ORIGINAL

Cambia aquí el monto si te equivocaste al registrar la cuenta (ej. registraste €100.000 cuando eran €200.000). Esto **NO** es un pago ni un abono. Para registrar un pago al proveedor, usa el botón verde "Registrar abono" desde el listado o el detalle.

2 ESTADO ACTUAL DE LA CUENTA

MONTO TOTAL ORIGINAL	TOTAL ABONADO	SALDO PENDIENTE ACTUAL
€ 1 000,00	€ 0,00	€ 1 000,00

NUEVO MONTO TOTAL *

1000

€ 1 000,00

Modifica solo el monto total de la deuda original. El saldo pendiente se recalculará automáticamente restándole lo ya abonado.

3 VISTA PREVIA DEL CAMBIO

Nuevo Saldo Pendiente = € 1 000,00 – € 0,00 = **€ 1 000,00**

FECHA VENCIMIENTO *

22/05/2026 

 **viernes, 22 de mayo de 2026 · Fecha en el pasado**

4 No se puede crear ni editar una CXP con una fecha anterior a hoy. Elige una fecha posterior a hoy.
La fecha debe ser posterior a hoy (no se permite hoy ni días pasados).

OBSERVACIONES

Fuente: Elaboración propia

Figura 26*Diseño interfaz caja chica*

+

 Registrar Movimiento

CUENTA CONTABLE *

— Seleccione —

▼

TIPO DE TRANSACCIÓN *

— SELECCIONE CUENTA PRIMERO —

TIPO DE MOVIMIENTO *

— Seleccione —

▼

MÉTODO DE PAGO *

— Seleccione —

▼

MONTO (€) *

0.00

FECHA Y HORA

31/05/2026 15:16

📅

DESCRIPCIÓN *

Cancelar

Guardar

Figura 27*Diseño interfaz caja chica editar*

✎ Editar Movimiento ✕

i Solo puedes editar **método**, **tipo de mov.** y **descripción**. El monto, cuenta y tipo de transacción son fijos para preservar la integridad contable. Si necesitas cambiar algo más, elimina este movimiento y crea uno nuevo.

CUENTA	MONTO	TRANSACCIÓN
505.0.9 — Lapiceros	₡ 50 000,00	Egreso

TIPO DE MOVIMIENTO •	MÉTODO DE PAGO •
Gasto Caja Chica ▼	Tarjeta de crédito ▼

FECHA Y HORA

22/05/2026 12:11 📅

DESCRIPCIÓN •

prueba del 50% ✎

Cancelar 📄 Guardar Cambios

Fuente: Elaboración propia

Figura 28*Diseño interfaz cierre de caja*

Realizar Cierre de Caja

Este es el resumen del día. Revisalo y confirmá el cierre. El efectivo en caja se calcula automáticamente desde los métodos marcados como "Es Efectivo".

Ingresos del día €21 187,50

↳ Tarjeta de débito €21 187,50

Gastos del día €0,00

↳ Sin gastos registrados

Ganancia del día €21 187,50

Efectivo físico esperado €0,00
Calculado automáticamente desde métodos marcados como "Es Efectivo"

Cancelar Confirmar Cierre

*Fuente: Elaboración propia***Figura 29***Diseño interfaz cierre de caja registrar un ingreso manual*

Registrar Ingreso por Pago

FACTURA *

— Seleccione una factura —

MÉTODO DE PAGO *

— Seleccione —

CUENTA CONTABLE *

— Seleccione —

MONTO (€) *

0.00

Cancelar Guardar

Fuente: Elaboración propia

Figura 30*Diseño interfaz presupuesto*

Formulario de diseño de interfaz para crear un nuevo período presupuestario. El formulario tiene un encabezado de color azul con el título "NUEVO PERÍODO".

NOMBRE DEL PERÍODO *
Ej: Presupuesto Anual 2026

FECHA INICIO *
dd/mm/aaaa

FECHA FIN *
dd/mm/aaaa

MONTO GLOBAL MENSUAL (€) *
0.00

DURACIÓN DEL PERÍODO
— meses

Fuente: Elaboración propia**Figura 31***Diseño interfaz presupuesto editar*

Formulario de diseño de interfaz para editar un período presupuestario existente. El formulario tiene un encabezado de color naranja con el título "DATOS DEL PERÍODO".

NOMBRE DEL PERÍODO *
Presupuesto Junio 2026

FECHA INICIO *
01/06/2026

FECHA FIN *
30/06/2026

MONTO GLOBAL MENSUAL (€) *
1000000.00

Botones: Guardar Cambios, Cancelar

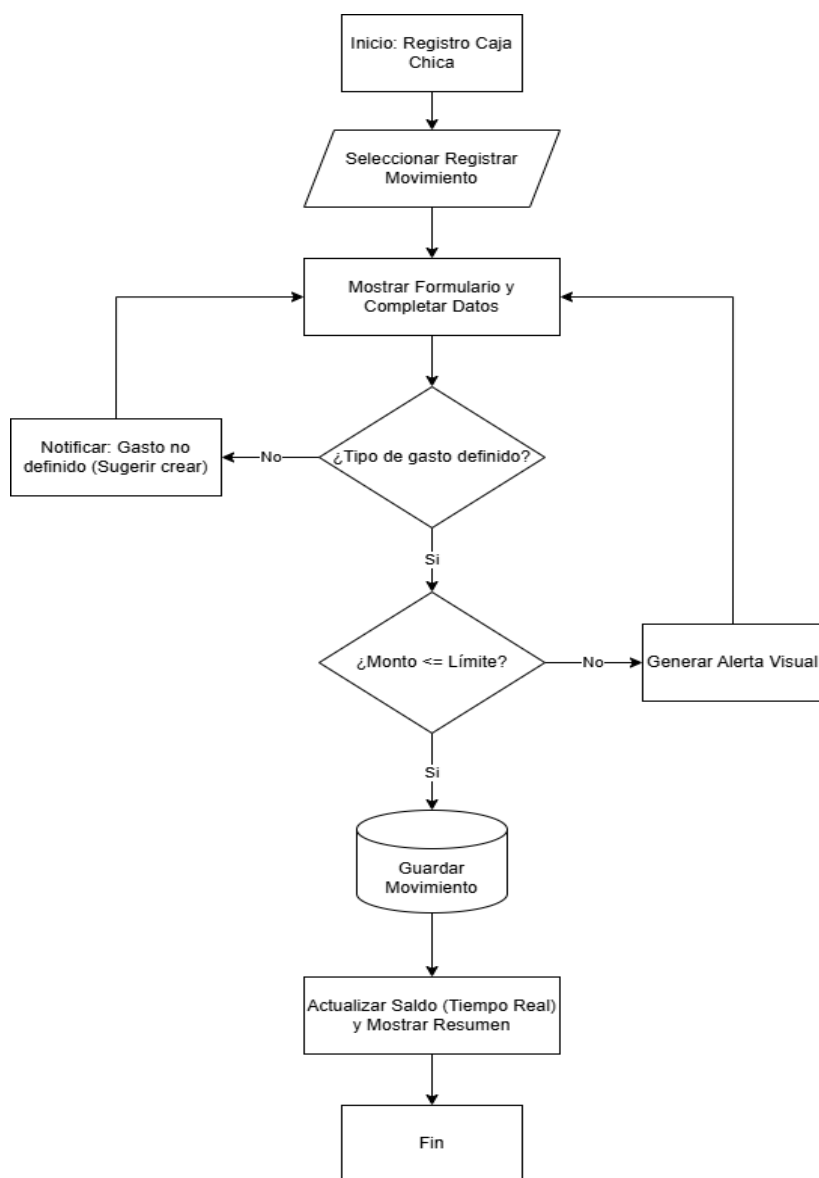
Fuente: Elaboración propia

Diseño físico de la base de datos

Diseño de procesos

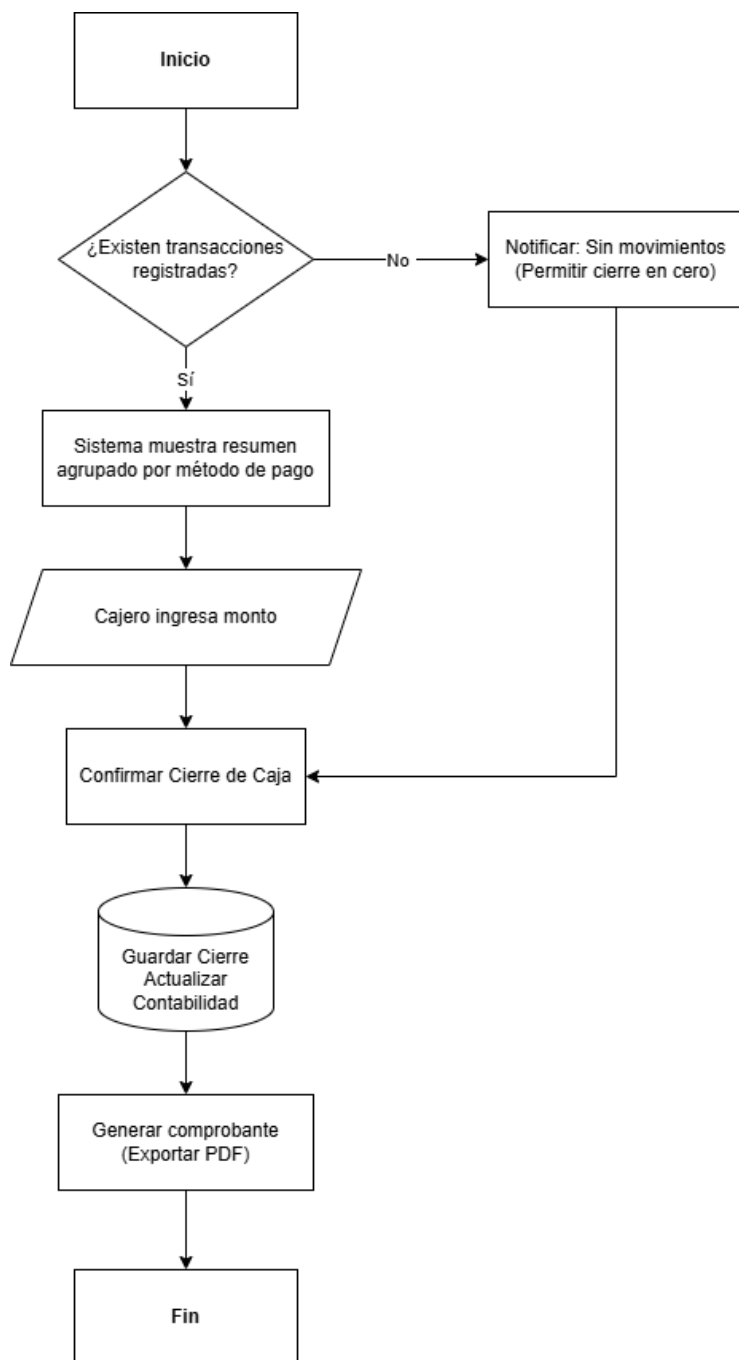
Figura 33

Caja chica



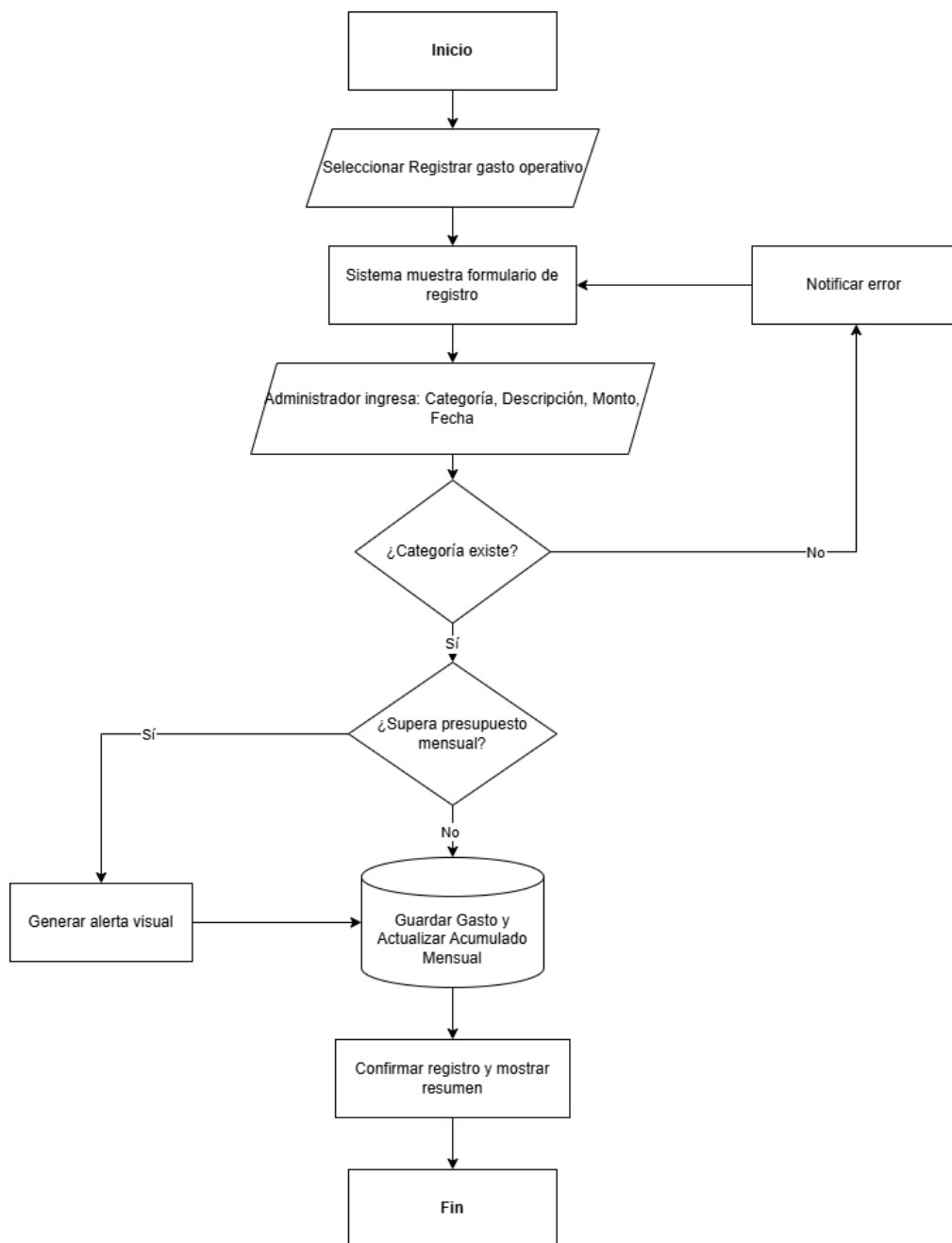
Fuente: Elaboración propia

Figura 34
Cierre de caja



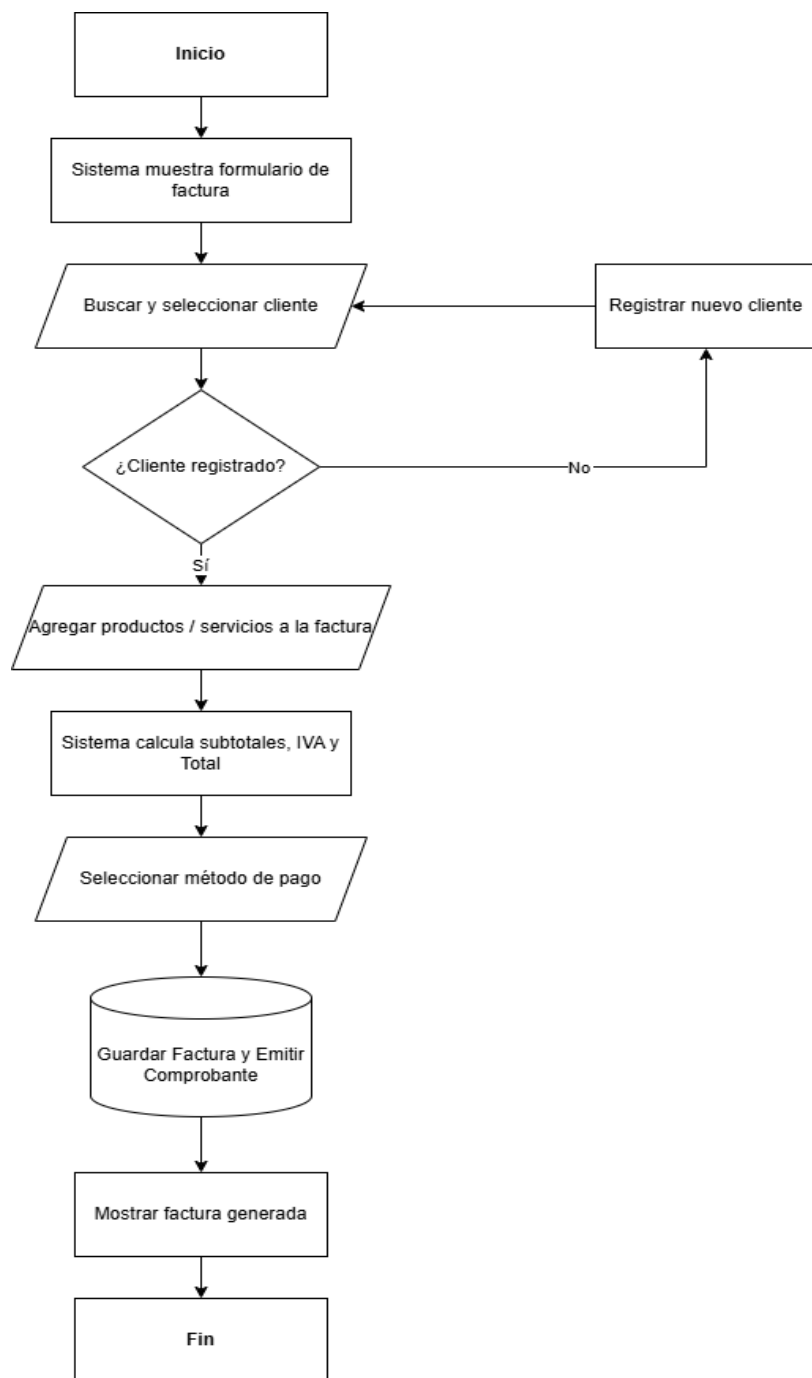
Fuente: Elaboración propia

Figura 35
Calcular gastos



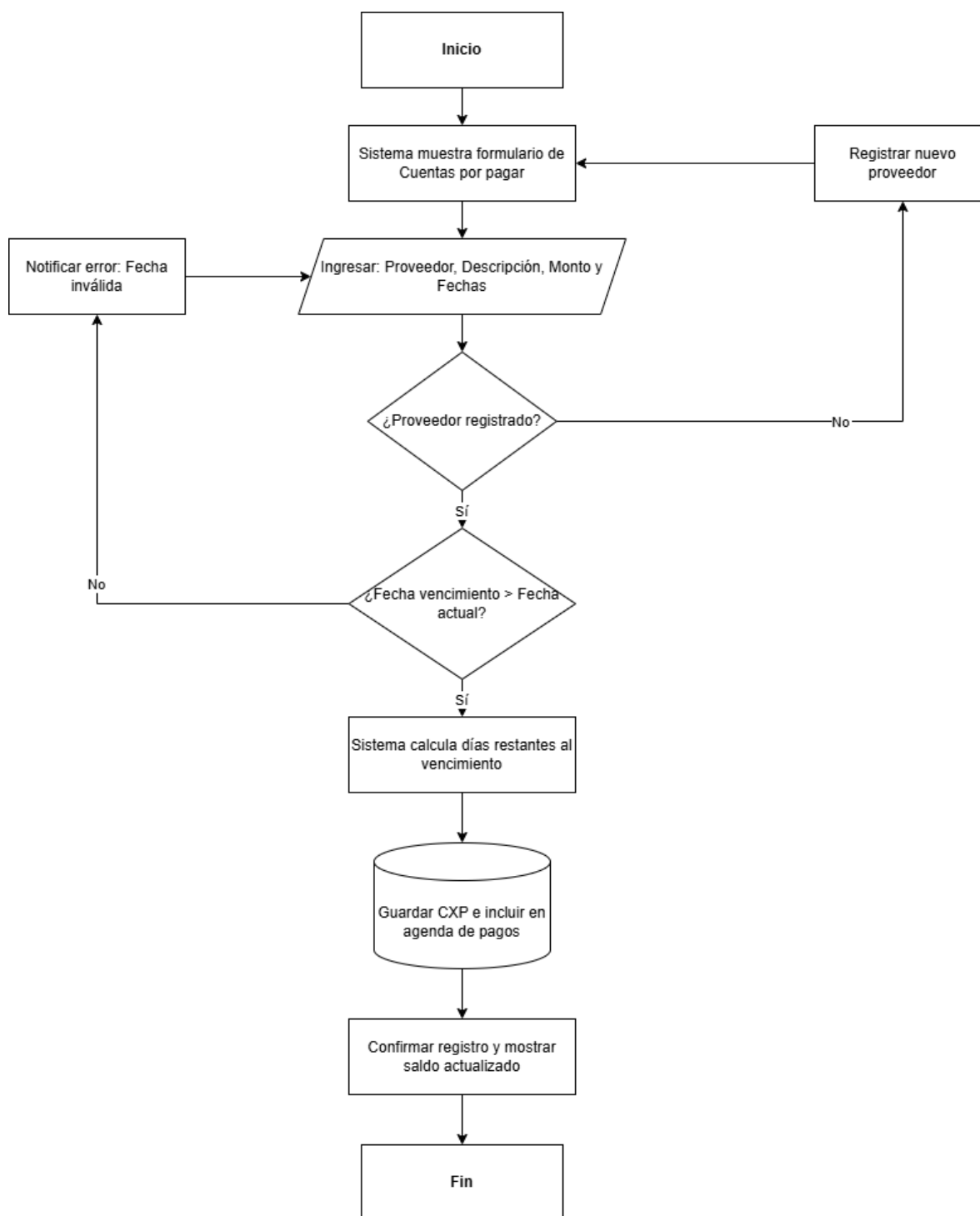
Fuente: Elaboración propia

Figura 36
Gestionar facturación



Fuente: Elaboración propia

Figura 37
Gestión de cuentas por pagar



Fuente: Elaboración propia

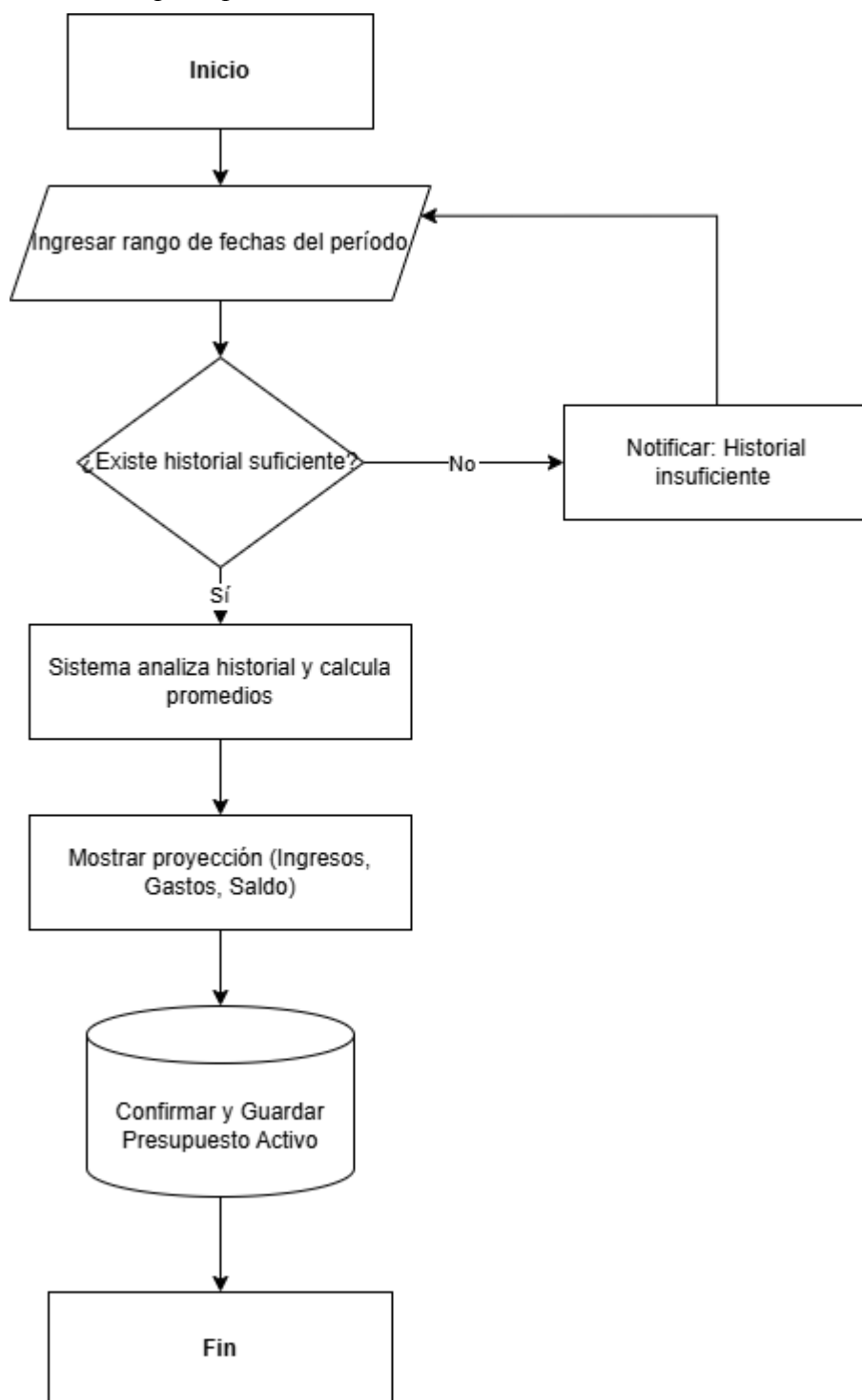
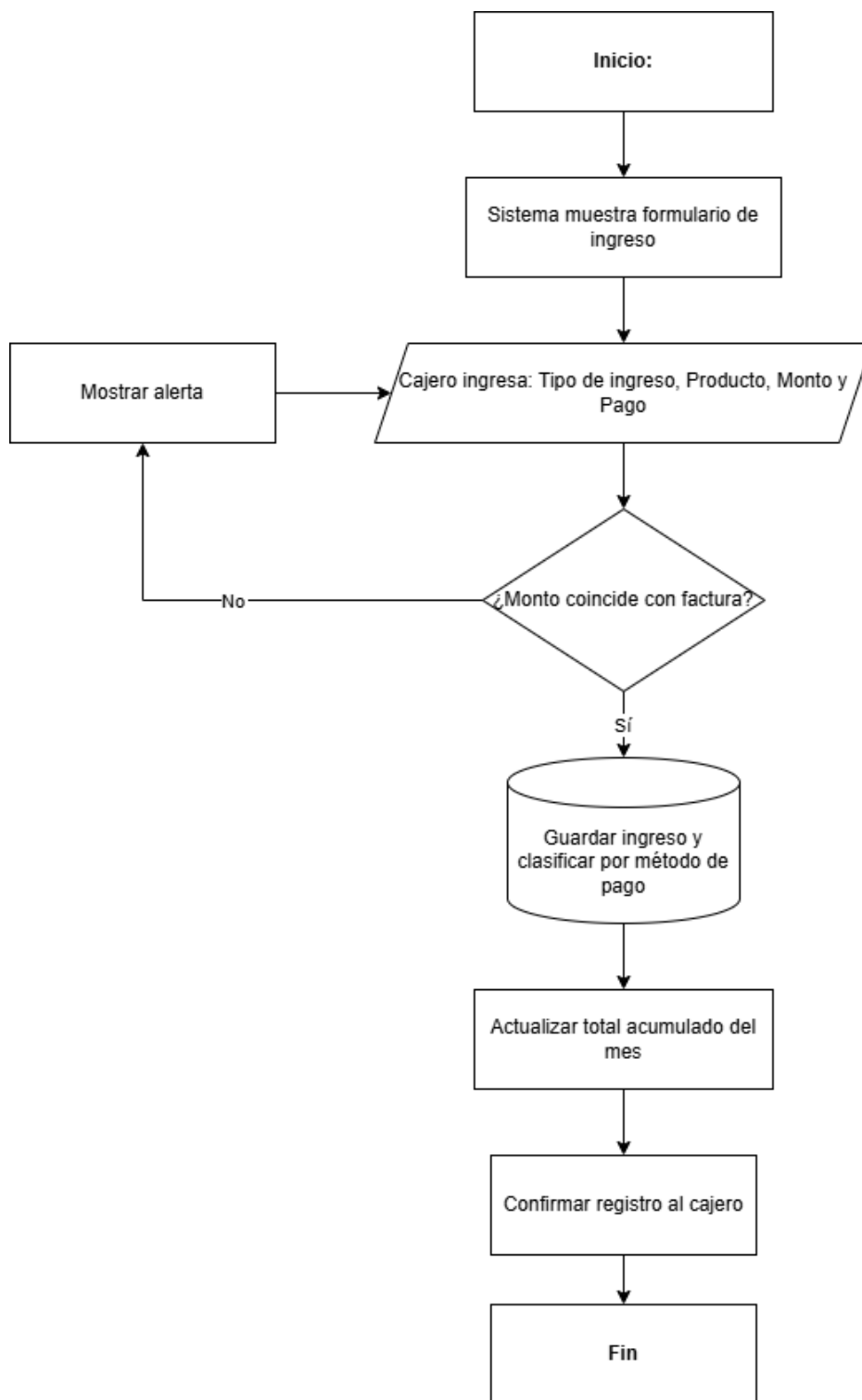
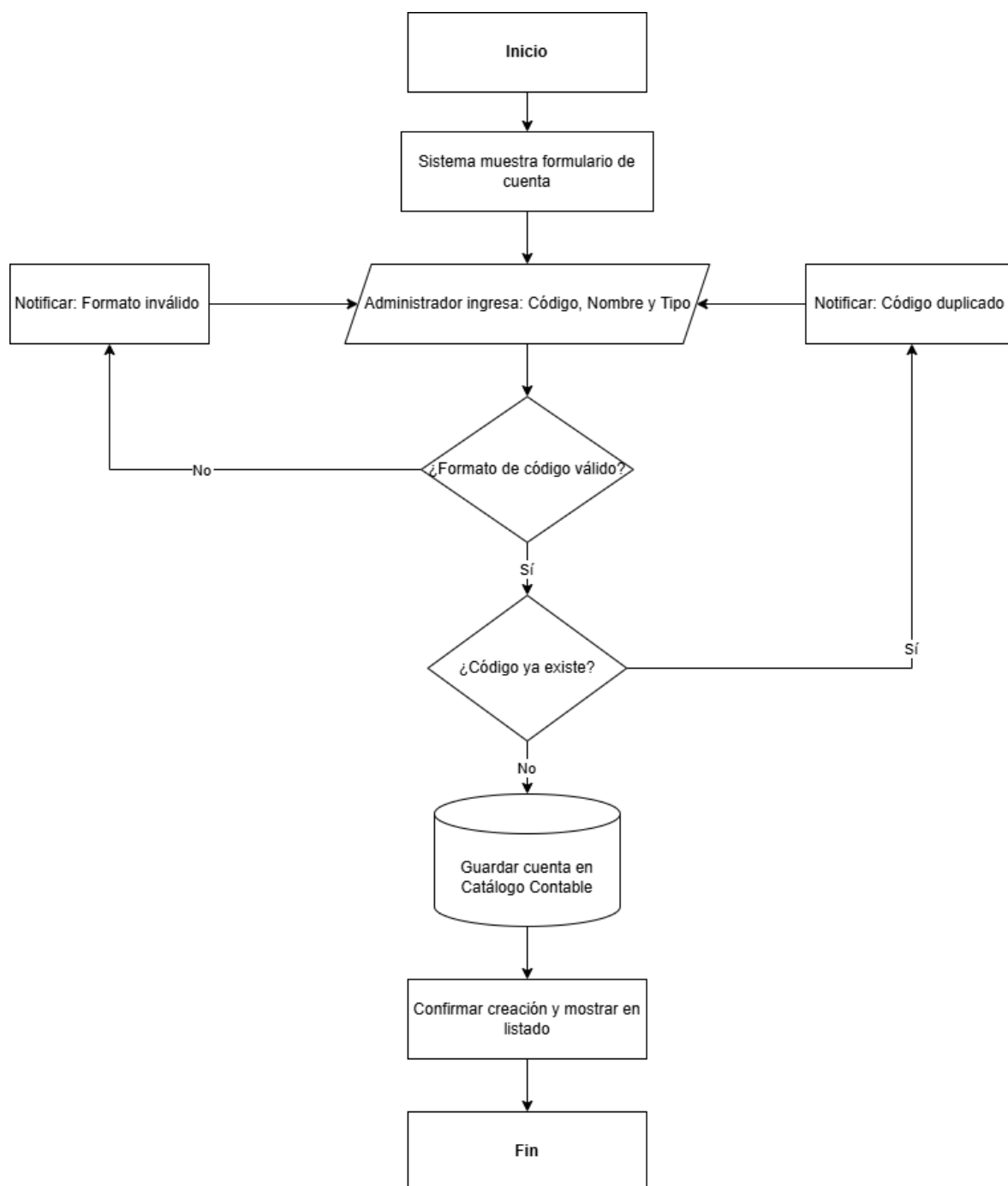
Figura 38*Gestión del presupuesto**Fuente: Elaboración propia*

Figura 39
Gestión de ingresos

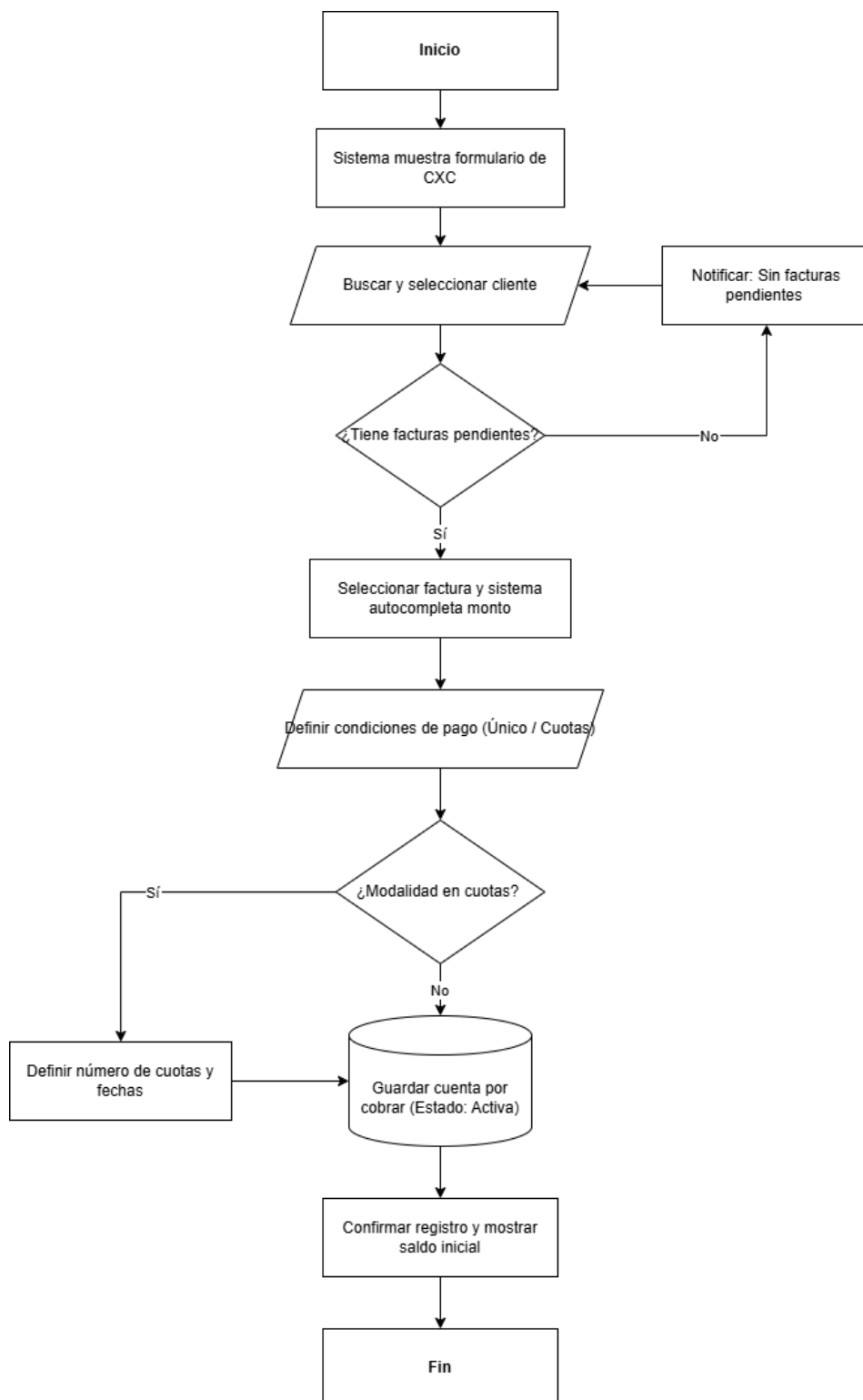


Fuente: Elaboración propia

Figura 40
Gestionar cuentas contables



Fuente: Elaboración propia

Figura 41*Gestión de cuentas por cobrar**Fuente: Elaboración propia*

Diseño de salidas

Figura 42
Catálogo contable

#	CÓDIGO	NOMBRE DE LA CUENTA	TIPO	NATURALEZA	CAJA CHICA	BALANCE	ESTADO	ACCIO
1.1.0	papel higiénico	Gasto caja chica	Deudora	CC	₡ 0,00	✓ Activa		
1.1.01	Materiales	Gasto caja chica	Deudora	CC	₡ 0,00	✓ Activa		
1.1.10	Papel de Agua	Gasto	Deudora		₡ 0,00	✓ Activa		

Fuente: Elaboración propia

Figura 43
Catalogo contable Excel

A	B	C	D	E	F
Catalogo Contable					
Codigo de Cuenta	Tipo de Cuenta	Nombre de la Cuenta	Naturaleza	Balance Inicial	Estado
1-01-01	Cuenta por pagar	Cuenta de proveedor	Deudora	1,500.00	Activo
1-01-02	Activo Circulante	Caja General	Deudora	5,000.00	Activo
2-01-01	Pasivo	Prestamo Bancario	Acreedora	80,000.00	Activo
5-01-01	Gasto Operativo	Suministros Oficina	Deudora	0.00	Inactivo

Fuente: Elaboración propia

Figura 44
Catalogo contable PDF

Codigo de Cuenta	Tipo de Cuenta	Nombre de la Cuenta	Naturaleza	Balance (colones)	Estado
1-01-01	Cuenta por pagar	Cuenta de proveedor	Deudora	1,500.00	Activo
1-01-02	Activo Circulante	Caja General	Deudora	5,000.00	Activo
2-01-01	Pasivo	Prestamo Bancario	Acreedora	80,000.00	Activo
5-01-01	Gasto Operativo	Suministros Oficina	Deudora	0.00	Inactivo

Fuente: Elaboración propia

Figura 45
Factura

FAC-20260530-0003	30/05/2026 16:55	Rocío López 111320891	Tyron	₡ 150 000,00	₡ 19 500,00	₡ 169 500,00	PAGADA
FAC-20260530-0002	30/05/2026 14:51	Tyron Tyron 444000977	Tyron	₡ 112 500,00	₡ 14 625,00	₡ 127 125,00	ANULADA
FAC-20260530-0001	30/05/2026 14:50	Tyron Tyron 444000977	Tyron	₡ 37 500,00	₡ 4 875,00	₡ 42 375,00	PAGADA
FAC-20260529-0006	29/05/2026 22:54	Tyron Tyron 444000977	Tyron	₡ 123 000,00	₡ 15 990,00	₡ 138 990,00	ANULADA
FAC-20260529-0005	29/05/2026 22:42	Prueba Figueres 555555555	Daniela	₡ 110 000,00	₡ 14 300,00	₡ 124 300,00	ANULADA
FAC-20260529-0004	29/05/2026 22:13	Rocío López 111320891	Daniela	₡ 37 500,00	₡ 4 875,00	₡ 42 375,00	ANULADA

Fuente: Elaboración propia

Figura 46
Factura Excel

N Factura	Cliente	Fecha	Subtotal (colones)	IVA (colones)	Total (colones)	Estado
FE-00003	Juan Perez Rodriguez	26/10/2023	45,000	5,850	50,850	Pagada
FE-00002	Maria Santos Lopez	25/10/2023	120,000	15,600	135,600	Pendiente
FE-00001	Carlos Gomez Ruiz	24/10/2023	30,000	3,900	33,900	Anulada

Fuente: Elaboración propia

Figura 47
Factura PDF

N Factura	Cliente	Fecha	Subtotal (colones)	IVA (colones)	Total (colones)	Estado
FE-00003	Juan Perez Rodriguez	26/10/2023	45,000	5,850	50,850	Pagada
FE-00002	Maria Santos Lopez	25/10/2023	120,000	15,600	135,600	Pendiente
FE-00001	Carlos Gomez Ruiz	24/10/2023	30,000	3,900	33,900	Anulada

Fuente: Elaboración propia

Figura 48
Ingreso

Cuenta	Método	Monto
Venta de Lentes Juan Pérez	Efectivo	€14000
Examen de Vista María García	Tarjeta	€14000
Venta de Accesorios Carlos Ruiz	Efectivo	€14000
Venta de Lentes Ana López	Tarjeta	€14000
Total Ingresos Hoy:		0000

Fuente: Elaboración propia

Figura 49
Ingreso Excel

Cuenta / Referencia	Metodo de Pago	Monto (colones)
Venta de Lentes - Juan Perez	Efectivo	14,000
Examen de Vista - Maria Garcia	Tarjeta	14,000
Venta de Accesorios - Carlos Ruiz	Efectivo	14,000
Venta de Lentes - Ana Lopez	Tarjeta	14,000
Total Ingresos Ho		56,000

Fuente: Elaboración propia

Figura 50
Gastos

#	CÓDIGO	NOMBRE DE LA CUENTA	# MOVIMIENTOS	TOTAL GASTADO	% DEL TOTAL	ACCIÓN
1.1.10		Pago de Agua	0	€0,00	0,0%	Ver movimientos
1.111.444		Alquiler	0	€0,00	0,0%	Ver movimientos
10.0.9		Pago de Luz	1	€200 000,00	44,8%	Ver movimientos
3.0.1		Pago a proveedores	5	€226 000,00	50,7%	Ver movimientos
5.0.0		Marketing	1	€20 000,00	4,5%	Ver movimientos

Fuente: Elaboración propia

Figura 51*Gastos Excel*

Hora	Cuenta	Detalle	Referencia	Monto (colones)
10:15 AM	Suministros Oficina	Papeleria	FAC-9921	-8,400
09:30 AM	Mantenimiento	Reparacion de aire	FAC-8832	-22,000
08:45 AM	Gastos Varios	Limpieza	REC-0045	-5,000
			Total Gastos	-35,400

Fuente: Elaboración propia**Figura 52***Gastos PDF*

Hora	Cuenta	Detalle	Referencia	Monto (colones)
10:15 AM	Suministros Oficina	Papeleria y tintas	FAC-9921	-8,400
09:30 AM	Mantenimiento	Reparacion de aire	FAC-8832	-22,000
08:45 AM	Gastos Varios	Limpieza fachada	REC-0045	-5,000
			Total Gastos del Dia	35,400

Fuente: Elaboración propia**Figura 53***Cuentas por cobrar 1*

#1	Luis Ramos López	FAC-20260528-0001	€ 42 375,00	€ 42 375,00	27/06/2026	ANULADA	👁	🔄
#2	Prueba Figueres Rojas	FAC-20260529-0003	€ 124 300,00	€ 124 300,00	29/08/2026	ANULADA	👁	🔄

Fuente: Elaboración propia

Figura 54*Cuentas por cobrar Excel*

	Cliente	N Factura	Total Factura (colones)	Saldo Pendiente (colones)	Ultimo Pago
2	Juan Perez	FAC-2023-001	120,000.00	85,000.00	10 Oct 2023
3	Maria Garcia	FAC-2023-012	45,000.00	0.00	12 Oct 2023
4	Carlos Rodriguez	FAC-2023-088	210,000.00	105,000.00	05 Oct 2023
5	Ana Lopez	FAC-2023-092	32,000.00	12,000.00	09 Oct 2023

Fuente: Elaboración propia**Figura 55***Cuentas por cobrar PDF*

Cliente	N Factura	Total Factura (colones)	Saldo Pendiente (colones)	Ultimo Pago
Juan Perez	FAC-2023-001	120,000.00	85,000.00	10 Oct 2023
Maria Garcia	FAC-2023-012	45,000.00	0.00	12 Oct 2023
Carlos Rodriguez	FAC-2023-088	210,000.00	105,000.00	05 Oct 2023
Ana Lopez	FAC-2023-092	32,000.00	12,000.00	09 Oct 2023

Fuente: Elaboración propia**Figura 56***Cuentas por pagar 1*

#7	lentes S.AAAA	€ 1 000,00	€ 1 000,00	22/05/2026 Vencida hace 9 días	VENCIDA	   
#13	lentes S.AAAA	€ 76 000,00	€ 0,00	28/06/2026	PAGADA	  
#14	lentes S.AAAA	€ 75 000,00	€ 0,00	29/06/2026	PAGADA	  
#15	lentes S.AAAA	€ 75 000,00	€ 0,00	29/06/2026	PAGADA	  

Fuente: Elaboración propia

Figura 57
Cuentas por pagar Excel

2	Proveedor	Monto Total (colones)	Saldo pendiente (colones)	Fecha Limite	Estado
3	Optica S.A.	15,000.00	15,000.00	15/12/2024	Pendiente
4	Lentes Express	8,420.00	8,420.00	05/12/2024	Pendiente
5					

Fuente: Elaboración propia

Figura 58
Cuentas por pagar PDF

Proveedor	Monto Total (colones)	Saldo Pendiente (colones)	Fecha Limite	Estado
Optica S.A.	15,000.00	15,000.00	15/12/2024	Pendiente
Lentes Express	8,420.00	8,420.00	05/12/2024	Pendiente

Fuente: Elaboración propia

Figura 59
Caja chica

HISTORIAL DE MOVIMIENTOS							
26/05/2026 22:18	1.1.11 Reposición Caja	Reposición de caja	Tarjeta de crédito	INGRESO	+ €5 000,00	Ver	Editar
26/05/2026 02:18	1.1.11 Reposición Caja	Reposición de caja	Efectivo	INGRESO	+ €20 000,00	Ver	Editar
22/05/2026 12:11	505.0.9 Lapiceros	Gasto Caja Chica	Tarjeta de crédito	EGRESO	- €50 000,00	Ver	Editar

Fuente: Elaboración propia

Figura 60
Caja chica Excel

2	Fecha de Cambio	Usuario	Monto Apertura (colones)	Monto Limite (colones)	Activo
3	2023-10-25 09:15	Admin Principal	5,000.00	500.00	Si
4	2023-10-01 08:00	Juan Perez	3,000.00	300.00	No
5	2023-09-15 14:20	Admin Principal	2,500.00	250.00	No
6					
7					

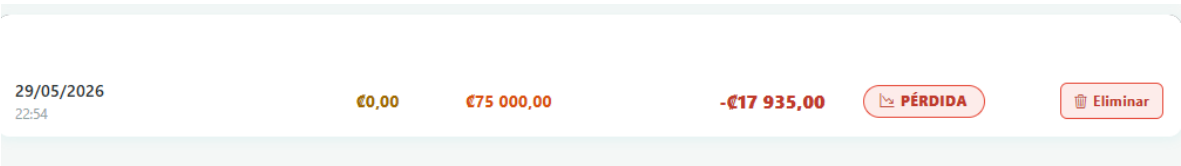
Fuente: Elaboración propia

Figura 61
Caja chica PDF

Fecha de Cambio	Usuario	Monto Apertura (colones)	Monto Limite (colones)	Activo
2023-10-25 09:15	Admin Principal	5,000.00	500.00	Si
2023-10-01 08:00	Juan Perez	3,000.00	300.00	No
2023-09-15 14:20	Admin Principal	2,500.00	250.00	No

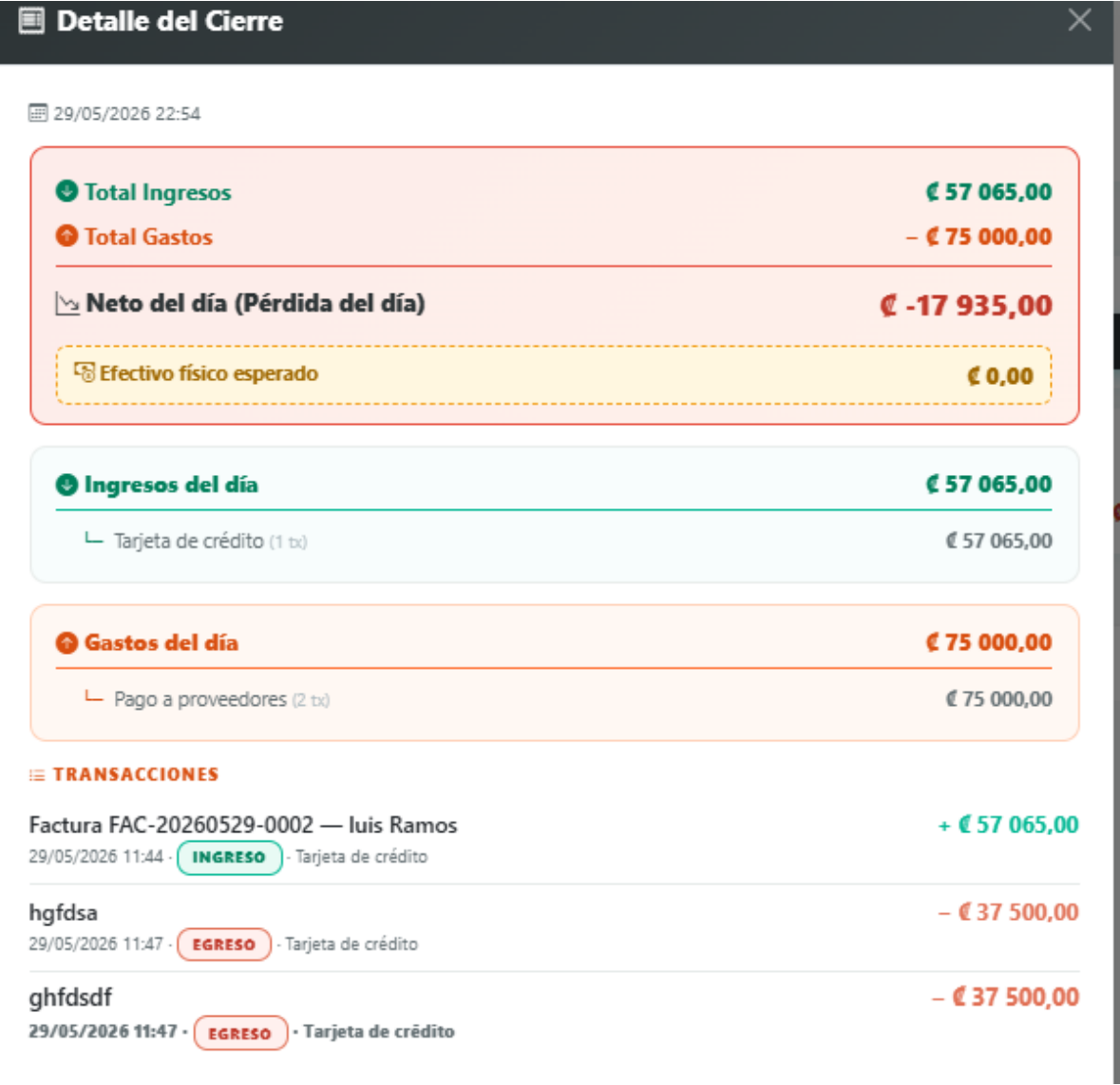
Fuente: Elaboración propia

Figura 62
Cierre de caja



Fuente: Elaboración propia

Figura 63
Cierre de caja,detalle



Fuente: Elaboración propia

Figura 64
Cierre de caja PDF

Fecha Cierre	Saldo Sistema (colones)	Saldo Real (colones)	Diferencia (colones)	Usuario Cajero
23/05/2024 18:30	1,120.00	1,120.00	0.00	Admin_Caja_01
22/05/2024 19:05	945.50	950.00	4.50	User_Caja_02
21/05/2024 18:45	1,500.00	1,480.00	-20.00	Admin_Caja_01

Fuente: Elaboración propia

Figura 65
Presupuesto

COMPARATIVO POR CUENTA							
Haga clic en una fila para ver el desglose de movimientos.							
1.1.11	Reposición Caja	DEUDORA	€25 000,00	€0,00	- €25 000,00	HOLGADO	0.0%
1.1.7	Ventas por servicios	ACREEDORA	€14 690,00	€0,00	- €14 690,00	DÉFICIT	0.0%
1.2.0	Ventas	ACREEDORA	€858 800,00	€0,00	- €858 800,00	DÉFICIT	0.0%
10.0.9	Pago de Luz	DEUDORA	€200 000,00	€0,00	- €200 000,00	HOLGADO	0.0%
3.0.1	Pago a proveedores	DEUDORA	€226 000,00	€0,00	- €226 000,00	HOLGADO	0.0%
5.0.0	Marketing	DEUDORA	€20 000,00	€0,00	- €20 000,00	HOLGADO	0.0%
505.0.9	Lapiceros	DEUDORA	€50 000,00	€0,00	- €50 000,00	HOLGADO	0.0%

Fuente: Elaboración propia

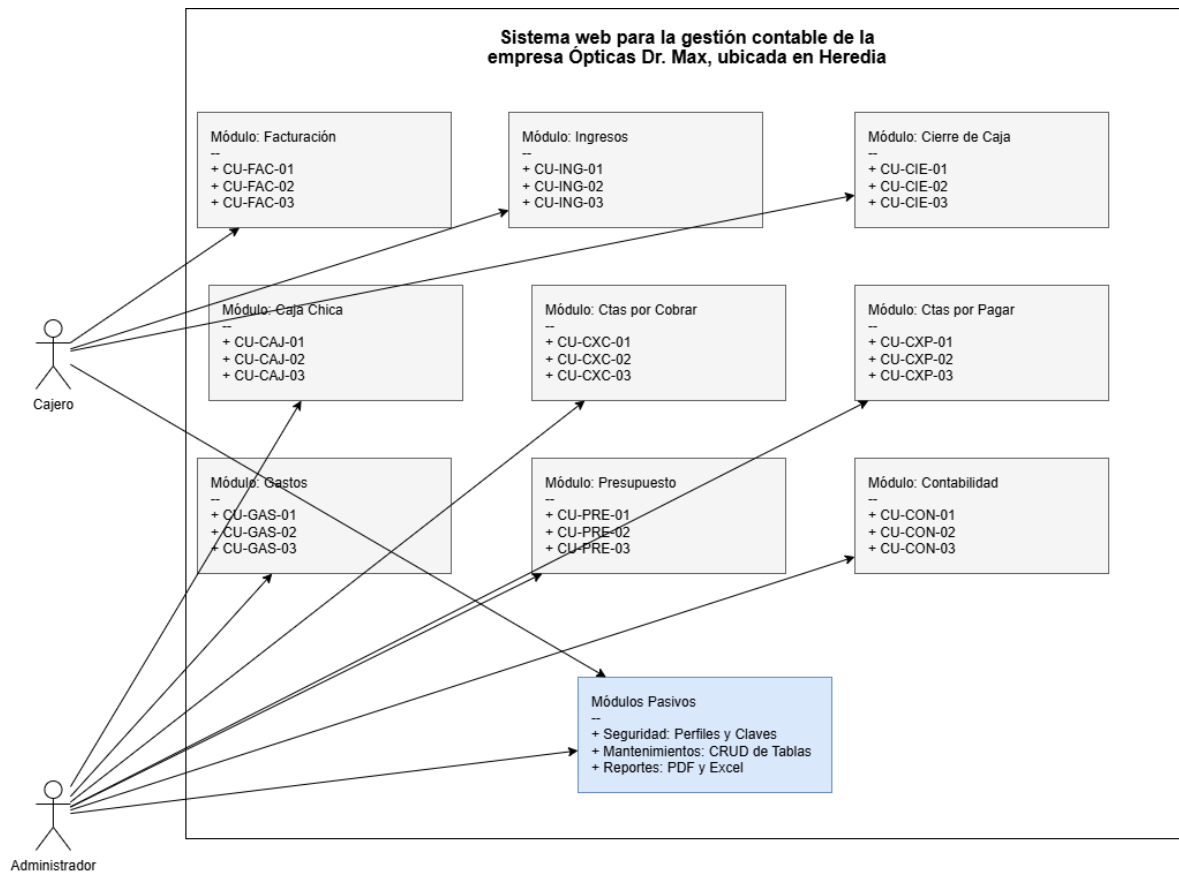
Figura 66
Presupuesto Excel

	Cuenta	Mes	Presupuestado (colones)	Real Acumulado (colones)	Variacion (colones)	Porcentaje de Ejecucion
2						
3	601-01 Gastos Operativos	Enero	500,000.00	450,000.00	50,000.00	90%
4	601-02 Ingresos	Enero	1,200,000.00	1,200,000.00	0.00	100%
5	601-04 Suministros	Enero	350,000.00	385,000.00	-35,000.00	110%

Fuente: Elaboración propia

Diseño de UML

Figura 67
UML



Fuente: Elaboración propia

Programación/entradas

En este apartado se muestra parte del código que se utiliza con el fin de elaborar las entradas de datos del sistema contable, las cuales brindan funcionalidad a los módulos. En el apartado del *back-end* se utilizó Visual Studio con algunas de sus herramientas, como Entity Framework, Razor y JavaScript. Estas herramientas facilitaron la programación de las entradas al brindar usos útiles en cada fase.

Figura 68*Entradas contabilidad*

```

62 ViewBag.FiltroCodigo = filtroCodigo;
63 ViewBag.FiltroNombre = filtroNombre;
64 ViewBag.FiltroTipo = filtroTipo;
65 return View();
66
67
68 [HttpPost]
69 public IActionResult Crear(string codigo, string nombre, int idTipoCuenta, string naturaleza,
70 bool aplicaCajachica = false, bool aplicaCxp = false, bool aplicaCxc = false,
71 int? idTipoMovDefault = null)
72 {
73     if (string.IsNullOrEmpty(codigo) || codigo.Trim().Length < 1)
74     { TempData["Error"] = "El código es obligatorio."; return RedirectToAction("Cuentas"); }
75     if (!System.Text.RegularExpressions.Regex.IsMatch(codigo.Trim(), @"^d+(\.d+)?$"))
76     { TempData["Error"] = "El código solo puede contener números y puntos (ej: 1.1.01)."; return
77       RedirectToAction("Cuentas"); }
78     if (string.IsNullOrEmpty(nombre) || nombre.Trim().Length < 2)
79     { TempData["Error"] = "El nombre debe tener al menos 2 caracteres."; return RedirectToAction
80       ("Cuentas"); }
81     if (!System.Text.RegularExpressions.Regex.IsMatch(nombre.Trim(), @"^[a-zA-ZáéíóúÁÉÍÓÚÑñ\s]+$"))
82     { TempData["Error"] = "El nombre solo puede contener letras."; return RedirectToAction("Cuentas"); }
83     if (naturaleza != "D" && naturaleza != "A")
84     { TempData["Error"] = "Naturaleza inválida (debe ser D o A)."; return RedirectToAction("Cuentas"); }
85     EjecutarGestionarCuenta("CREAR", null, codigo.Trim(), nombre.Trim(), idTipoCuenta, naturaleza);
86     ActualizarAplicaCajachicaPorCodigo(codigo.Trim(), aplicaCajachica);
87     ActualizarFlagsCxpCxcPorCodigo(codigo.Trim(), aplicaCxp, aplicaCxc);
88     ActualizarTipoMovDefaultPorCodigo(codigo.Trim(), idTipoMovDefault);
89     return RedirectToAction("Cuentas");
90 }
91 [HttpPost]

```

Fuente: Elaboración propia

Se elaboró un método para crear una cuenta en el controlador de contabilidad con la función de agregar: código, nombre, tipo de cuenta, naturaleza, si aplica para un módulo en específico y la clase de movimiento al cual está vinculado. Esto último se realizó, con el fin de tener las cuentas contables organizadas en un módulo y poder registrar nuevas cuentas en caso de que sea necesario.

Figura 69*Entradas factura*

```

102 ViewBag.IVA_PCT = IVA_PCT * 100;
103 return View(new Factura { Fecha = DateTime.Now });
104
105
106 // -- CREATE POST --
107 [HttpPost]
108 [ValidateAntiForgeryToken]
109 public IActionResult Create(
110 int idCliente,
111 int idMetodoPago,
112 bool esCredito,
113 int numeroPlazas,
114 string? lineasJson)
115 {
116     if (idCliente <= 0) { TempData["Error"] = "Seleccionar cliente."; return RedirectToAction("Create"); }
117     if (idMetodoPago <= 0) { TempData["Error"] = "Seleccionar método de pago."; return RedirectToAction("Create"); }
118     if (string.IsNullOrEmpty(lineasJson) || lineasJson == "[ ]")
119     { TempData["Error"] = "Agregar al menos una línea."; return RedirectToAction("Create"); }
120
121     List<LineaInput> lineasIn;
122     try
123     {
124         lineasIn = JsonSerializer.Deserialize<List<LineaInput>>(lineasJson,
125             new JsonSerializerOptions { PropertyNameCaseSensitive = true }) ?? new();
126     }
127     catch (Exception ex) { TempData["Error"] = "Líneas inválidas: " + ex.Message; return RedirectToAction("Create"); }
128
129     if (lineasIn.Count == 0) { TempData["Error"] = "Agregar al menos una línea."; return RedirectToAction("Create"); }
130
131     DateTime? fechaVencimientoCredito = null;
132     if (esCredito)
133     {
134         if (numeroPlazas < 1) numeroPlazas = 1;
135         if (numeroPlazas > 3)
136         {
137             TempData["Error"] = "Máximo 3 meses de crédito permitidos por política de empresa.";
138             return RedirectToAction("Create");
139         }
140         fechaVencimientoCredito = DateTime.Today.AddMonths(numeroPlazas);
141     }
142 }
143

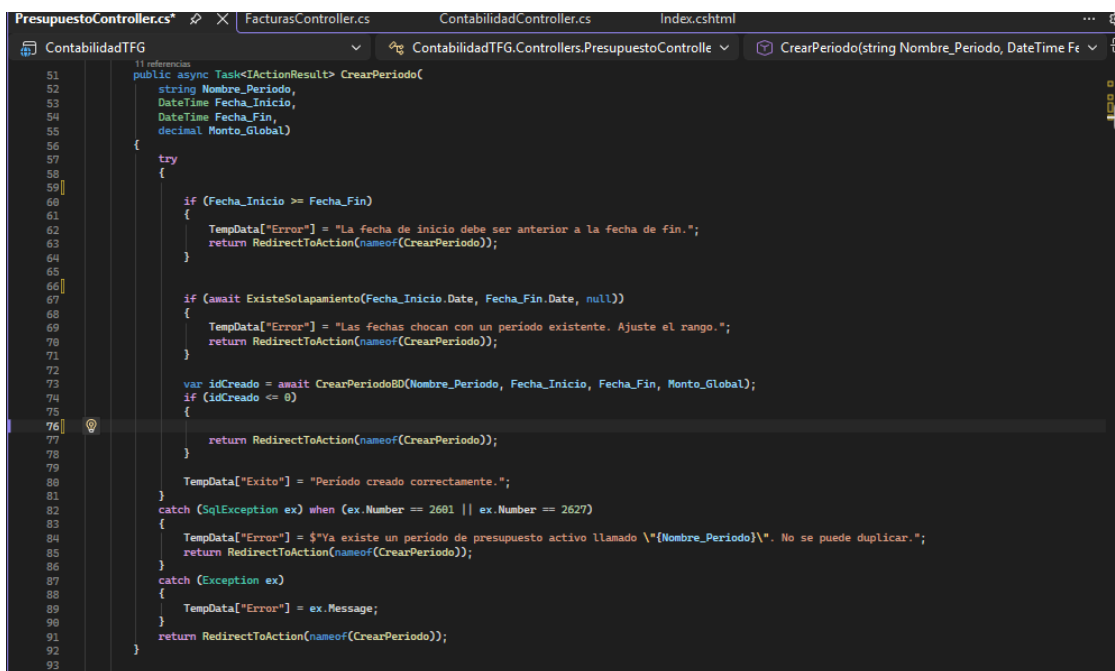
```

Fuente: Elaboración propia

Se implementó un método Create en el controlador de facturas con la función de recibir: Cliente, MetodoPago, Credito, numeroPlazos. El método valida que Cliente y MetodoPago sean mayores que cero; en caso contrario, asigna mensajes en TempData["Error"] y redirige a la acción Create.

Figura 70

Entradas presupuesto



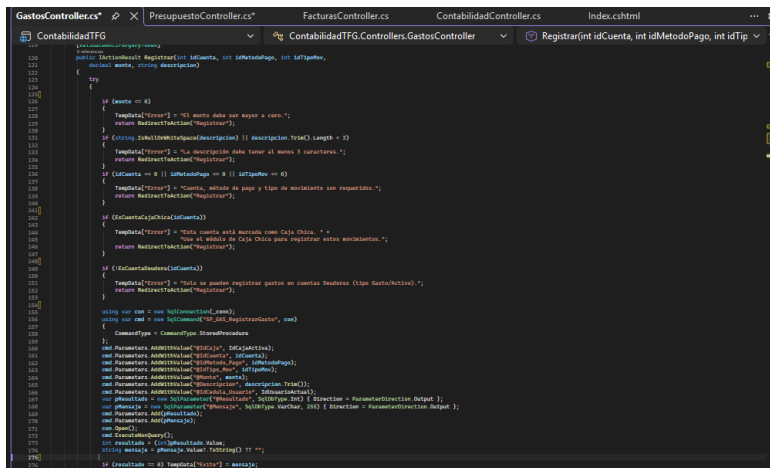
```

11 referencia
51 public async Task<ActionResult> CrearPeriodo(
52     string Nombre_Periodo,
53     DateTime Fecha_Inicio,
54     DateTime Fecha_Fin,
55     decimal Monto_Global)
56 {
57     try
58     {
59         if (Fecha_Inicio >= Fecha_Fin)
60         {
61             TempData["Error"] = "La fecha de inicio debe ser anterior a la fecha de fin.";
62             return RedirectToAction(nameof(CrearPeriodo));
63         }
64
65         if (await ExisteSolapamiento(Fecha_Inicio.Date, Fecha_Fin.Date, null))
66         {
67             TempData["Error"] = "Las fechas chocan con un periodo existente. Ajuste el rango.";
68             return RedirectToAction(nameof(CrearPeriodo));
69         }
70
71         var idCreado = await CrearPeriodoBD(Nombre_Periodo, Fecha_Inicio, Fecha_Fin, Monto_Global);
72         if (idCreado <= 0)
73         {
74             return RedirectToAction(nameof(CrearPeriodo));
75         }
76
77         TempData["Exito"] = "Periodo creado correctamente.";
78     }
79     catch (SqlException ex) when (ex.Number == 2601 || ex.Number == 2627)
80     {
81         TempData["Error"] = $"Ya existe un periodo de presupuesto activo llamado \"{Nombre_Periodo}\". No se puede duplicar.";
82         return RedirectToAction(nameof(CrearPeriodo));
83     }
84     catch (Exception ex)
85     {
86         TempData["Error"] = ex.Message;
87         return RedirectToAction(nameof(CrearPeriodo));
88     }
89 }
90
91
92
93

```

Fuente: Elaboración propia

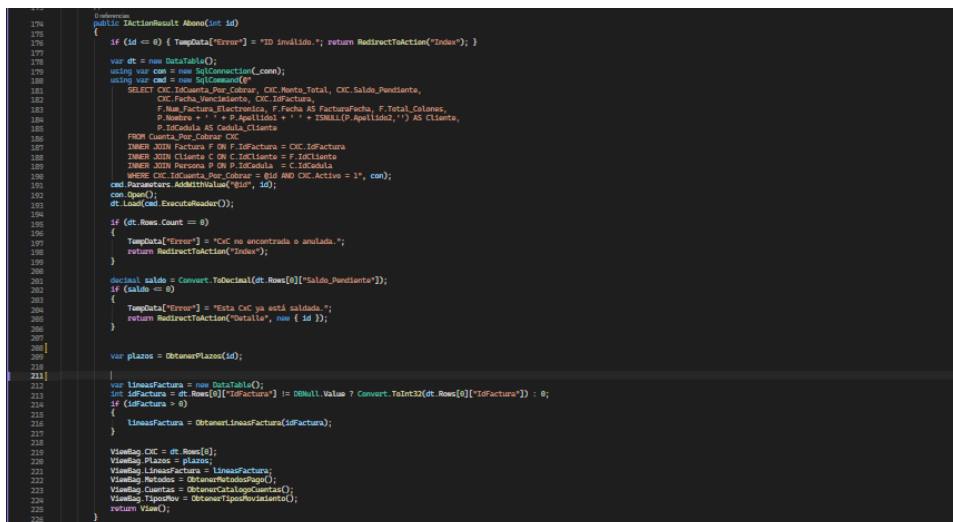
Se creó CrearPeriodo, que recibe Nombre_período, fecha_Inicio, fecha_Fin y Monto_Global; valida que el inicio sea anterior al fin, evita solapamientos con períodos existentes, inserta el período en la base de datos, maneja duplicados por clave única con excepciones SQL y proporciona un reporte si se logró mediante TempData.



Fuente: Elaboración propia

Se definió el método *Registrar* en el controlador de gastos, el cual recibe Cuenta, MetodoPago, TipoMov, monto y descripción; valida las entradas (cuenta, método, descripción y montos), ejecuta el procedimiento almacenado SP_GastoRegistrar_Gasto con los parámetros correspondientes y obtiene mensajes de salida para mostrar el resultado en TempData y redirigir según el caso.

Figura 72
Entradas cuentas por pagar



Fuente: Elaboración propia

Método Abono que valida el id, consulta la factura y el cliente en la base de datos, verifica que exista y tenga saldo pendiente, obtiene plazos y líneas de factura, prepara datos en ViewBag (plazos y catálogos) y devuelve la vista para procesar el abono; usa TempData para errores y redirecciones.

Figura 73

Entradas cuentas por cobrar

```

101 public IActionResult Registrar(int idProveedor,
102 [Microsoft.AspNetCore.Mvc.FromForm(Name = "montoTotal")] string? montoTotalRaw,
103 DateTime fechaVencimiento, string? observaciones)
104 {
105     decimal montoTotal = ParsearMontoFlexible(montoTotalRaw);
106
107     if (idProveedor <= 0) { TempData["Error"] = "Seleccioná un proveedor válido."; return RedirectToAction("Registrar"); }
108     if (!ExisteProveedorActivo(idProveedor)) { TempData["Error"] = "El proveedor no existe o está inactivo."; return RedirectToAction("Registrar"); }
109     if (montoTotal <= 0)
110     {
111         TempData["Error"] = string.IsNullOrEmpty(montoTotalRaw)
112             ? "Ingresá un monto total mayor a cero."
113             : $"No pude interpretar el monto '{montoTotalRaw}'. Usá formato como 1000.00 o 1000,00.";
114         return RedirectToAction("Registrar");
115     }
116     if (montoTotal > 999.999.999.99m)
117     { TempData["Error"] = "El monto excede el máximo permitido."; return RedirectToAction("Registrar"); }
118     if (fechaVencimiento.Date < DateTime.Today.AddYears(-1) || fechaVencimiento.Date > DateTime.Today.AddYears(5))
119     { TempData["Error"] = "Fecha fuera de rango (entre 1 año atrás y 5 años adelante)."; return RedirectToAction("Registrar"); }
120
121     if (InsertarCxpDirecto(idProveedor, montoTotal, fechaVencimiento.Date, observaciones, out int? nuevoId, out string err))
122     {
123         try
124         {
125             using var con3 = new SqlConnection(_conn); con3.Open();
126             RegistrarAuditoria(con3, 1, "Cuenta_Por_Pagar", nuevoId ?? 0,
127                 $"Registro CXP {montoTotal:N2}, vence {fechaVencimiento:dd/MM/yyyy}");
128         }
129         catch { }
130         TempData["Exito"] = "CXP registrada.";
131     }
132     else TempData["Error"] = "No se pudo crear. " + err;
133     return RedirectToAction("Index");
134 }
135

```

Fuente: Elaboración propia

Se creó un método llamado *Registrar* que procesa cuentas por pagar: valida proveedor activo, rango y límites del monto, valida fecha de vencimiento en un rango permitido, inserta el registro en la base de datos y comunica éxito o error mediante TempData.

Figura 74

Entradas cierre de caja

```

276 public async Task<ActionResult> RealizarCierre(decimal Monto_Fisico_Efectivo)
277 {
278     try
279     {
280         if (Monto_Fisico_Efectivo < 0)
281         {
282             TempData["Error"] = "El monto fisico debe ser mayor o igual a cero.";
283             return RedirectToAction(nameof(Cierre));
284         }
285
286         var pRes = new SqlParameter("@Resultado", SqlDbType.Int) { Direction = ParameterDirection.Output };
287         var pMsg = new SqlParameter("@Mensaje", SqlDbType.VarChar, 255) { Direction = ParameterDirection.Output };
288
289         await _context.Database.ExecuteSqlRawAsync(
290             @"EXEC SP_CIE_RealizarCierre
291             @IdCedula_Usuario, @Monto_Fisico_Efectivo,
292             @Resultado OUTPUT, @Mensaje OUTPUT",
293             new SqlParameter("@IdCedula_Usuario", IdAdmin),
294             new SqlParameter("@Monto_Fisico_Efectivo", Monto_Fisico_Efectivo),
295             pRes, pMsg);
296
297         int resultado = (int)pRes.Value;
298         string mensaje = pMsg.Value?.ToString() ?? "";
299
300         if (resultado == 0) TempData["Exito"] = mensaje;
301         else TempData["Error"] = $"{[resultado]} {mensaje}";
302     }
303
304     catch (SqlException ex) when (ex.Number == 2601 || ex.Number == 2627)
305     {
306         TempData["Error"] = "Ya existe un cierre de caja registrado hoy para este usuario. No se puede hacer dos cierres el mismo día.";
307     }
308
309     catch (Exception ex)
310     {
311         TempData["Error"] = ex.Message;
312     }
313
314     return RedirectToAction(nameof(Cierre));
315 }

```

Fuente: Elaboración propia

Método RealizarCierre (decimal Monto_Fisico_Efectivo) que valida que el monto no sea negativo, ejecuta el SP_Cierre_RealizarCierre con parámetros de salida, interpreta el resultado para mostrar un mensaje en TempData y captura excepciones SQL específicas (cierres duplicados) y generales.

Figura 75

Entradas caja chica

```

[HttpPost]
[ValidateAntiForgeryToken]
0 referencias
public async Task<ActionResult> ConfigurarLimite(decimal NuevoLimite)
{
    try
    {
        if (NuevoLimite <= 0)
        {
            TempData["Error"] = "El monto límite debe ser mayor a cero.";
            return RedirectToAction(nameof(Index));
        }

        var pRes = new SqlParameter("@Resultado", SqlDbType.Int) { Direction = ParameterDirection.Output };
        var pMsg = new SqlParameter("@Mensaje", SqlDbType.VarChar, 255) { Direction = ParameterDirection.Output };

        await _context.Database.ExecuteSqlRawAsync(
            @"EXEC SP_CAJ_ConfigurarLimite
            @IdCaja, @NuevoLimite, @IdCedula_Usuario,
            @Resultado OUTPUT, @Mensaje OUTPUT",
            new SqlParameter("@IdCaja", IdCajaActiva),
            new SqlParameter("@NuevoLimite", NuevoLimite),
            new SqlParameter("@IdCedula_Usuario", IdUsuarioActual),
            pRes, pMsg);

        int resultado = (int)pRes.Value;
        string mensaje = pMsg.Value?.ToString() ?? "";
        if (resultado == 0) TempData["Exito"] = mensaje;
        else TempData["Error"] = $"{[resultado]} {mensaje}";
    }
    catch (Exception ex) { TempData["Error"] = ex.Message; }
    return RedirectToAction(nameof(Index));
}

```

Fuente: Elaboración propia

En el controlador de caja chica se creó un método *ConfigurarLimite* que valida que el nuevo límite sea mayor que cero, ejecuta SP_CAJ_ConfigurarLimite con parámetros y salidas y muestra el mensaje de éxito o error en TempData antes de redirigir a Index.

Programación/salidas

Figura 76

Salidas Contabilidad

```

0 referencias
public IActionResult Historial(int id, DateTime? fechaInicio, DateTime? fechaFin)
{
    RecalcularBalanceCuentas();
    var infoCuenta = new DataTable();
    var movimientos = new DataTable();
    try
    {
        using var con = new SqlConnection(_conn);
        using var cmd = new SqlCommand("SP_CON_HistorialMovimientos", con) { CommandType = CommandType.StoredProcedure };
        cmd.Parameters.AddWithValue("@IdCuenta", id);
        cmd.Parameters.AddWithValue("@FechaInicio", fechaInicio.HasValue ? (object)fechaInicio.Value.Date : DBNull.Value);
        cmd.Parameters.AddWithValue("@FechaFin", fechaFin.HasValue ? (object)fechaFin.Value.Date : DBNull.Value);
        con.Open();
        using var reader = cmd.ExecuteReader();
        infoCuenta.Load(reader);
        if (reader.NextResult()) movimientos.Load(reader);
    }
    catch { }
    if (infoCuenta.Rows.Count == 0) infoCuenta = ObtenerInfoCuentaDirecto(id);
    movimientos = ObtenerHistorialUnificado(id, fechaInicio, fechaFin);
    ViewBag.InfoCuenta = infoCuenta;
    ViewBag.Movimientos = movimientos;
    ViewBag.IdCuenta = id;
    return View();
}

```

Fuente: Elaboración propia

En este apartado del código se muestra parte del código usado para generar las salidas del módulo de contabilidad. Aquí se validan y procesan las cuentas, naturaleza y tipos que permiten organizar las cuentas por módulo y registrar nuevas cuando se necesite. Se trabajó en el *back-end* con Visual Studio y se emplearon utilidades que facilitaron la entrada y salida de datos.

Figura 77
Salidas factura

```

34 public IActionResult Index(int? idCliente, DateTime? desde, DateTime? hasta, string? estado)
35 {
36     var facturas = new DataTable();
37     using (var con = new SqlConnection(cn))
38     {
39         con.Open();
40         var sql = @"
41             SELECT F.IdFactura, F.Num_Factura_Electronica, F.Fecha,
42                   F.Subtotal, F.Impuesto_IVA, F.Total_Colones, F.Activo,
43                   F.IdCliente, P.Nombre + ' ' + P.Apellido AS NombreCliente,
44                   P.IdCedula AS CedulaCliente,
45                   EF.Description AS Estado, F.IdEstado_Factura,
46                   F.IdCedula_Usuario, U.Username,
47                   ISNULL(MP.Description, '-') AS MetodoPago,
48                   ISNULL(F.Tipo_Pago, 'Contado') AS TipoPago
49             FROM Factura F
50             INNER JOIN Cliente C ON C.IdCliente = F.IdCliente
51             INNER JOIN Persona P ON P.IdCedula = C.IdCedula
52             INNER JOIN Estado_Factura EF ON EF.IdEstado_Factura = F.IdEstado_Factura
53             INNER JOIN Usuario U ON U.IdCedula = F.IdCedula_Usuario
54             LEFT JOIN Metodo_Pago MP ON MP.IdMetodo_Pago = F.IdMetodo_Pago
55             WHERE 1=1
56                   AND (@cli IS NULL OR F.IdCliente = @cli)
57                   AND (@desde IS NULL OR F.Fecha >= @desde)
58                   AND (@hasta IS NULL OR F.Fecha <= DATEADD(DAY, 1, @hasta))
59                   AND (@est IS NULL OR EF.Description = @est)
60             ORDER BY F.Fecha DESC, F.IdFactura DESC;";
61         using var cmd = new SqlCommand(sql, con);
62         cmd.Parameters.AddWithValue("@cli", (object?)idCliente ?? DBNull.Value);
63         cmd.Parameters.AddWithValue("@desde", (object?)desde ?? DBNull.Value);
64         cmd.Parameters.AddWithValue("@hasta", (object?)hasta ?? DBNull.Value);
65         cmd.Parameters.AddWithValue("@est", (object?)estado ?? DBNull.Value);
66         using var da = new SqlDataAdapter(cmd);
67         da.Fill(facturas);
68     }
69
70     decimal totalVendido = 0, totalIVA = 0;
71     int countPagadas = 0, countCredito = 0, countAnuladas = 0;
72     foreach (DataRow r in facturas.Rows)
73     {
74         if (Convert.ToBoolean(r["Activo"]))
75         {
76             totalVendido += Convert.ToDecimal(r["Total_Colones"]);
77             totalIVA += Convert.ToDecimal(r["Impuesto_IVA"]);
78             var e = r["Estado"]?.ToString() ?? "";
79             if (e == "Pagada") countPagadas++;
80             if (e == "Credito") countCredito++;
81         }
82         else countAnuladas++;
83     }

```

Fuente: Elaboración propia

En el controlador de factura se creó un método que recibe los datos de la factura (cliente, método de pago) y valida las entradas. Los errores se reportan mediante TempData y se redirige para corregirlos. Esto asegura que solo se procesen facturas con datos válidos.

Figura 78
Salidas presupuesto

```

1  @reference
2  public async Task<ActionResult> Index(int? idPresupuesto)
3  {
4      ViewBag.Periodos = await ObtenerPeriodos();
5      ViewBag.Comparativo = new DataTable();
6      ViewBag.Global = new DataTable();
7
8      if (idPresupuesto.HasValue)
9      {
10         var (comparativo, global) = await EjecutarSPDoble(
11             "SP_PRE_CompararPresupuesto",
12             new Dictionary<string, object> { { "@IdPresupuesto" = idPresupuesto.Value } });
13
14         ViewBag.Comparativo = comparativo;
15         ViewBag.Global = global;
16         ViewBag.IdPresupuesto = idPresupuesto;
17         ViewBag.NombrePeriodo = await ObtenerNombrePeriodo(idPresupuesto.Value);
18     }
19
20     return View();
21 }

```

Fuente: Elaboración propia

Se creó un método que muestra la lógica para obtener períodos de presupuesto: utiliza validación de fechas, comprobación de solapamientos y creación en la base de datos con manejo de errores por duplicados. Al final se notifica éxito o error para que la persona usuaria ajuste lo necesario.

Figura 79
Salidas gastos

```

100 public ActionResult DetalleCategoria(int idCuenta, DateTime? fechaInicio, DateTime? fechaFin)
101 {
102     if (idCuenta == 0) return Json(new { error = "Cuenta inválida." });
103
104     try
105     {
106         var infoCuenta = new Dictionary<string, object>();
107         using (var con = new SqlConnection(_conn))
108         using (var cmd1 = new SqlCommand())
109         {
110             cmd1.CommandText = @"
111             SELECT CC.IdCuenta, CC.Codigo, CC.Nombre, CC.Balance_Actual,
112                   CC.NivelCuenta, TC.Descripcion AS TipoCuenta
113             FROM Catalogo_Cuenta CC
114             INNER JOIN Tipo_Cuenta_Constable TC ON TC.IdTipo_Cuenta = CC.IdTipo_Cuenta
115             WHERE CC.IdCuenta = @idC";
116             cmd1.Parameters.AddWithValue("idC", idCuenta);
117             cmd1.Open();
118             var rdr = cmd1.ExecuteReader();
119             if (!rdr.HasRows()) return Json(new { error = "Cuenta no encontrada." });
120             infoCuenta["codigo"] = rdr["Codigo"].ToString() ?? "";
121             infoCuenta["nombre"] = rdr["Nombre"].ToString() ?? "";
122             infoCuenta["balance"] = Convert.ToDouble(rdr["Balance_Actual"]);
123             infoCuenta["nivelCuenta"] = rdr["NivelCuenta"].ToString() ?? "";
124             infoCuenta["tipoCuenta"] = rdr["TipoCuenta"].ToString() ?? "";
125         }
126
127         var lista = new List<object>();
128         decimal total = 0m;
129         using (var con2 = new SqlConnection(_conn))
130         using (var cmd2 = new SqlCommand())
131         {
132             cmd2.CommandText = @"
133             SELECT MC.IdMovimiento, MC.Fecha, MC.Monto, MC.Descripcion, MC.Tipo_Transaccion,
134                   TR.Descripcion AS TipoMov
135             FROM Movimiento_Caja MC
136             INNER JOIN Tipo_Movimiento_Caja TM ON TM.IdTipo_Mov = MC.IdTipo_Mov
137             LEFT JOIN Metodo_Pago MP ON MP.IdMetodo_Pago = MC.IdMetodo_Pago
138             WHERE MC.IdCuenta = @id AND MC.Fecha >= @f1 AND MC.Fecha <= @f2
139             AND ISNULL(TM.Aplica_Cuenta, 0) = 1
140             AND ISNULL(TM.Aplica_Caja_Chique, 0) = 0
141             AND (@f1 IS NULL OR MC.Fecha >= DATETIME(@f1, @fz))
142             ORDER BY MC.Fecha DESC";
143             cmd2.Parameters.AddWithValue("idC", idCuenta);
144             cmd2.Parameters.AddWithValue("f1", fechaInicio.HasValue ? (object)fechaInicio.Value.Date : DBNull.Value);
145             cmd2.Parameters.AddWithValue("f2", fechaFin.HasValue ? (object)fechaFin.Value.Date : DBNull.Value);
146             cmd2.Open();
147             using (var rdr = cmd2.ExecuteReader())
148             while (rdr.HasRows())
149             {
150                 var monto = Convert.ToDouble(rdr["Monto"]);
151                 total += monto;
152                 lista.Add(new
153                 {
154                     idMovimiento = Convert.ToInt32(rdr["IdMovimiento"]),
155                     fecha = Convert.ToDateTime(rdr["Fecha"]).ToString("dd/MM/yyyy HH:mm"),
156                     descripcion = rdr["Descripcion"].ToString() ?? "",
157                     tipoTransaccion = rdr["Tipo_Transaccion"].ToString() ?? "",
158                     metodo = rdr["Metodo"].ToString() ?? "",
159                     tipoMov = rdr["TipoMov"].ToString() ?? "",
160                     monto
161                 });
162             }
163         }
164     }
165 }

```

Fuente: Elaboración propia

En esta sección de código del controlador de gastos se muestra el flujo para registrar gastos: validación de cuenta, método y montos, ejecución del procedimiento almacenado y captura del mensaje o ID de salida para informar al usuario; así se controla la integridad y trazabilidad del registro.

Figura 80*Salidas cuentas cobrar*

```

31 1. Consultar
32 public ActionResult Index(int? idCliente, string? estado, int diasAlerta = 30)
33 {
34     string est = (estado ?? "").ToLower();
35     bool soloAmuladas = est == "amulada";
36     bool soloActivas = est == "pendiente" || est == "vencida" || est == "pagada";
37
38     DataTable cxc = soloAmuladas
39         ? ConsultarCxcDirecto(idCliente, incluirAmuladas: true, soloAmuladas: true)
40         : ConsultarCxcDirecto(idCliente, incluirAmuladas: !soloActivas, soloAmuladas: false);
41
42     decimal totalDeudado = 0m, totalPendiente = 0m;
43     int countVencidas = 0, countProximas = 0, countPagadas = 0, countAmuladas = 0;
44     var hoy = DateTime.Today;
45
46     var dtMpi = ConsultarCxcDirecto(idCliente, incluirAmuladas: true, soloAmuladas: false);
47     foreach (DataRow r in dtMpi.Rows)
48     {
49         bool act = r.Table.Columns.Contains("Active") && r["Active"] != DBNull.Value && Convert.ToBoolean(r["Active"]);
50         decimal monto = ToDecimal(r, "Monto_Total");
51         decimal saldo = ToDecimal(r, "Saldo_Pendiente");
52         DateTime venc = ToDate(r, "Fecha_Vencimiento");
53
54         if (!act) { countAmuladas++; continue; }
55         totalDeudado += monto;
56         totalPendiente += saldo;
57         if (saldo == 0) countPagadas++;
58         else if (venc < hoy) countVencidas++;
59         else if ((venc - hoy).Days <= diasAlerta) countProximas++;
60     }
61
62     if (!string.IsNullOrWhiteSpace(estado) && !soloAmuladas)
63     {
64         var rowsFiltered = cxc.AsEnumerable().Where(r =>
65         {
66             decimal saldo = ToDecimal(r, "Saldo_Pendiente");
67             DateTime venc = ToDate(r, "Fecha_Vencimiento");
68             return estado.ToLower() switch
69             {
70                 "pendiente" => saldo > 0 && venc >= hoy,
71                 "vencida" => saldo > 0 && venc < hoy,
72                 "pagada" => saldo <= 0,
73                 _ => true
74             };
75         });
76         var filtered = cxc.Clone();
77         foreach (var r in rowsFiltered) filtered.ImportRow(r);
78         cxc = filtered;
79     }
80
81     ViewBag.CuentasCobrar = cxc;
82     ViewBag.Clientes = DBContext.Clientes();
83     ViewBag.DiasAlerta = diasAlerta;
84     ViewBag.IdCliente = idCliente;
85     ViewBag.EstadoFiltro = estado ?? "";
86     ViewBag.TotalDeudado = totalDeudado;
87     ViewBag.TotalPendiente = totalPendiente;
88     ViewBag.CountVencidas = countVencidas;
89     ViewBag.CountProximas = countProximas;
90 }

```

Fuente: Elaboración propia

Se muestra la consulta y el procesamiento de las cuentas por cobrar a los clientes: carga de facturas, cálculo de totales, clasificación en pendientes, vencidas o pagadas y preparación de datos para la vista, lo que permite filtrar por estado y fechas.

Figura 81*Salidas cuentas pagar*

```

29  public IActionResult Index(int? idProveedor, string? estado, int diasAlerta = 30)
30  {
31      string est = estado?.ToLower() ?? "";
32      bool soloAnuladas = est == "anulada";
33      bool soloActivas = est == "pendiente" || est == "vencida" || est == "pagada";
34      DataTable cxp = soloAnuladas
35          ? ConsultarCxpDirecta(idProveedor, diasAlerta, incluirAnuladas: true, soloAnuladas: true)
36          : ConsultarCxpDirecta(idProveedor, diasAlerta, incluirAnuladas: soloActivas, soloAnuladas: false);
37
38      decimal totalAdudado = 0m, totalPendiente = 0m;
39      int countVencidas = 0, countProximas = 0, countPagadas = 0, countAnuladas = 0;
40      var hoy = DateTime.Today;
41      var dtCxp = ConsultarCxpDirecta(idProveedor, diasAlerta, incluirAnuladas: true, soloAnuladas: false);
42      foreach (DataRow r in dtCxp.Rows)
43      {
44          bool act = r.Table.Columns.Contains("Active") && r["Active"] != DBNull.Value && Convert.ToBoolean(r["Active"]);
45          decimal monto = ToDecimal(r, "Monto_Total");
46          decimal saldo = ToDecimal(r, "Saldo_Pendiente");
47          DateTime venc = ToDate(r, "Fecha_Vencimiento");
48          if (act) { countAnuladas++; continue; }
49          totalAdudado += monto; totalPendiente += saldo;
50          if (saldo <= 0) countPagadas++;
51          else if (venc < hoy) countVencidas++;
52          else if ((venc - hoy).Days <= diasAlerta) countProximas++;
53      }
54
55      if (!string.IsNullOrEmpty(estado) && !soloAnuladas)
56      {
57          var rowsFiltered = cxp.AsEnumerable().Where(r =>
58          {
59              decimal saldo = ToDecimal(r, "Saldo_Pendiente");
60              DateTime venc = ToDate(r, "Fecha_Vencimiento");
61              return estado.ToLower() switch
62              {
63                  "pendiente" => saldo > 0 && venc >= hoy,
64                  "vencida" => saldo > 0 && venc < hoy,
65                  "pagada" => saldo <= 0,
66                  _ => true
67              };
68          });
69          var filtered = cxp.Clone();
70          foreach (var F in rowsFiltered) filtered.ImportRow(F);
71          cxp = filtered;
72      }
73
74      ViewBag.CuentasPagar = cxp;
75      ViewBag.Proveedores = ObtenerProveedores();
76      ViewBag.DiasAlerta = diasAlerta;
77      ViewBag.IdProveedor = idProveedor;
78      ViewBag.EstadoFiltro = estado ?? "";
79      ViewBag.TotalAdudado = totalAdudado;
80      ViewBag.TotalPendiente = totalPendiente;
81      ViewBag.CountVencidas = countVencidas;
82      ViewBag.CountProximas = countProximas;
83      ViewBag.CountPagadas = countPagadas;
84      ViewBag.CountAnuladas = countAnuladas;

```

Fuente: Elaboración propia

Similar al módulo de cuentas por cobrar, aquí se cargan las CxP de proveedores, se calculan totales (pendientes, vencidas, pagadas y anuladas) y se filtran según criterios para mostrar la información relevante al usuario.

Figura 82*Salidas cierre de caja*

```

public async Task<ActionResult> DetalleCierre(int idCierre)
{
    try
    {
        var cierre = await EjecutarConsulta(
            @"SELECT IdCierre, Fecha_Cierre, Monto_Total_Sistema, Monto_Total_Fisico, Monto_Diferencia
            FROM CierreCaja WHERE IdCierre = @Id",
            new SqlParameter("@Id", idCierre));

        if (cierre.Rows.Count == 0) return NotFound();
        var r = cierre.Rows[0];
        var #Cierre = Convert.ToDateTime(r["Fecha_Cierre"]);
        var #Dia = #Cierre.Date;

        var porMetodo = await EjecutarConsulta(
            @"WITH Ingresos_dia AS (
                SELECT MC.IdMetodo_Pago, MC.Monto
                FROM MovimientoCaja MC
                WHERE MC.Activo = 1 AND MC.Tipo_Transaccion = 'Ingreso'
                AND CAST(MC.Fecha AS DATE) = @#
            ) UNION ALL
            SELECT F.IdMetodo_Pago, F.Total_Colones
            FROM Factura F
            WHERE F.Activo = 1
            AND ISNULL(F.Tipo_Pago, 'Contado') = 'Contado'
            AND CAST(F.Fecha AS DATE) = @#
            )
            SELECT MP.Descripcion AS Metodo,
                COUNT(*) AS Cantidad,
                ISNULL(SUM(I.Monto), 0) AS Total
            FROM Ingresos_dia I
            INNER JOIN Metodo_Pago MP ON MP.IdMetodo_Pago = I.IdMetodo_Pago
            GROUP BY MP.Descripcion
            ORDER BY MP.Descripcion";
            new SqlParameter("@#", #Dia));

        DataTable porGasto = new DataTable();
        try
        {
            porGasto = await EjecutarConsulta(
                @"SELECT ISNULL(CC.Nombre, 'Sin categoria') AS Categoria,
                COUNT(*) AS Cantidad,
                ISNULL(SUM(MC.Monto), 0) AS Total
            FROM MovimientoCaja MC
            LEFT JOIN Catalogo_Cuenta CC ON CC.IdCuenta = MC.IdCuenta
            WHERE MC.Activo = 1 AND MC.Tipo_Transaccion = 'Egreso'
            AND CAST(MC.Fecha AS DATE) = @#
            GROUP BY CC.Nombre
            ORDER BY 3 DESC";
                new SqlParameter("@#", #Dia));
        }
        catch { }
    }
}

```

Fuente: Elaboración propia

Se detalla el proceso de cierre de consulta del cierre, los totales del sistema y físico y el desglose por categorías y formas de pago. La información se prepara en tablas para el reporte y la revisión del cierre.

Figura 83*Salidas caja chica*

```

[HttpGet]
[Route("api/Detail/{id}")]
public async Task<ActionResult> DetalleMovimiento(int id)
{
    try
    {
        var dt = await EjecutarConsulta(
            @"SELECT MC.IdMovimiento, MC.Fecha, MC.Monto, MC.Descripcion, MC.Tipo_Transaccion,
            MC.IdCuenta, MC.IdMetodo_Pago, MC.IdTipo_Mov,
            CC.Codigo, CC.Nombre AS Cuenta, CC.Naturaleza,
            MP.Descripcion AS Metodo,
            TM.Descripcion AS TipoMov
            FROM Movimiento_Caja MC
            INNER JOIN Catalogo_Cuenta CC ON CC.IdCuenta = MC.IdCuenta
            INNER JOIN Metodo_Pago MP ON MP.IdMetodo_Pago = MC.IdMetodo_Pago
            INNER JOIN Tipo_Movimiento_Caja TM ON TM.IdTipo_Mov = MC.IdTipo_Mov
            WHERE MC.IdMovimiento = @id",
            new SqlParameter("@id", id));

        if (dt.Rows.Count == 0) return NotFound();
        var r = dt.Rows[0];
        return Json(new
        {
            idMovimiento = Convert.ToInt32(r["IdMovimiento"]),
            fecha = Convert.ToDateTime(r["Fecha"]).ToString("dd/MM/yyyy HH:mm"),
            monto = Convert.ToDecimal(r["Monto"]),
            descripcion = r["Descripcion"]?.ToString(),
            tipoTransaccion = r["Tipo_Transaccion"]?.ToString(),
            idCuenta = Convert.ToInt32(r["IdCuenta"]),
            codigo = r["Codigo"]?.ToString(),
            cuenta = r["Cuenta"]?.ToString(),
            naturaleza = r["Naturaleza"]?.ToString(),
            idMetodo = Convert.ToInt32(r["IdMetodo_Pago"]),
            metodo = r["Metodo"]?.ToString(),
            idTipoMov = Convert.ToInt32(r["IdTipo_Mov"]),
            tipoMov = r["TipoMov"]?.ToString()
        });
    }
    catch (Exception ex) { return Json(new { error = ex.Message }); }
}

```

Fuente: Elaboración propia

Se muestra la configuración y las operaciones de caja chica: ajuste de límites, registro de movimientos y ejecución de SP para aplicar cambios; los resultados se devuelven en TempData para notificar éxito o errores.

Programación/procesos

Figura 84*Procesos caja chica*

	IdCaja	Monto_Limite	Saldo_Actual	Ultima_Actualizacion	Activo
1	1	900000.00	830000.00	2026-06-02 11:28:46.510	1

Fuente: Elaboración propia

Se muestra el proceso de caja chica. Se pueden visualizar las sentencias SQL y los procedimientos que validan límites, actualizan saldos y registran movimientos. Al ingresar la información, el sistema ejecuta SP que devuelven mensajes y códigos, lo que permite controlar errores y confirmar el registro.

Figura 85*Procesos factura*

6	FAC-20260526-0003	7	118360977	4	2026-05-26 22:17:23.850	8400.00	1092.00	9492.00	0
7	FAC-20260526-0004	4	118360977	4	2026-05-26 22:23:23.553	52500.00	6825.00	59325.00	0
8	FAC-20260527-0001	4	118360977	2	2026-05-27 16:24:28.290	52500.00	6825.00	59325.00	1
9	FAC-20260527-0002	2	118360977	2	2026-05-27 16:26:43.097	440000.00	57200.00	497200.00	1
10	FAC-20260528-0001	4	118360977	3	2026-05-28 22:39:05.397	37500.00	4875.00	42375.00	1

Fuente: Elaboración propia

Proceso de facturación con las consultas y comandos SQL que se ejecutan al crear una factura, inserciones y validaciones de crédito. Las imágenes muestran cómo se reciben los datos, se validan y se guardan en la base de datos.

Figura 86*Procesos cierre de caja*

Results

Messages

	IdCierre	IdCedula_Usuario	Fecha_Cierre	Monto_Total_Sistema	Monto_Total_Fisico	Monto_Diferencia	Activo
1	1	333333333	2026-05-14 22:35:51.010	0.00	100000.00	100000.00	0
2	2	322760977	2026-05-21 10:55:57.467	200000.00	100000.00	-100000.00	0
3	3	333333333	2026-05-29 22:54:49.050	0.00	0.00	0.00	1

Fuente: Elaboración propia

Proceso de cierre de caja, donde al ingresar el monto físico se ejecutan procedimientos que comparan los totales del sistema y el físico, calculan las diferencias y registran el cierre. Las consultas SQL y los parámetros de salida permiten auditar el resultado y evitar cierres duplicados.

Figura 87*Procesos cuentas por cobrar*

Results		Messages						
	IdCuenta_Por_Cobrar	IdFactura	Monto_Total	Saldo_Pendiente	Fecha_Vencimiento	IdCedula_Usuario_Reg	Activo	Numero_Plazos
1	1	13	42375.00	42375.00	2026-06-27	118360977	1	1
2	2	16	124300.00	124300.00	2026-08-29	118360977	0	3
3	3	23	42375.00	0.00	2026-07-31	118360977	0	2

Fuente: Elaboración propia

En este bloque se muestra el flujo de cuentas por cobrar, las consultas para cargar facturas, el cálculo de saldos pendientes, la clasificación en vencidas o pagadas y los filtros por fechas. Las sentencias SQL preparan los DataTables que luego se presentan en la interfaz.

Figura 88*Procesos catálogo contable*

	IdCuenta	Codigo	Nombre	IdTipo_Cuenta	Naturaleza	Balance_Actual	Activo	Aplica_Caja_Chica	IdTipo_Mov_Default
1	1	1.1.0	papel higiénico	1	D	-20000.00	1	1	6
2	2	1.2.0	Ventas de producto	2	A	689300.00	1	0	NULL
3	3	5.0.0	Marketing	1	D	0.00	1	0	NULL
4	4	505.9.0	pruebas	3	D	-100000.00	0	0	NULL
5	5	1.1.10	Pago de Agua	1	D	-10000.00	1	0	NULL
6	6	505.0.9	Lapiceros	1	D	-50000.00	1	1	6
7	7	3.0.1	Pago a proveedores	1	D	-226000.00	1	0	NULL
8	8	4.9.9	Papel	1	D	0.00	1	1	6
9	9	1.1.01	Materiales	1	D	0.00	1	1	6
10	10	10.0.9	Pago de Luz	1	D	0.00	1	0	NULL

Fuente: Elaboración propia

En la Figura 88 se logra ver el proceso del catálogo contable al insertar o actualizar cuentas, validaciones de código y nombre (caja chica, CxP, CxC). Se pueden visualizar las consultas y procedimientos que mantienen el catálogo ordenado.

Figura 89*Procesos cuentas por pagar*

	IdCuenta_Por_Pagar	IdProveedor	IdCedula_Usuario_Reg	Monto_Total	Saldo_Pendiente	Fecha_Vencimiento	Activo	Origen
1	1	1	333333333	50000.00	50000.00	2026-06-18	0	Manual
2	2	3	333333333	1000000.00	1700000.00	2026-05-20	0	Manual
3	3	1	333333333	50000.00	50000.00	2026-06-18	0	Manual
4	4	3	333333333	100000.00	100000.00	2026-06-18	0	Manual
5	5	1	333333333	150000.00	150000.00	2026-06-18	0	Manual
6	7	1	118360977	1000.00	0.00	2026-05-22	0	Manual
7	8	3	118360977	300000.00	300000.00	2026-06-21	0	Manual
8	9	1	118360977	250000.00	250000.00	2026-06-24	0	Manual
9	10	3	118360977	1000000.00	1000000.00	2026-06-24	0	Manual
10	11	1	118360977	500000.00	500000.00	2026-06-25	0	Manual

Fuente: Elaboración propia

Se muestra el proceso de cuentas por pagar de la empresa, que incluye: validación de proveedor y fechas y ejecución de SP para guardar la CxP. Las consultas SQL devuelven mensajes y estados que permiten controlar el flujo y los saldos pendientes.

Figura 90*Procesos presupuesto*

	IdPresupuesto	Nombre_Periodo	Fecha_Inicio	Fecha_Fin	Monto_Global_Mensual	IdCedula_Creador	Activo
1	1	Presupuesto Febrero	2026-02-01	2026-02-28	2000000.00	333333333	0
2	2	Presupuesto enero	2026-01-01	2026-01-30	3000000.00	333333333	0
3	3	Presupuesto Marzo	2026-03-01	2026-03-30	2000000.00	333333333	0
4	4	Presupuesto Abril	2026-04-01	2026-04-30	700000.00	333333333	0
5	5	Mayo	2026-05-01	2026-05-31	10000000.00	333333333	1
6	6	presupuesto junio	2026-06-01	2026-06-30	1000000.00	333333333	0
7	7	Presupuesto julio	2026-07-01	2026-07-30	2000000.00	118360977	0
8	8	Presupuesto Junio 2026	2026-06-01	2026-06-30	1000000.00	333333333	1

Fuente: Elaboración propia

Proceso del módulo de presupuesto que contiene validación de fechas, comprobación de solapamientos y ejecución de inserciones en la base de datos. Se detallan mediante una tabla los datos importantes al realizar un presupuesto. La empresa realiza presupuestos mensuales, por lo que dos presupuestos no pueden ser del mismo mes en el mismo año.

Programación/validaciones

Figura 91

Ingresar un ‘ tanto en el usuario como en la contraseña y tratar de ingresar

The screenshot displays the login interface for 'Ópticas Dr. Max', a 'Sistema Contable'. At the top is a logo featuring a teal circle with an orange pulse line. Below the logo, the text 'Ópticas Dr. Max' and 'Sistema Contable' are shown. A section titled 'Iniciar Sesión' is followed by a red error message: '⚠ Usuario o contraseña incorrectos.' Below this, the 'USUARIO' field contains 'Tyrón' and the 'CONTRASEÑA' field contains 'Tyrón1234567'. A teal 'Ingresar' button is positioned below the password field. A light blue informational box states: 'El sistema te llevará al inicio según tu perfil: **Administrador** ve todos los módulos, **Cajero** ve solo Facturación, Ingresos y Cierre de Caja.' The footer indicates '© 2026 - Ópticas Dr. Max'.

Fuente: Elaboración propia

La prueba fue exitosa, ya que muestra que el sistema no permite el ingreso de un usuario si el nombre y la contraseña no coinciden exactamente con los registrados en el sistema. Esto proporciona una capa de seguridad adicional al sistema, protegiendo los datos financieros de la empresa.

Figura 92

Dejar vacío el usuario y la contraseña y tratar de ingresar

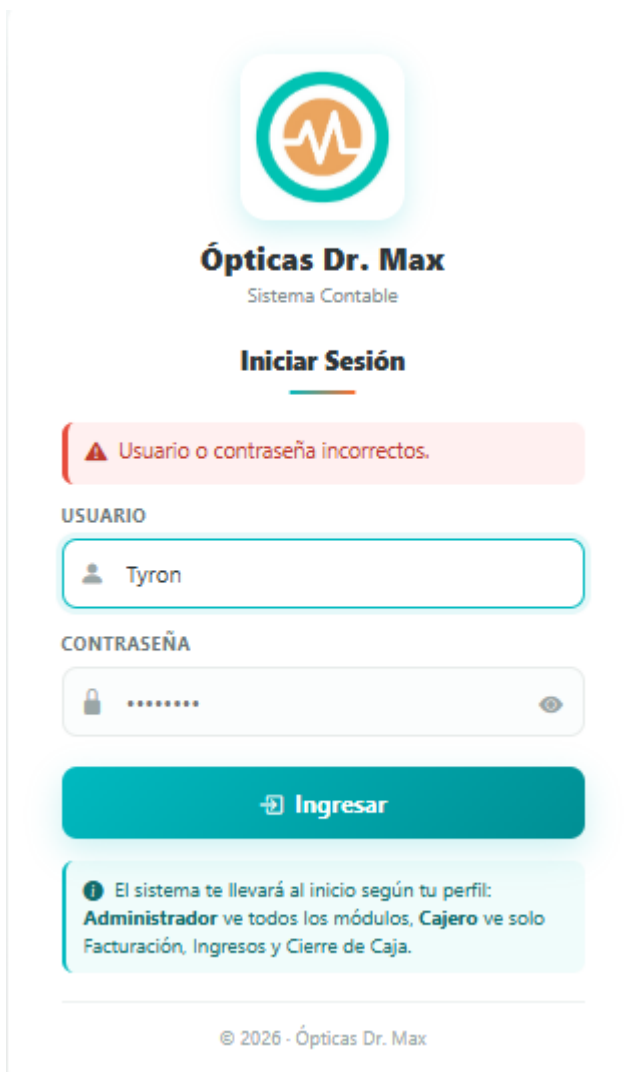
The image shows a login interface for 'Ópticas Dr. Max'. At the top is a logo with a green circle and an orange pulse line. Below it, the text 'Ópticas Dr. Max' and 'Sistema Contable' are displayed. A section titled 'Iniciar Sesión' is followed by two input fields: 'USUARIO' and 'CONTRASEÑA'. The 'USUARIO' field contains the placeholder text 'Tu nombre de usuario'. The 'CONTRASEÑA' field is empty and has a red error message box above it that says 'Completa este campo'. Below the fields is a green 'Ingresar' button. At the bottom, a light blue box contains a message: 'El sistema te llevará al inicio según tu perfil: Administrador ve todos los módulos, Cajero ve solo Facturación, Ingresos y Cierre de Caja.' The footer shows '© 2026 - Ópticas Dr. Max'.

Fuente: Elaboración propia

En esta prueba se dejaron los campos de usuario y contraseña vacíos, con el fin de probar la función del *login*. La prueba fue exitosa al no permitir el acceso a la página principal por falta de credenciales.

Figura 93

Si el usuario es un correo, no poner un @ y darle ingresar



The screenshot displays the login interface for 'Ópticas Dr. Max', a 'Sistema Contable'. At the top is a logo with a teal circle and an orange pulse line. Below the logo, the text 'Ópticas Dr. Max' and 'Sistema Contable' are shown. A teal button labeled 'Iniciar Sesión' is present. A red error message box states: '⚠ Usuario o contraseña incorrectos.' Below this are two input fields: 'USUARIO' containing 'Tyron' and 'CONTRASEÑA' with masked characters. A teal 'Ingresar' button is below the fields. A light blue information box at the bottom explains that the system will redirect to the start based on the profile: 'Administrador' sees all modules, while 'Cajero' sees only 'Facturación, Ingresos y Cierre de Caja.' The footer shows '© 2026 - Ópticas Dr. Max'.

Fuente: Elaboración propia

Se modificó el usuario Tyron y se le colocó un @gmail.com al final del nombre de usuario. La prueba fue exitosa, ya que no dejó ingresar al usuario al sistema porque este no tenía el @gmail.com al final del nombre.

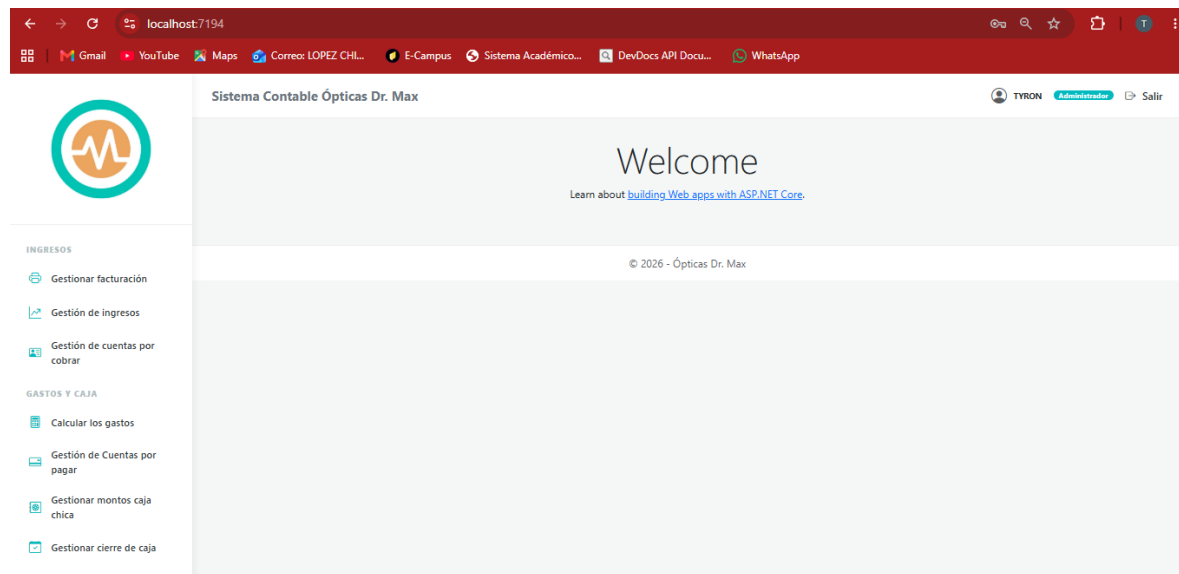
Figura 94

Poner la contraseña de un usuario y la clave de otro usuario distinto y darle ingresar

The screenshot displays the login screen for 'Ópticas Dr. Max', a 'Sistema Contable'. At the top is a logo with a teal circle and an orange pulse line. Below it, the text 'Ópticas Dr. Max' and 'Sistema Contable' are shown. A teal 'Iniciar Sesión' button is present. A red error message box states: '⚠ Usuario o contraseña incorrectos.' Below this are input fields for 'USUARIO' (containing 'Tyron') and 'CONTRASEÑA' (containing 'D1234567'). A teal 'Ingresar' button is at the bottom. A light blue information box explains: 'El sistema te llevará al inicio según tu perfil: **Administrador** ve todos los módulos, **Cajero** ve solo Facturación, Ingresos y Cierre de Caja.' The footer shows '© 2026 - Ópticas Dr. Max'.

Fuente: Elaboración propia

Se utilizó la contraseña de otro usuario llamada Daniela, con la contraseña D1234567, en el intento de inicio de sesión del usuario Tyron. La validación del inicio de sesión no permitió el acceso al usuario debido a que la contraseña no es correcta.

Figura 95*Ingresar un usuario y contraseña válida**Fuente:* Elaboración propia

Se ingresó el usuario y la contraseña correcta, por lo tanto, el sistema permitió el ingreso a la página principal. El ingreso se realizó con el usuario llamado Tyron y la contraseña T1234567.

Figura 96*Ingresar un nuevo usuario con números en el nombre y los apellidos*

DATOS PERSONALES		CONTACTO	
CÉDULA (formato 1-2345-6789)		TELÉFONO (8 dígitos, no inicia con 0)	
000000000		88887777	
La cédula no puede empezar con 0			
NOMBRE		TIPO TELÉFONO	
		Personal	
PRIMER APELLIDO		CORREO	
		usuario@gmail.com	
SEGUNDO APELLIDO		TIPO CORREO	
		Personal	
FECHA NACIMIENTO		USUARIO	
dd/mm/aaaa		ROL	
GÉNERO		Administrador	
Masculino		USERNAME	

Fuente: Elaboración propia

El sistema contable tiene validaciones que no permiten ingresar números en campos donde solo se pueden digitar letras, por lo tanto, esta prueba resultó exitosa al no poder crear el usuario debido a la falta de datos o a datos erróneos.

Figura 97

Ingresar un nuevo usuario con fecha de hoy

The screenshot shows a user registration form with the following fields and values:

- PRIMER APELLIDO:** jbj (Correcto ✓)
- SEGUNDO APELLIDO:** (Correcto ✓)
- FECHA NACIMIENTO:** 03/06/2026 (Error: Debe ser mayor de edad (edad actual: 0 años))
- GÉNERO:** Masculino
- DIRECCIÓN:** Barrio las luisas (17/200 caracteres)
- CORREO:** tyronlopez97@gmail.com (Correo válido ✓)
- TIPO CORREO:** Personal
- USUARIO:** (Button)
- ROL:** Administrador
- USERNAME:** jknh (Username válido ✓)
- CONTRASEÑA:** (Password field with strength indicator: Contraseña Fuerte ✓)
- CONFIRMAR CONTRASEÑA:** (Empty field)

At the bottom, there are buttons for "Guardar" and "Volver", and a brightness slider.

Fuente: Elaboración propia

La validación no permite crear un usuario que no tenga más de 18 años o que sea mayor de 90 años. El sistema arroja un mensaje indicando que la fecha es errónea y señala la condición de edad al crear un usuario o editarlo.

Figura 98*Ingresar un nuevo usuario con letras en la identificación*

↑ DATOS PERSONALES

CÉDULA (formato 1-2345-6789)

 La cédula no puede empezar con 0

NOMBRE

 Correcto ✓

PRIMER APELLIDO

 Correcto ✓

SEGUNDO APELLIDO

 Correcto ✓

FECHA NACIMIENTO

 Debe ser mayor de edad (edad actual: 0 años)

📞 CONTACTO

TÉLEFONO (8 dígitos, no inicia con 0)

 Teléfono válido ✓ 9999-9999

TIPO TELÉFONO

CORREO

 Correo válido ✓

TIPO CORREO

👤 USUARIO

ROL

Fuente: Elaboración propia

La validación que se utiliza no permite ingresar letras en la identificación ni comenzar con 0. El formato de la identificación es el siguiente: x-xxxx-xxxx y este formato se basa en la cédula de identificación que se usa en Costa Rica.

Figura 99*Crear el usuario y reiniciar la contraseña*

PRIMER APELLIDO

SEGUNDO APELLIDO

FECHA NACIMIENTO

GÉNERO

DIRECCIÓN

 12/200 caracteres

PERSONAL

CORREO

 Correo válido ✓

TIPO CORREO

ROL

🔑 CREDENCIALES

USERNAME

NUEVA CONTRASEÑA (dejar vacío para no cambiar)

 Contraseña Fuerte ✓

Guardar Cambios [← Volver](#)

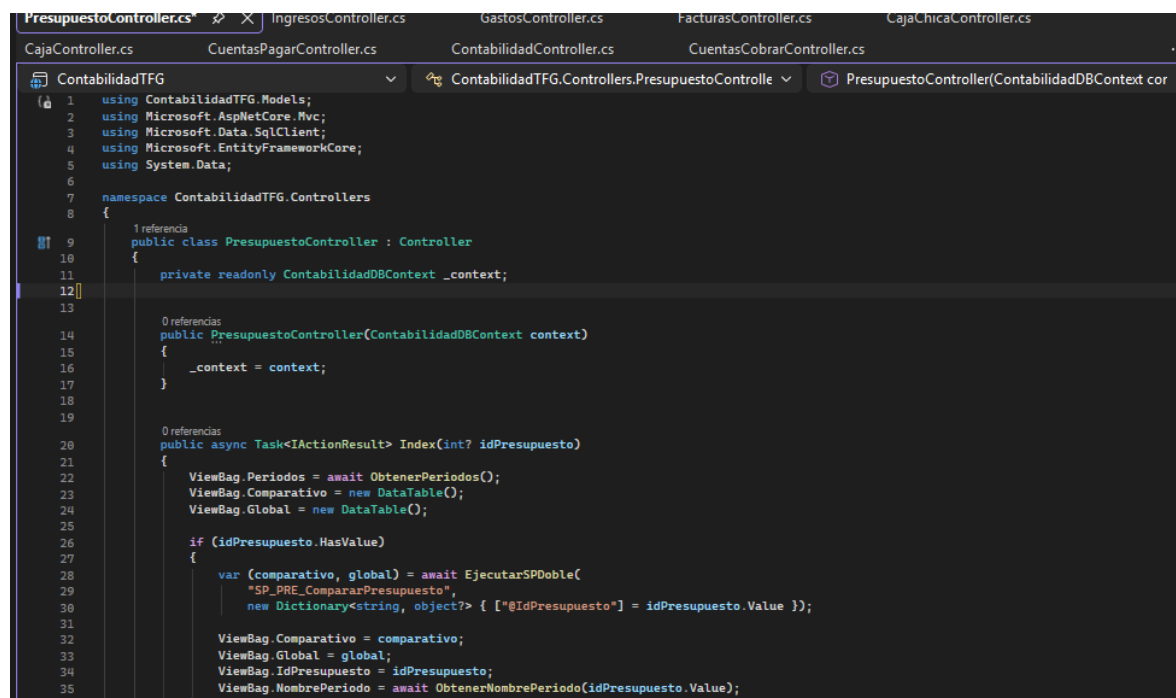
Fuente: Elaboración propia

El sistema permite crear un usuario y asignarle un rol. Luego, se colocan el nombre de usuario y la contraseña que se utilizan en el inicio de sesión de la página. Se pueden editar todos los datos del usuario, excepto la identificación. En cuanto a la contraseña, se puede modificar o dejar en blanco para que se mantenga la que ya tiene actualmente.

Programación/módulos señalados en el alcance

Figura 100

Controller Presupuesto



```

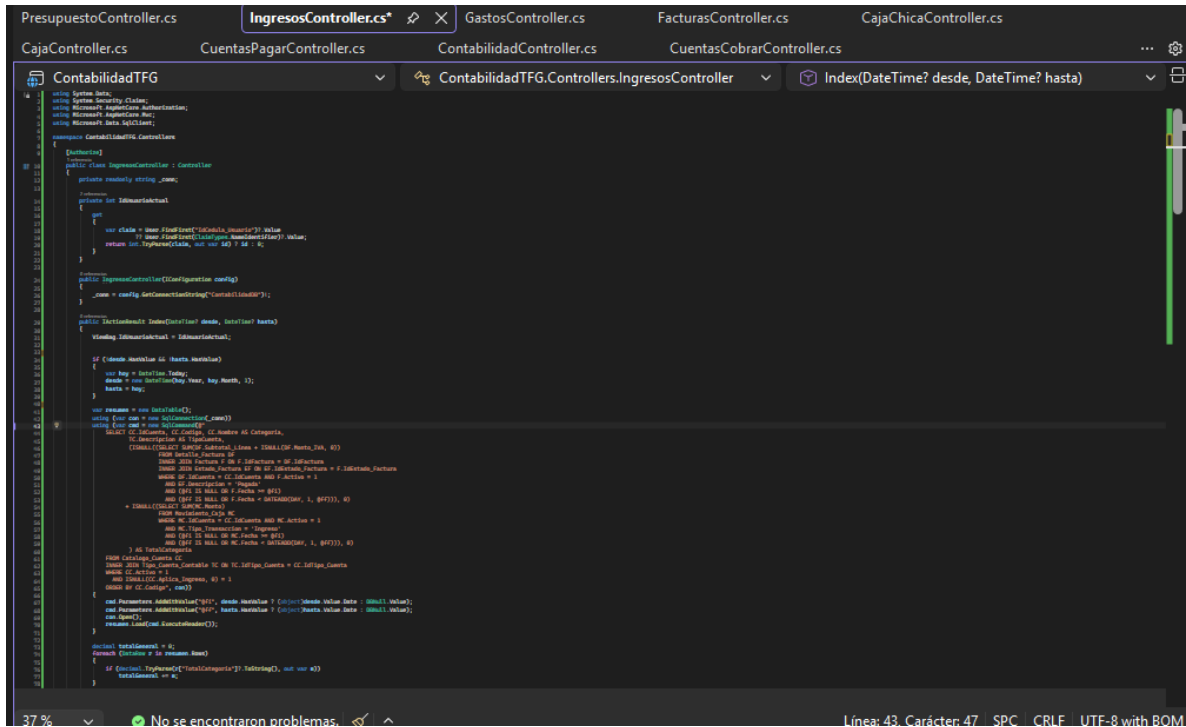
1 using ContabilidadTFG.Models;
2 using Microsoft.AspNetCore.Mvc;
3 using Microsoft.Data.SqlClient;
4 using Microsoft.EntityFrameworkCore;
5 using System.Data;
6
7 namespace ContabilidadTFG.Controllers
8 {
9     1 referencia
10     public class PresupuestoController : Controller
11     {
12         private readonly ContabilidadDbContext _context;
13
14         0 referencias
15         public PresupuestoController(ContabilidadDbContext context)
16         {
17             _context = context;
18         }
19
20         0 referencias
21         public async Task<IActionResult> Index(int? idPresupuesto)
22         {
23             ViewBag.Periodos = await ObtenerPeriodos();
24             ViewBag.Comparativo = new DataTable();
25             ViewBag.Global = new DataTable();
26
27             if (idPresupuesto.HasValue)
28             {
29                 var (comparativo, global) = await EjecutarSPDoble(
30                     "SP_PRE_CompararPresupuesto",
31                     new Dictionary<string, object?> { { "IdPresupuesto", idPresupuesto.Value } });
32
33                 ViewBag.Comparativo = comparativo;
34                 ViewBag.Global = global;
35                 ViewBag.IdPresupuesto = idPresupuesto;
36                 ViewBag.NombrePeriodo = await ObtenerNombrePeriodo(idPresupuesto.Value);
37             }
38         }
39     }
40 }

```

Fuente: Elaboración propia

En este apartado se muestra el controlador Presupuesto, el cual maneja la vista de períodos, la creación y edición de estos y la generación de proyecciones. El código usa Entity Framework y llamadas directas a procedimientos almacenados para insertar o comparar datos en la BD, además de validar fechas y evitar solapamientos. Al ingresar datos, se controlan errores con TempData y se devuelven mensajes al usuario. Asimismo, hay verificaciones para no modificar períodos con movimientos reales.

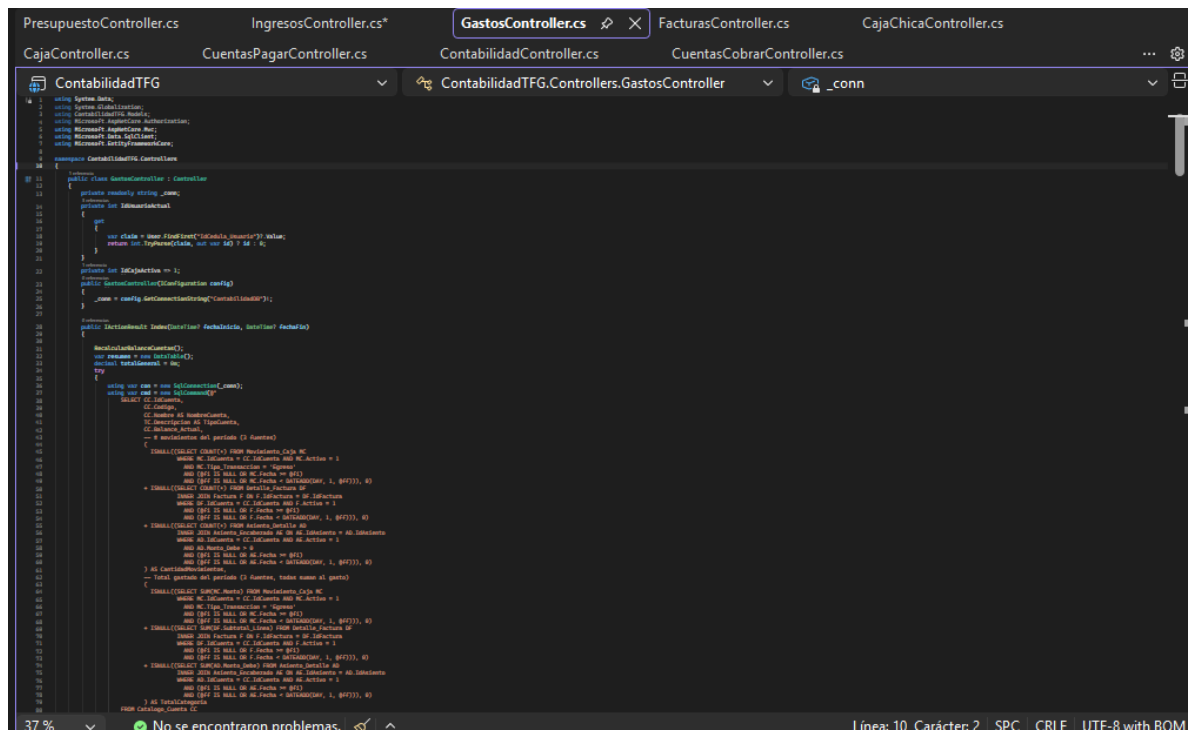
Figura 101
Controller Ingresos



Fuente: Elaboración propia

El *controller* Personas se encarga de gestionar clientes y proveedores desde el *back-end* usando EntityFramework y llamadas a procedimientos SQL. Recibe los datos del formulario, valida y crea o actualiza la persona, los teléfonos y los correos y también permite desactivar registros cuando es necesario. Al ingresar la información ejecuta SP y guarda los cambios en la BD, mostrando mensajes en TempData para informar al usuario.

Figura 102
Controller Gastos



Fuente: Elaboración propia

Controlador que maneja todo lo que se relaciona con los gastos, carga resúmenes por categoría y calcula totales del período mediante consultas SQL. Al registrar un gasto, valida el monto, la descripción y el tipo de cuenta; evita las cuentas de caja chica y llama al procedimiento almacenado que persiste el movimiento. Los resultados del procedimiento almacenado devuelven códigos y mensajes que se muestran al usuario a través de TempData y se recalculan los balances para mantener todo actualizado.

Figura 104
Controller CajaChica

PresupuestoController.cs

IngresosController.cs*

GastosController.cs

FacturasController.cs

CajaChicaController.cs*

CajaController.cs

CuentasPagarController.cs

ContabilidadController.cs

CuentasCobrarController.cs

ContabilidadTFG

ContabilidadTFG.Controllers.CajaChicaController

Index(DateTime? desde, DateTime? hasta)

```

using System;
using ContabilidadTFG.Models;
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Logging;

namespace ContabilidadTFG.Controllers
{
    public class CajaChicaController : Controller
    {
        private readonly ContabilidadTFGContext _context;
        private readonly ILogger<CajaChicaController> _logger;

        public CajaChicaController(ContabilidadTFGContext context, ILogger<CajaChicaController> logger)
        {
            _context = context;
            _logger = logger;
        }

        // GET: CajaChica
        public async Task Index(DateTime? desde, DateTime? hasta)
        {
            var query = _context.CajaChicaMovimientos;
            if (desde != null) query = query.Where(m => m.Fecha >= desde);
            if (hasta != null) query = query.Where(m => m.Fecha <= hasta);

            return View(await query.ToListAsync());
        }

        // GET: CajaChica/Details/5
        public async Task Details(int? id)
        {
            if (id == null) return NotFound();

            var cajaChicaMovimiento = await _context.CajaChicaMovimientos.FindAsync(id);
            if (cajaChicaMovimiento == null) return NotFound();

            return View(cajaChicaMovimiento);
        }

        // POST: CajaChica/Save
        [HttpPost]
        public async Task Save(CajaChicaMovimiento cajaChicaMovimiento)
        {
            if (!ModelState.IsValid) return View(cajaChicaMovimiento);

            _context.CajaChicaMovimientos.Add(cajaChicaMovimiento);
            await _context.SaveChangesAsync();

            return RedirectToAction(nameof(Index));
        }

        // DELETE: CajaChica/Delete/5
        public async Task Delete(int? id)
        {
            if (id == null) return NotFound();

            var cajaChicaMovimiento = await _context.CajaChicaMovimientos.FindAsync(id);
            if (cajaChicaMovimiento == null) return NotFound();

            _context.CajaChicaMovimientos.Remove(cajaChicaMovimiento);
            await _context.SaveChangesAsync();

            return RedirectToAction(nameof(Index));
        }
    }
}

```

37%

No se encontraron problemas.

Linea: 28, Carácter: 29

SPC

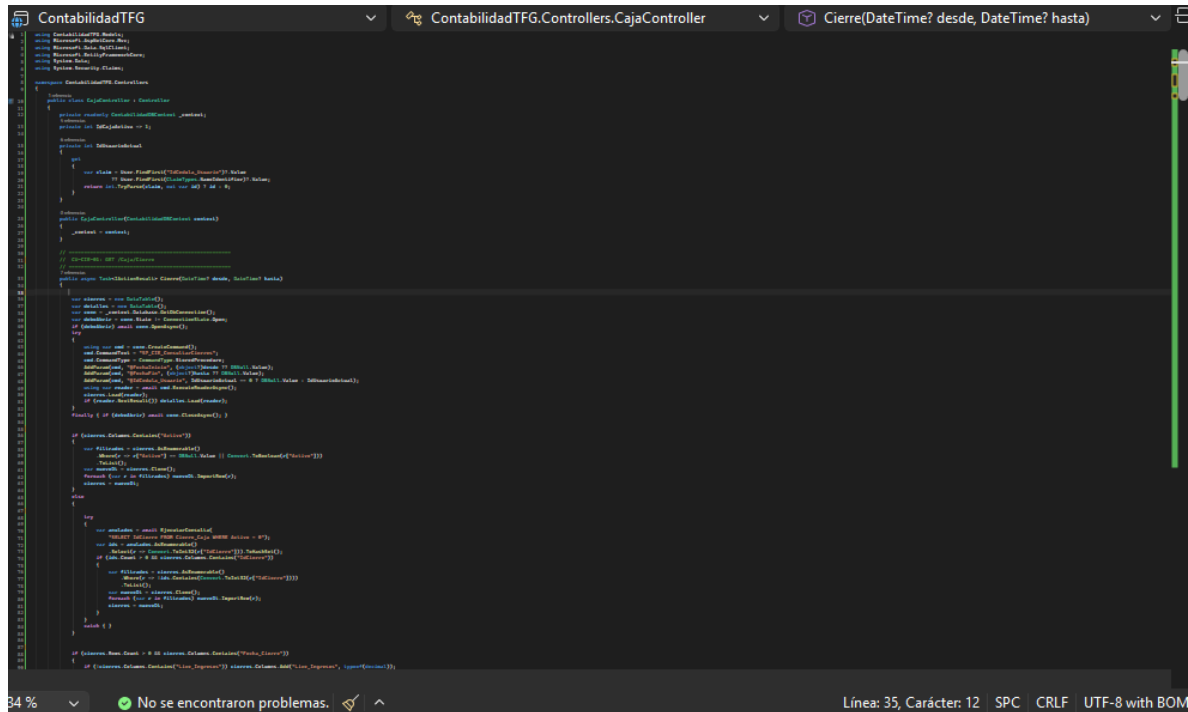
CRLF

UTF-8 with BOM

Fuente: Elaboración propia

En este *controller* se programó el módulo de caja chica, que posee varias funcionalidades, como mostrar saldos, desglose y movimientos entre fechas de forma rápida y controlada. Al ingresar un movimiento, valida el monto, la descripción, el tipo y la fecha. Además, verifica que la cuenta aplique a caja chica antes de persistir.

Figura 105
Controller CierreCaja



Fuente: Elaboración propia

En el *controller* CierreCaja se programó todo lo que se relaciona con el cierre. Cuenta con la función de realizar un cierre por día y de eliminarlo si ocurrió un error al crearlo. Además, se desglosan los detalles de los movimientos realizados en el día.

Figura 106
Controller CuentasPagar

```

Contabilidad IFG
C:\Users\Ilyro\Downloads\IFG\Contabilidad IFG\Contabilidad IFG\Controllers\CuentasPagarController.cs decimal(DataRow r, string col)

using Microsoft.AspNetCore.Mvc;
using Microsoft.AspNetCore.Mvc.Rendering;
using Microsoft.EntityFrameworkCore;
using Microsoft.Extensions.Logging;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace ContabilidadIFG.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class CuentasPagarController : ControllerBase
    {
        private readonly ApplicationDbContext _context;
        private readonly ILogger<CuentasPagarController> _logger;

        public CuentasPagarController(ApplicationDbContext context, ILogger<CuentasPagarController> logger)
        {
            _context = context;
            _logger = logger;
        }

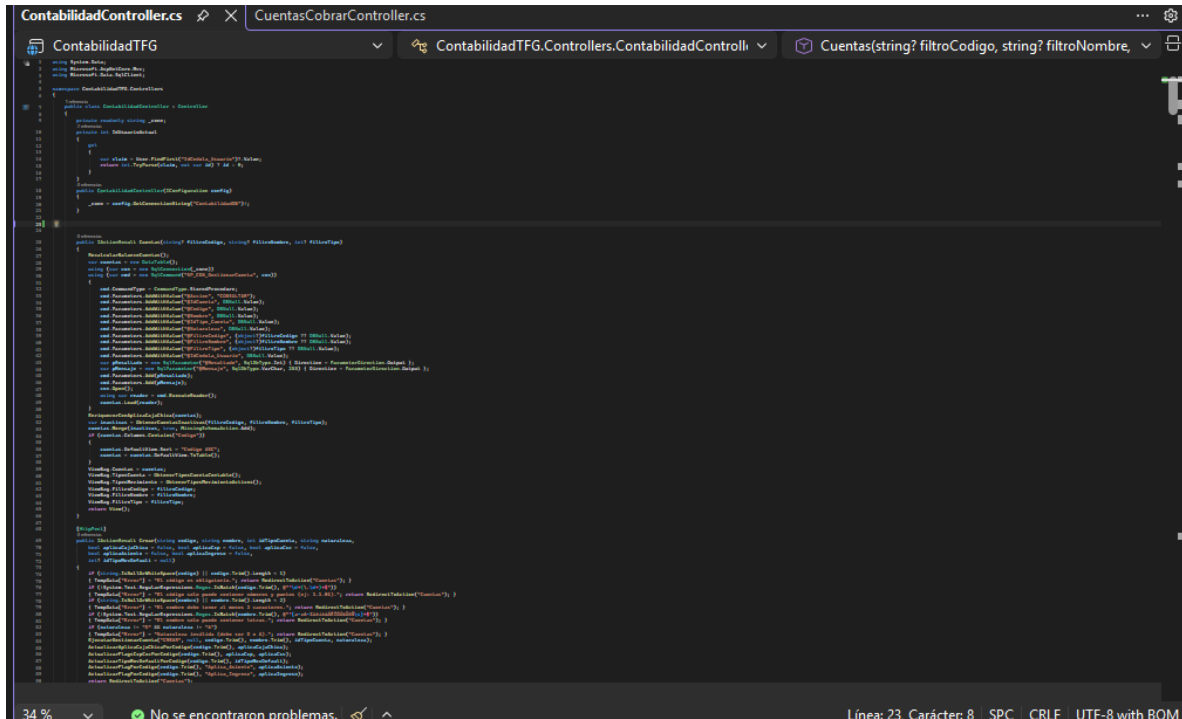
        // GET: api/CuentasPagar
        [HttpGet]
        public async Task

```

Fuente: Elaboración propia

Este *controller* se encarga de crear las cuentas por pagar de la empresa, ya sea de manera automática mediante un encargo de productos o de forma manual. De igual manera, se pueden realizar abonos a la cuenta por pagar y también se puede visualizar la cuenta pendiente o las pagadas de la compañía.

Figura 107
Controller Contabilidad



Fuente: Elaboración propia

La creación de un *controller* que maneje el libro contable y sus implicaciones es lo más fundamental al realizar un sistema contable, ya que este es la base para los demás módulos. Este módulo se encarga de crear las cuentas con sus respectivos datos: tipo de cuenta, nombre, naturaleza y a qué módulo pertenece la cuenta. Contiene la función de mostrar un historial de la cuenta, resultando en un medio útil para que un administrador pueda visualizar los distintos montos que se registraron en un mes.

Figura 108
Controller CuentasCobrar

[illegible]

Fuente: Elaboración propia

Para la elaboración correcta del código de cuentas por pagar se debe realizar primero la factura. Luego, en este apartado del código se realizan funciones como abonar, eliminar abono o ver el detalle de la cuenta, así como visualizar el total de cuotas que el cliente debe pagar.

Pruebas

Figura 109

Prueba 1 Presupuesto

Configurar Períodos Presupuestarios [← Volver al Comparativo](#)

● Presupuesto eliminado correctamente. Ese rango de fechas queda libre para crear o proyectar uno nuevo. ✕

NUEVO PERÍODO

NOMBRE DEL PERÍODO *
presupuesto julio
Correcto ✓

FECHA INICIO *
01/07/2026
Correcto ✓

FECHA FIN *
31/07/2026
Correcto ✓

MONTO GLOBAL MENSUAL (€) *
1500000
€ 1 500 000.00 ✓

DURACIÓN DEL PERÍODO
2 meses

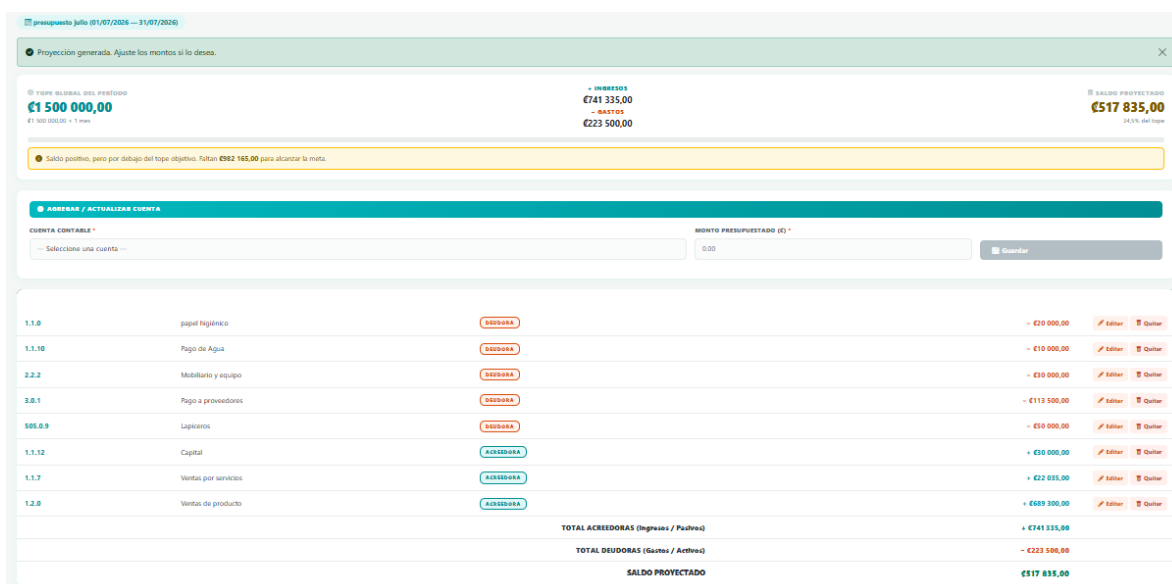
Crear Período

PERÍODOS EXISTENTES

Fuente: Elaboración propia

En esta prueba se creó un presupuesto para el mes de julio con un monto de €1.500.000. La prueba fue exitosa, ya que se creó el presupuesto. Este módulo tiene validaciones que no permiten crear un presupuesto en un mes que ya existe o poner una fecha de fin anterior a la de inicio.

Figura 110
Prueba 2 Presupuesto



Fuente: Elaboración propia

La prueba 2 consiste en realizar una reproyección del presupuesto de julio. La reproyección utiliza presupuestos anteriores para obtener el nuevo presupuesto, toma las cuentas que se utilizan y sus movimientos, con el fin de garantizar una proyección adecuada.

Figura 111
Prueba 1 Contabilidad

Nueva Cuenta Contable

X

CÓDIGO

1.1.0

Solo números y puntos. Ej. 1.1.01

NOMBRE

papel higiénico

Solo letras y espacios.

TIPO DE CUENTA

Gasto

▼

NATURALEZA

Deudora

Acreedora

¿EN QUÉ MÓDULOS APLICA ESTA CUENTA?

☒ Caja Chica

☐ Cuentas por Pagar

☐ Cuentas por Cobrar

☐ Asiento Contable (Capital, Patrimonio, Préstamos)

☐ Ingreso (módulo Ingresos)

TIPO DE MOVIMIENTO POR DEFECTO

Gasto Caja Chica

▼

Se auto-seleccionará al usar esta cuenta en Caja Chica.

Cancelar

Guardar

Fuente: Elaboración propia

La prueba 1 del módulo del libro contable consiste en crear una cuenta, la cual posee un tipo de cuenta al que pertenecerá, la naturaleza, el módulo en que se usa y la clase de movimiento que por lo general tiene.

Figura 112
Prueba 2 Contabilidad

Editar Cuenta Contable

CÓDIGO

Solo números y puntos. Ej. 1.1.01

NOMBRE

Solo letras y espacios.

TIPO DE CUENTA

NATURALEZA
☒ Deudora ☐ Acreedora

¿EN QUÉ MÓDULOS APLICA ESTA CUENTA?

☒ Caja Chica

☐ Cuentas por Pagar

☐ Cuentas por Cobrar

☐ Asiento Contable

☐ Ingreso (módulo Ingresos)

TIPO DE MOVIMIENTO POR DEFECTO

Se auto-seleccionará al usar esta cuenta en Caja Chica.

Fuente: Elaboración propia

La prueba 2 consiste en editar la cuenta que se creó con el fin de cambiar el nombre, la naturaleza, el módulo en que se utiliza y el tipo al que pertenece. Esta prueba tiene la finalidad de que el administrador, en caso de un error, pueda modificar de manera rápida el libro sin tener que borrar y volver a hacerlo.

Figura 113
Prueba 1 Cierre de Caja

Realizar Cierre de Caja

Este es el resumen del día. Revisalo y confirma el cierre. El efectivo en caja se calcula automáticamente desde los métodos marcados como "Es Efectivo".

Ingresos del día

€42 375,00

Efectivo

€42 375,00

Gastos del día

€1 000,00

Pago a proveedores

€1 000,00

Ganancia del día

€41 375,00

Efectivo físico esperado

€42 375,00

Cancelar

Confirmar Cierre

Fuente: Elaboración propia

Se realizó una prueba en el módulo de cierre de caja que consiste en realizar el cierre de caja del día 4/6/2026. La prueba fue exitosa al mostrar el detalle del cierre antes de ejecutarlo. Al oprimir el botón *confirmar cierre*, se guarda el cierre correspondiente.

Figura 114
Prueba 2 Cierre de Caja

Sistema Contable Ópticas Dr. Max

TIENDA

Realizar Cierre

Salir

Cierre de Caja

Registrar Ingreso

Realizar Cierre

El cierre de hoy ya fue realizado. Puede consultarlo en el historial al final.

RESUMEN DEL DÍA (04/06/2026)

Efectivo

€42 375,00

1 transacción(es)

TOTAL DEL DÍA

€42 375,00

TRANSACCIONES DE HOY

13:42

FAC Venta de Contacto

Efectivo

Ingreso

+ €42 375,00

Factura FAC 20260604 0002 - Rodolfo López

12:45

3.0.1 Pago a proveedores

Efectivo

Salida

- €1 000,00

hkvthv

HISTORIAL DE CIERRES

Desde

04/06/2026

Hasta

04/06/2026

Filtrar

04/06/2026

12:45

€42 375,00

€1 000,00

€41 375,00

GANANCIA

Eliminar

© 2026 - Ópticas Dr. Max

Fuente: Elaboración propia

En esta prueba se validó que no se pueda crear otro cierre de caja si ya existe uno ese día. A la vez, se puede eliminar el cierre creado para volver a realizar este proceso, lo cual se realizó de esta manera, por si ocurriera un error al finalizar las finanzas del día.

Figura 115
Prueba 1 Caja Chica

Caja Chica Configurar Límite Registrar Movimiento

MONTO ASIGNADO (LÍMITE) €900 000,00	TOTAL INGRESOS €0,00	TOTAL EGRESOS €170 000,00	SALDO ACTUAL €730 000,00 81,1% del límite - Caja sana
--	---------------------------------------	--	---

CÓMO SE CALCULA EL SALDO ACTUAL

Límite €900 000,00 - Egresos €170 000,00 + Ingresos €0,00 = €730 000,00

DESDE: dd/mm/aaaa HASTA: dd/mm/aaaa Filtrar Limpia

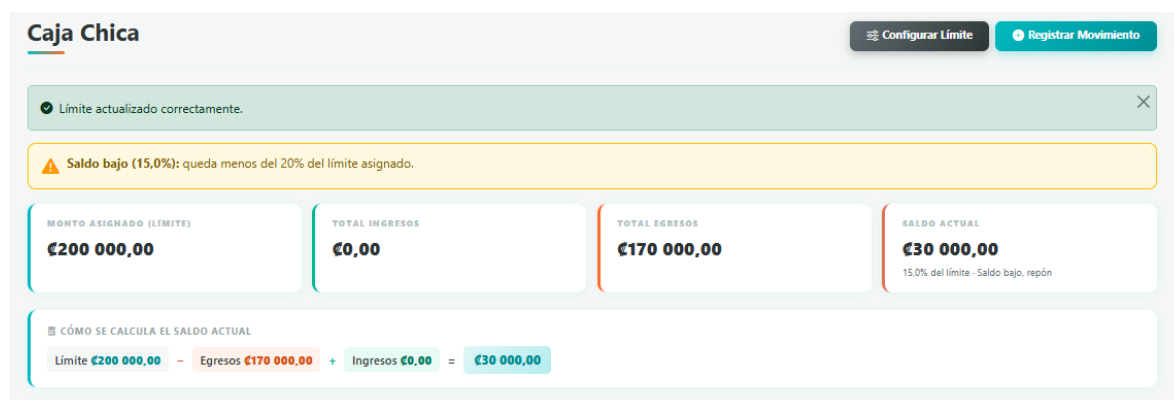
HISTORIAL DE MOVIMIENTOS

FECHA	CÓDIGO	DESCRIPCIÓN	TIPO	MONTOS	ACCIONES
04/06/2026 14:57	1.1.01	Materiales	Gasto Caja Chica	Transferencia bancaria	EGRESO - €100 000,00 Ver Editar Eliminar
01/06/2026 22:44	1.1.0	papel higiénico	Gasto Caja Chica	Efectivo	EGRESO - €20 000,00 Ver Editar Eliminar
22/05/2026 12:11	505.0.9	Lapiceros	Gasto Caja Chica	Tarjeta de crédito	EGRESO - €50 000,00 Ver Editar Eliminar

Fuente: Elaboración propia

Para esta prueba se creó un movimiento en el apartado de caja chica, con el fin de mostrar cómo se reduce la cantidad o monto que dispone la caja chica. A la vez, se pueden eliminar y editar los movimientos realizados para corregir errores que puedan suceder.

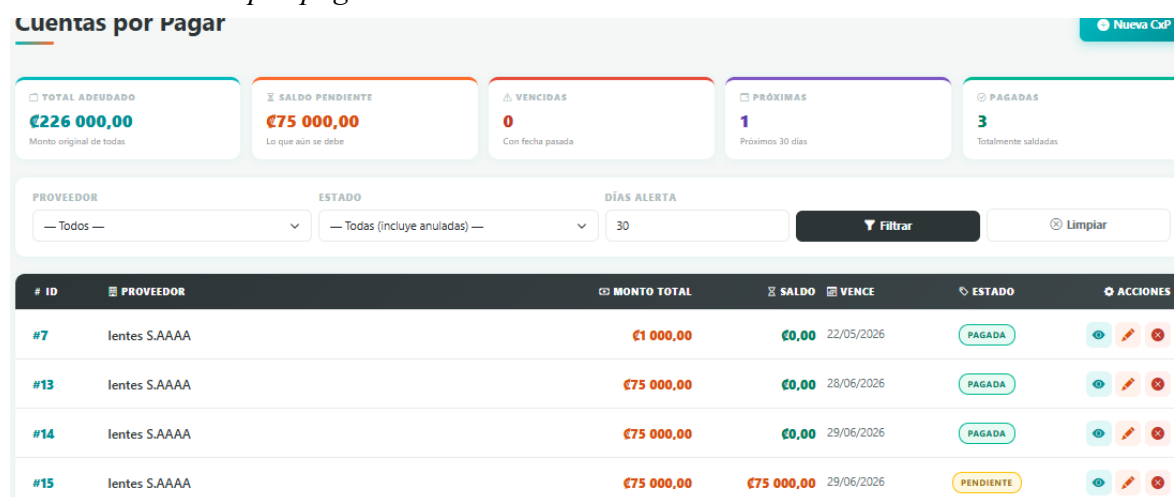
Figura 116
Prueba 2 Caja Chica



Fuente: Elaboración propia

En la prueba 2 de la caja chica se modificó el límite de la caja. Cuando se modifica el límite, se calcula cuánto queda por gastar según el nuevo límite; se muestra la operación y, en caso de un movimiento, si este es mayor al límite actual, no se agrega y aparece un mensaje de error: “monto supera límite de caja chica”.

Figura 117
Prueba 1 Cuentas por pagar



Fuente: Elaboración propia

La prueba realizada consiste en efectuar la venta de un producto cuyo *stock* está bajo, lo que genera automáticamente una cuenta por pagar. Esta cuenta se puede anular en caso de no querer dicho producto.

Figura 118

Prueba 2 Cuentas por pagar

The screenshot displays a web interface for recording a payment. It is divided into two main sections: 'DATOS DEL PAGO' (Payment Data) on the left and 'DATOS DE LA CXP' (Accounts Payable Data) on the right.

DATOS DEL PAGO:

- MONTO DEL ABONO:** A text input field contains '75000,00'. Below it, a green status indicator says 'Monto válido.' A large teal box displays the amount '€ 75 000,00'.
- MÉTODO DE PAGO:** A dropdown menu shows 'Tarjeta de crédito'.
- CUENTA DE LA QUE SALE EL DINERO:** A dropdown menu shows '3.0.1 - Pago a proveedores'. Below it, a green status indicator says 'Típicamente Efectivo. Caja o Banco.'
- TIPO DE MOVIMIENTO:** A dropdown menu shows 'Pago a proveedor'. Below it, a green status indicator says 'Ej: Pago a proveedor, Abono parcial.'
- DESCRIPCIÓN (OPCIONAL):** A text input field contains 'prueba'.
- Botón:** A teal button at the bottom is labeled 'Registrar Abono'.

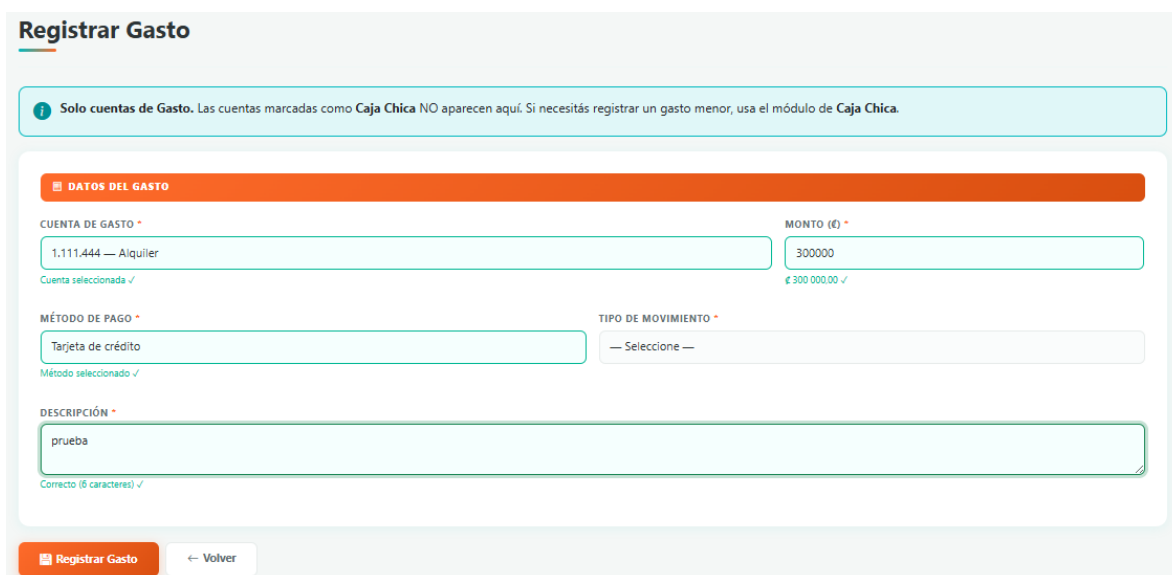
DATOS DE LA CXP:

- PROVEEDOR:** 'Ientes S.AAAA'
- VENCIMIENTO:** '29/06/2026'
- # CXP ID:** '#15'
- MONTO TOTAL:** '€75 000,00'
- YA PAGADO:** '€0,00'
- SALDO PENDIENTE:** A red box displays '€75 000,00'.

Fuente: Elaboración propia

Se probó la función de realizar abonos a las cuentas por pagar de la empresa. Si se realiza un abono inferior al monto de la cuenta, se calcula cuánto se debe pagar para finalizar la deuda y se genera un recibo para tener un mayor orden financiero.

Figura 119
Prueba 1 Gastos



Fuente: Elaboración propia

En esta prueba se creó un gasto de alquiler, con el fin de mostrar la función de crear y actualizar el monto de una cuenta. El módulo de gastos cuenta con la funcionalidad de registrar algunos movimientos de manera automática, obteniendo los datos de otros módulos.

Figura 120
Prueba 2 Gastos



Fuente: Elaboración propia

La segunda prueba consiste en mostrar la funcionalidad de que tiene el sistema al desplegar los detalles de un gasto en la cuenta. Se puede eliminar el movimiento a gusto, lo que permite corregir errores causados al registrar de manera incorrecta un movimiento.

Figura 121
Prueba 1 factura

Nueva Factura Volver al listado

CLIENTE Y MODALIDAD DE PAGO

CLIENTE * Tyron Tyron López ID 44400977 MÉTODO DE PAGO * Tarjeta de crédito MODALIDAD DE PAGO Pago a Crédito

Aplicará en Carta de Crédito

¿EN CUÁNTOS MESES?

1 mes - Pago único 2 meses - 2 cuotas 3 meses - 3 cuotas

• Máximo permitido por ley: 36 meses. Las cuotas se generan automáticamente cada mes (daily fee).

Plan de cuotas

CUOTA 1	CUOTA 2	CUOTA 3
€ 46 330,00 Desde 01/01/2024	€ 46 330,00 Desde 01/01/2024	€ 46 330,00 Desde 01/01/2024

• Total a crédito: € 138 990,00 en 3 cuotas. Última cuota vence el 01/01/2025.

DETALLE DE LA FACTURA Agregar producto/servicio

#	CÓDIGO	DESCRIPCIÓN	CANTIDAD	PRECIO UNIT.	IVA 10%	TOTAL
1.	Examen-0101	Examen de la vista Servicio: Ventas por servicios	1	14000,00	€ 1 400,00	€ 14 000,00
2.	Cata-20	Anteojos claros 15 días: Ventas de producto	1	11000,00	€ 1 100,00	€ 12 100,00

Subtotal (sin IVA): € 123 000,00

IVA (10%): € 15 990,00

TOTAL A PAGAR: € 138 990,00

Cancelar Crear Factura

Fuente: Elaboración propia

Se creó una factura a crédito de 3 meses como base de esta prueba. La empresa Ópticas Dr. Max solo brinda un plazo de 3 meses para realizar una compra a crédito. Se pueden agregar varios productos y servicios, al igual que la cantidad de estos.

Figura 122
Prueba 2 factura

Detalle de Factura

PAGADA - EFECTIVO

Pago de Contado

Factura Electrónica
FAC-20240604-0002
04 de junio 2024 10:42

Ópticas Dr. Max
Sistema Contable - Óptica Dr. Max

CLIENTE
Rocio López Rodríguez
Cédula: 11102897

CAJERO / VENDEDOR
Tiram
Usuario: 06100000

#	CÓDIGO	DESCRIPCIÓN	CANT.	PRECIO UNIT.	SUBTOTAL	IVA	TOTAL
1	Ant-30	Anteojo con lentes 75 mm x 150 mm	1	€ 37,500.00	€ 37,500.00	€ 4,875.00	€ 42,375.00

Subtotal: € 37,500.00
IVA (13%): € 4,875.00
TOTAL: € 42,375.00

Cantidad de líneas: 1
ID Sistema: 100

Documento generado por el
Sistema Contable Óptica Dr. Max
04/06/2024 10:42

© 2024 - Ópticas Dr. Max

Fuente: Elaboración propia

En este apartado de la prueba se puede anular una factura o imprimirla. Además, se detalla el producto que adquirió el cliente, junto con el precio + IVA. Asimismo, incluye la información del vendedor, el número de factura y el tipo de pago.

Figura 123
Prueba 1 Cuentas por cobrar

Sistema Contable Ópticas Dr. Max

Cuenta por Cobrar

Rocio López Rodríguez
Cédula: 11102897

FECHA DE VENCIMIENTO
27/06/2024

FECHA PARA PAGAR
En 23 días

Factura origen: FAC-0000000001
Factura n° 0000000001 Total factura € 42,375.00 Cédula n° 11102897

MONEDA TOTAL € 42,375.00
IVA COBRADO € 0.00
SALDO PENDIENTE € 42,375.00

Plan de Pagos / Plazos

#	CUENTA	MONEDA CUENTA	SALDO PEND.	ESTADO
01	27/06/2024	€ 42,375.00	€ 42,375.00	Pendiente

Productos / Servicios facturados

CÓDIGO	DESCRIPCIÓN	CANT.	PRECIO UNIT.	IVA	TOTAL LÍNEA
Ant-30	Anteojo con lentes 75 mm x 150 mm	1	€ 37,500.00	€ 4,875.00	€ 42,375.00

Historial de Abonos

Aún no hay abonos registrados para esta C/C.

Registrar Primer Abono

Fuente: Elaboración propia

Para ejecutar la prueba de cuentas por cobrar, primero se debe confeccionar una factura de crédito. Se genera automáticamente una cuenta por cobrar al cliente

correspondiente. Se pueden llevar a cabo abonos a su cuenta y se escogen los plazos que tiene para pagar la deuda.

Figura 124
Prueba 2 Cuentas por cobrar

Estados de Cuenta — Clientes

Solo se muestran clientes con saldo pendiente. Haz clic en "Ver estado" para ver el detalle de cada cliente: facturas, abonos realizados y saldo por pagar.

444000077	Tyren Tyron López	1	€124 300,00	€124 300,00	Ver estado
100000001	Iula Ramos López	1	€42 375,00	€42 375,00	Ver estado
111300001	Rocio López Rodríguez	1	€42 375,00	€42 375,00	Ver estado

Fuente: Elaboración propia

Se prueba la funcionalidad de ver a todos los clientes con cuentas por cobrar. Se pueden visualizar los movimientos de la cuenta que tienen, el monto total que deben y lo que abonaron a su deuda.

Figura 125
Prueba 1 Ingresos

Calcular Ingresos

TOTAL GENERAL DE INGRESOS	CATEGORÍAS CON INGRESOS
€57 065,00	2

FECHA INICIO: 04/06/2026 FECHA FIN: 04/06/2026 Filter Limpiar

Resumen de Ingresos - solo se muestran cuentas marcadas con **Aplica Ingreso** en Contabilidad. Total = suma de facturas pagadas + ingresos manuales en el periodo. Haz clic en **Ver movimientos** para revisar cada venta individualmente.

1.1.7	Ventas por servicios	Ingreso	€14 690,00	Ver movimientos
1.2.0	Ventas de producto	Ingreso	€42 375,00	Ver movimientos
TOTAL			€57 065,00	

Fuente: Elaboración propia

Se prueba la funcionalidad del módulo de ingreso al mostrar los movimientos y las cuentas afectadas, con el fin de obtener las ganancias de la empresa. Cuenta con un filtro que ayuda a mostrar los ingresos de un período en específico, lo que ayuda al administrador a organizarse con base en la información que se muestra.

Figura 126
Prueba 2 Ingresos

Detalle de Ingresos ← Volver

Cuenta 1.2.8
Ventas de producto
Ingresos: Naturales, Asesorías

TOTAL DE INGRESOS EN EL PERÍODO
€731 675,00

DESDE: 04/01/2025 HASTA: 04/06/2025 ▼ Filtrar

FECHA	HORA	NÚMERO DE FACTURA	DESCRIPCIÓN	CÓDIGO DE PRODUCTO	MONTANTO
04/06/2025	11:42	Factura FAC-20260604-0002	Antojitos con lentejas	FAC-20260604-0002	+ €42 375,00
05/06/2025	14:50	Factura FAC-20260610-0001	Antojitos con lentejas	FAC-20260610-0001	+ €42 375,00
26/06/2025	11:44	Factura FAC-20260629-0002	Antojitos con lentejas	FAC-20260629-0002	+ €42 375,00
27/06/2025	16:26	Factura FAC-20260627-0002	Antojitos con lentejas	FAC-20260627-0002	+ €497 206,00
27/06/2025	16:24	Factura FAC-20260627-0001	Lentejas	FAC-20260627-0001	+ €39 325,00
26/06/2025	00:34	Factura FAC-20260626-0001	Examen de la vista	FAC-20260626-0001	+ €11 300,00
25/06/2025	19:24	Factura FAC-20260625-0002	Antojitos con lentejas	FAC-20260625-0002	+ €36 725,00

Fuente: Elaboración propia

La última prueba que se realizó con respecto al módulo de ingresos es la opción que posee el sistema, la cual consiste en mostrar los movimientos de una cuenta en específico. Esto genera confiabilidad en el sistema, ya que demuestra el porqué del monto que se observa en la pantalla principal del módulo.

Capítulo VI. Conclusiones y recomendaciones

Conclusiones

Conclusión 1: analizar los requerimientos funcionales y no funcionales

Se realizó un análisis de los requerimientos del sistema mediante reuniones con el gerente de la empresa y se crearon los casos de uso para cada uno de los módulos que se identifican, tales como facturación, contabilidad, caja chica, inventario, cuentas por cobrar y cuentas por pagar. Lo anterior permitió diferenciar entre los requerimientos funcionales, que describen qué debe hacer el sistema y los no funcionales, los cuales contemplan aspectos como seguridad, rendimiento y mantenibilidad. Esta etapa resultó fundamental, ya que estableció una base sólida para el diseño y la programación del sistema, garantizando que la solución desarrollada respondiera a las necesidades reales del negocio de Ópticas Dr. Max.

Conclusión 2: construir el diseño de la base de datos mediante el uso de MySQL Workbench

Se creó la base de datos del sistema mediante el uso de herramientas como MySQL Workbench y SQL Server Management Studio. En Workbench se realizó la correcta diagramación de las tablas, las cuales contienen FK, PK, VARCHAR, INT, entre otras. Lo anterior se llevó a cabo para concretar de manera correcta la función de los módulos. Esta herramienta fue de gran ayuda, ya que brindó una previsualización de la base del sistema, lo que facilitó el desarrollo del código.

Conclusión 3: programar los módulos mediante el uso de C# y con la ayuda de Entity Framework

La programación del *back-end* y *front-end* se realizó en Visual Studio con ayuda de Entity Framework. Estas herramientas fueron de gran ayuda al programar los controladores que contienen la funcionalidad del sistema y de los módulos propuestos. A la vez, se utilizó un modelo MVC, el cual ayudó con el orden al organizar en secciones las diversas etapas del desarrollo.

Conclusión 4: probar los diferentes módulos de forma independiente y también de manera integral, garantizando los resultados correctos del sistema

Se ejecutaron pruebas sobre cada uno de los módulos del sistema, con el fin de validar que el comportamiento individual y la interacción entre estos respondieran a los requerimientos establecidos. Lo anterior se realizó mediante la ejecución de casos de prueba diseñados a partir de escenarios de uso reales, lo que permitió identificar y corregir de manera oportuna los errores antes de la entrega final. A la vez, las pruebas integrales garantizaron que la información circulara de manera correcta entre los módulos contables, de facturación e inventario, lo cual asegura que los resultados generados por el sistema son consistentes y confiables para la toma de decisiones por parte de la administración del negocio.

Recomendaciones

Recomendación 1: implementar un esquema de respaldos automatizado de la base de datos

Se recomienda configurar respaldos diarios incrementales y respaldos completos semanales de la base de datos, almacenarlos en un servidor secundario o en almacenamiento en la nube. Lo anterior tiene el fin de garantizar la continuidad operativa ante un eventual pérdida o corrupción de datos.

Responsable: administrador de TI/encargado de sistemas.

Tiempo estimado: 2 semanas para configuración inicial y pruebas de restauración.

Recomendación 2: establecer redundancia de enlaces de conexión a Internet y a la base de datos

Se recomienda contar con un proveedor secundario de Internet y una conexión de respaldo a la base de datos, de manera que ante una falla en el enlace principal, el sistema continúe operando sin interrupciones, especialmente durante las horas de mayor carga operativa del negocio.

Responsable: Departamento de TI/proveedor de servicios.

Tiempo estimado: 4 a 6 semanas para contratación, instalación y pruebas.

Recomendación 3: migrar el sistema a un entorno de producción con servidor dedicado y certificado TLS

Se recomienda alojar el sistema en un servidor dedicado de producción con dominio propio y certificado SSL/TLS, con el objetivo de cifrar las comunicaciones entre el cliente y el servidor, así como mejorar la disponibilidad y el rendimiento del sistema en uso real.

Responsable: desarrollador/administrador de TI.

Tiempo estimado: 3 semanas para configuración, despliegue y migración de datos.

Recomendación 4: implementar un plan de contingencia y recuperación ante desastres

Se recomienda elaborar y documentar un plan formal de recuperación que contemple los procedimientos ante incidentes mayores, como caídas prolongadas del servicio, pérdida total de datos o ataques de seguridad y que defina los tiempos objetivo de recuperación y los responsables de cada acción.

Responsable: Gerencia Administrativa junto con el Departamento de TI.

Tiempo estimado: 1 mes para la elaboración y aprobación del plan.

Referencias

- Appinio Research. (2023). *Exploratory research: Definition, types, examples*.
<https://www.appinio.com/en/blog/market-research/exploratory-research>
- Arimetrics. (2026). *Qué es Visual Studio*. <https://www.arimetrics.com/glosario-digital/visual-studio>
- Blanco, Á. (2023). SQL Server Management Studio: Qué es, ventajas y cómo descargarlo. *El Blog de Revolution Soft sobre Software*. <https://blog.revolutionsoft.net/sql-server-management-studio/>
- Bosys Company. (2026). *Facturas Contables: Qué Son y Por Qué Son Importantes*.
<https://bosys.company/Facturas-Contables-Que-Son-y-Por-Que-Son-Importantes.php>
- CMC Global. (2025). *Waterfall model: Definition, pros and cons*.
<https://cmcglobal.com.vn/it-insights/waterfall-model-definition-pros-and-cons-updated-in-2025/>
- Concepto.de. (2021). *Método cuantitativo*. <https://concepto.de/metodo-cuantitativo/>
- Concepto.de. (2021). *Observación*. <https://concepto.de/observacion/>
- CONTPAQi. (2025). *Sistemas contables: ¿Qué son y cómo elegir el mejor en línea?*
<https://www.contpaqi.com/publicaciones/nube/que-es-un-sistema-contable-en-linea>
- Conzultek. (2026). *¿Qué es un sistema contable? Y sus beneficios*.
<https://blog.conzultek.com/que-es-un-sistema-contable>
- Cueva Luza, T.; Jara Córdova, O.; Arias Gonzáles, J.; Flores Limo, F. y Balmaceda Flores, C. (2023). *Métodos mixtos de investigación para principiantes*.
<https://editorial.inudi.edu.pe/editorialinudi/es/catalog/download/119/161/190?inline=1>
- Das, S. (2025). *What is system testing? (examples, use cases, types)*. BrowserStack.
<https://www.browserstack.com/guide/what-is-system-testing>
- Durán, L. (2026). *¿Qué es ingreso en contabilidad? Definición, tipos y ejemplos. Blog Inteligencia Empresarial*. https://blog.alegra.com/republica-dominicana/que-es-ingreso-en-contabilidad/?cq_plac=&cq_net=g&cq_pos=&cq_med=&cq_plt=gp&utm_term=&utm_campaign=21378496040&utm_source=google&utm_medium=ppc&utm_content=183795530306&hsa_acc=6849555148&hsa_cam=21378496040&hsa_grp=183795530306&hsa_ad=795493275082&hsa_src=g&hsa_tgt=dsa-59937357129&hsa_kw=&hsa_mt=&hsa_net=adwords&hsa_ver=3&gad_source=1&gad_campaignid=21378496040&gbraid=0AAAAAClJnFpPP-

- xB0W7kErrIt7TWYpWaM&gclid=Cj0KCQiAnJHMBhDAARIsABr7b87BVbEqgj1q4KMN0A4Vw3dzE_va13pgx6U5ASVvnn9AmWTLatsOA6gaAnTjEALw_wcB
- Explorable.com. (2022). *Variables conceptuales*. <https://explorable.com/es/variables-conceptuales>
- Fernando, J. (2025). *Accounting Explained With Brief History and Modern Job Requirements*. Investopedia. <https://www.investopedia.com/terms/a/accounting.asp>
- FJ Intelligence. (2023). *Fuentes secundarias*. <https://fjintelligence.com/terminos/fuentes-secundarias/>
- Google Cloud. (2026). *¿Qué es una base de datos relacional (RDBMS)?* <https://cloud.google.com/learn/what-is-a-relational-database?hl=es-419>
- Google.com. (2026). *Google Cloud pricing calculator*. https://cloud.google.com/products/calculator?_gl=1Ifg4xa8_up*MQ..&gclid=Cj0KCQiAhOfLBhCCARIsAJPiopOE7PUE4mtqc3qijbOud0Iwe287Ky8BLXOoA8wSMjLHxAdZAZSQiQ0aAgXMEALw_wcB&gclsrc=aw.ds&dl=CjhDaVF5TURZeU9XWTRaQzAzWmpSaUxUUTRZall0T0dSbU9TMDBOV0UxTW1VMU16STNaRFIRQVE9PRAHGIRDNjcxQzY5MS01MTc0LTQxNEMtQTM5Ni01MEE4NzlCN0YwQjE
- GRCPlus. (2024). *Blog - SUGEF: Supervisión financiera, confianza y cumplimiento en Costa Rica*. <https://grcplus.cl/blog-sugef-costa-rica-supervision-financiera/>
- Hecker, J. y Kalpokas, N. (2025). *¿Qué es la investigación cualitativa?* ATLAS.ti. <https://atlasti.com/es/guias/guia-investigacion-cualitativa-parte-1/investigacion-cualitativa>
- Inesdi. (2025). *Qué es el software, que tipos existen y cómo funcionan*. <https://www.inesdi.com/blog/tipos-de-software/>
- Innovacommerce. (2021). *¿Qué es el Cierre de Caja? ¿Y un Arqueo?* <https://www.innovacommerce.com/cierre-de-caja/>
- IONOS Digital Guide. (2024). *¿Qué es el Frontend?* <https://www.ionos.com/es-us/digitalguide/paginas-web/creacion-de-paginas-web/que-es-el-frontend/>
- Ionos. (2024). *¿Qué es el Frontend?* <https://www.ionos.es/digitalguide/paginas-web/creacion-de-paginas-web/que-es-el-frontend/>
- Jain, S. (2024). *What is waterfall lifecycle*. BrowserStack. <https://www.browserstack.com/guide/waterfall-model>
- Kaushik, N. y Kaushik, N. (2025). *What Is Petty Cash? Meaning, Process & Accounting Treatment*. Happay. <https://happay.com/blog/petty-cash/>

- Ley n.º 1269. (1951). *Ley Orgánica del Colegio de Contadores Privados de Costa Rica*.
https://pgrweb.go.cr/scij/Busqueda/Normativa/Normas/nrm_texto_completo.aspx?param1=NRTC&nValor1=1&nValor2=186&nValor3=195&strTipM=TC
- Ley n.º 3284. (1964). *Código de Comercio*.
https://pgrweb.go.cr/scij/Busqueda/Normativa/normas/nrm_texto_completo.aspx?nValor1=1&nValor2=6239
- Ley n.º 6683. (2015). *Ley de derechos de autor y derechos conexos*.
<https://www.ucr.ac.cr/medios/documentos/2015/LEY-6683.pdf>
- Ley n.º 8148. (2001). *Adición de los artículos 196 BIS, 217 BIS y 229 BIS al Código Penal, Ley n.º 4573 para reprimir y sancionar los delitos informáticos*.
https://pgrweb.go.cr/scij/Busqueda/Normativa/Normas/nrm_texto_completo.aspx?param1=NRTC&nValor1=1&nValor2=47430&nValor3=50318&strTipM=TC
- Ley n.º 9635. (2018). *Fortalecimiento de las finanzas públicas*.
https://pgrweb.go.cr/scij/Busqueda/Normativa/Normas/nrm_texto_completo.aspx?nValor1=1&nValor2=87720
- Mamian, M. (2025). ¿Qué son los gastos en contabilidad? *Siigo.com*.
<https://www.siigo.com/blog/contador/que-son-los-gastos-en-contabilidad/>
- Mi sitio de Framer. (2023). ¿Qué son las cuentas por pagar y cómo se clasifican?
<https://www.chipax.com/es/blog/cuentas-por-pagar>
- Microsoft.com. (2025). *Pricing calculator*. https://azure.microsoft.com/en-us/pricing/calculator/?ef_id=_k_Cj0KCQiAhOfLBhCCARIsAJPiopN6tmQ4bwlcXhmDxk1tcSse304y3qWOP5UPbc9TIRfoYXSl_U78ACQaAuk-EALw_wcB_k_&OCID=AIDcmmxm7v0egx_SEM__k_Cj0KCQiAhOfLBhCCARIsAJPiopN6tmQ4bwlcXhmDxk1tcSse304y3qWOP5UPbc9TIRfoYXSl_U78ACQaAuk-EALw_wcB_k_&gad_source=1&gad_campaignid=11421025082&gbraid=0AAAAADcJh_vDya3Jl6lK13TH-
- NetMentor. (2021). *Qué es Entity Framework Core?*
<https://www.netmentor.es/entrada/introduccion-entity-framework-core>
- Pérez, A. (2023). *Qué es la investigación explicativa y sus características*. Tesis doctorales online. <https://tesisdoctoralesonline.com/que-es-la-investigacion-explicativa-y-sus-caracteristicas/>
- Perucontable. (2022). ¿Qué es un presupuesto de contabilidad?
<https://www.perucontable.com/contabilidad/que-es-un-presupuesto-decontabilidad/>
- Piyush Verma. (2026). *What is C# Programming Language?*
<https://www.igmguru.com/blog/what-is-c-sharp>

- QuickBooks en español. (2023). *Sistema contable de una empresa: características y tipos*. <https://quickbooks.intuit.com/global/resources/es/contabilidad-y-registro-contable/estas-son-todas-las-caracteristicas-que-debe-tener-un-software-contable>
- Revolution Soft. (2023). *SQL Server Management Studio: Qué es, ventajas y cómo descargarlo*. <https://blog.revolutionsoft.net/sql-server-management-studio/>
- Sage. (2025). *Cuentas Contables: Definición, tipos y ejemplos prácticos*. <https://www.sage.com/es-es/blog/diccionario-empresarial/cuentas-contables/>
- SalusPlay. (2026). *Las variables de investigación*. <https://www.salusplay.com/apuntes/apuntes-metodologia-de-la-investigacion/tema-2-las-variables-de-investigacion>
- Science Direct. (2026). *System Requirement*. <https://www.sciencedirect.com/topics/computer-science/system-requirement>
- SentinelOne. (2024). *¿Qué es el desarrollo de sistemas? Explicación de los conceptos clave*. <https://www.sentinelone.com/es/cybersecurity-101/cybersecurity/what-is-systems-development/>
- Stewart, L. (2025). *¿Qué es la investigación descriptiva y cómo se utiliza? ATLAS.ti*. <https://atlasti.com/es/research-hub/investigacion-descriptiva>
- Superintendencia General de Entidades Financieras (Sugef). (s. f.). *Inicio*. <https://www.sugef.fi.cr/>
- Taylor, J. (2024). *Definition of Waterfall model*. <https://maddevs.io/glossary/waterfall-model>
- Unacademy. (2022). *The interview method of collecting data*. <https://unacademy.com/content/ssc/study-material/statistics/the-interview-method-of-collecting-data/>
- University of Minnesota Crookston. (s. f.). *Primary, secondary, and tertiary sources*. <https://crk.umn.edu/library/primary-secondary-and-tertiary-sources>
- Visual Studio. (2018). *Precios de Visual Studio: comparar planes de suscripción y costos*. <https://visualstudio.microsoft.com/es/vs/pricing/>
- Weibel, A. (2025). *What is a Design System and How Does it Tame Your UI?* Salesforce. <https://www.salesforce.com/blog/what-is-a-design-system/>

Anexos

Guía de la entrevista

Entidad:	Ópticas Dr. Max
Nombre del entrevistado:	Andrés Arturo Ulloa Fernández
Puesto del entrevistado:	Gerente
Nombre de la persona estudiante:	Tyron López Chinchilla
Fecha de la entrevista:	15/04/2026
Lugar o medio de la entrevista:	Establecimiento de Ópticas Dr. Max ubicado en Heredia

Fuente: Elaboración propia

Preguntas:

Pregunta n.º 1

¿Cómo verifica usted que el dinero recibido durante el día coincide con las ventas realizadas?

Se hace un cierre de caja en el que se dividen los ingresos dependiendo si fueron en efectivo o tarjeta, después se hace una suma para obtener el monto total del día.

Pregunta n.º 2

¿Cuáles son los principales gastos operativos y cómo llevan el control de esos gastos?

Los pagos de facturas a los distintos proveedores que tenemos y gastos públicos (agua, luz, teléfono, Internet)

Pregunta n.º 3

Cuando se realiza una venta, ¿qué información colocan en la factura?

Tiene un numero de factura, en ciertos casos el nombre del cliente, tipo de pago, monto, servicio o elemento vendido, la fecha y hora de la emisión de la factura. Si el cliente no realiza el pago de a contado y opta por el beneficio que tenemos de pagos a plazos en la factura le agregamos el plazo y la cantidad a pagar por plazo.

Pregunta n.º 4

¿Cómo registran y controlan los pagos pendientes que tiene la empresa?

Se manejan de manera manual en un Excel, en una hoja se agregan los gastos y pagos pendientes del mes.

Pregunta n.º 5

¿Cuáles tipos de gastos pequeños se pagan usualmente con el dinero disponible en la caja chica y cómo se registra actualmente cada uno de esos movimientos?

Objetos de oficina (papel, folder, lapiceros, etc.), limpieza (bolsas de basura, desinfectantes, etc.) y de uso del personal en el local (papel higiénico, etc.), actualmente tenemos esos gastos y segmentados de esta forma.

Pregunta n.º 6

¿Existen otras fuentes de ingreso como servicios de exámenes de la vista o reparaciones y, si existen, cómo registran estos ingresos?

Sí, tenemos lo que son los exámenes y reparaciones. Todos los servicios brindados en la óptica se facturan, el cambio se da en la factura porque se especifica el servicio o el elemento vendido

Pregunta n.º 7

¿Cómo organizan actualmente la información financiera de la óptica para saber cuánto dinero entra y cuánto sale?

Al final del día se cuenta el dinero de la caja, además de los pagos con tarjetas, el valor obtenido se guarda en un Excel.

Pregunta n.º 8

¿Qué tipo de información financiera le gustaría poder consultar fácilmente para tomar decisiones sobre el negocio?

Ver los gastos exactos de la caja chica, el dinero disponible en la caja chica, las facturas pendientes y un historial de facturas pagas, además de los ingresos obtenidos, ya sean diarios o mensuales.

Pregunta n.º 9

¿Cómo funciona el método que ustedes utilizan para controlar cuándo un cliente queda debiendo dinero de algún servicio que se brinda?

Por políticas no damos fiado, entonces eso no es un problema, sin embargo, nosotros tenemos lo que viene siendo un beneficio, que si el cliente no puede o no quiere pagar de contado le damos a un plazo 3 meses, el cliente con este beneficio puede cancelar un producto o un servicio de manera más cómoda.

Pregunta n.º 10

¿La empresa establece algún presupuesto mensual o anual para controlar los gastos o ingresos del negocio y cómo revisan este presupuesto para tomar acciones en caso de que lo requieran?

El presupuesto se da mensualmente y nosotros con base a lo que tenemos nos organizamos para los gastos que se den, pero es necesario tener un orden o porque a veces por apuntar las cosas de forma manual hay ciertos errores de cálculo que afectan al presupuesto mensual estimado,

Guía de la observación

Entidad:	Ópticas Dr. Max
-----------------	------------------------

Dirección física de la entidad:	Heredia centro, calle 2 entre avenida 6 y 8, costado este del mercado antiguo. Plaza San Francisco
Fecha de la actividad de observación:	15/04/2026
Nombre de la persona estudiante:	Tyron López Chinchilla

Fuente: Elaboración propia

Tabla de control de aspectos observados:

No	Aspectos por observar	Cumple	No cumple	Oportunidad de mejora	Detalle de observación
1	Método que se utiliza al registrar movimiento en la caja chica	Utilizan la herramienta de Microsoft Excel para registrar los movimientos de la caja chica.	No tienen un orden y se pueden confundir muchos datos, por lo que se suelen dar errores en los montos.	Se puede realizar un sistema, el cual tenga todos los datos que se deseen registrar sobre los movimientos de la caja chica.	Se busca observar qué método utiliza la empresa al registrar movimientos de la caja chica, esto con el fin de obtener información útil para el futuro sistema.
2	Forma en la que se registran los gastos	Tienen un libro de Microsoft Excel en el que se toman apuntes de los gastos que tiene la empresa (agua, luz, Internet, teléfono, etc.)	No cuentan con un sistema especializado que ayude a mantener las finanzas ordenadas.	Se puede realizar un sistema que ayude con este proceso en donde se pueda obtener estos datos de manera segura y rápida.	Se busca observar cómo la empresa registra los diferentes gastos operativos, con el fin de identificar los métodos que se utilizan y obtener información útil para el diseño del sistema.

No	Aspectos por observar	Cumple	No cumple	Oportunidad de mejora	Detalle de observación
3	Proceso que se utiliza para la creación de facturas	Tienen un método que realiza facturas, no obstante, estas son físicas. Se utilizan datos como nombre del cliente, tipo de pago, monto, servicio o elemento vendido, la fecha y hora de la emisión de la factura.	Al ser facturas físicas y no tener un sistema que las almacene se pueden perder o realizar mal los cálculos financieros.	Se puede elaborar un sistema que almacene las facturas y permita crear nuevas teniendo un historial que ayude a organizar las cuentas de la empresa.	Se busca observar el procedimiento que utiliza la empresa en el momento de emitir facturas, con el fin de conocer los datos que se registran y utilizarlos como referencia para el sistema.
4	Método que se utiliza para registrar los ingresos del negocio	Al igual que los gastos el registro de ingresos se guarda en un libro de Microsoft Excel (venta de producto, exámenes y reparaciones).	No tienen un sistema que ayude a contabilizar de manera correcta los ingresos y almacenar estos montos.	Al realizar un sistema contable se puede acceder a estos datos de manera rápida y aportando un alto grado de confianza al saber que estos datos se encuentran seguros.	Se busca observar la forma en la que la empresa registra los ingresos generados por las ventas o servicios ofrecidos.

No	Aspectos por observar	Cumple	No cumple	Oportunidad de mejora	Detalle de observación
5	Control de pagos a proveedores	La empresa cuenta con una libreta en la cual se detalla el proveedor y la cantidad por pagar junto con la fecha del pago.	La empresa no tiene algún sistema que ayude con estas obligaciones. Utilizan una libreta de apuntes en donde solo viene la fecha y la cantidad por pagar.	Se puede realizar un apartado en el sistema contable que tenga las cuentas por pagar de la empresa con los proveedores. Mejorando la obtención de estos datos.	Se busca observar cómo la empresa lleva el control de los pagos que debe realizar a proveedores, con el fin de identificar el método que se utiliza para registrar y dar seguimiento a estas obligaciones.
6	Control de cuentas por cobrar a clientes	Poseen una libreta en la que se apunta el nombre del cliente, producto o servicio y la cantidad que debe pagar, también se utiliza la factura para saber cuánto pagó y cuanto le queda por pagar	La empresa por cuestiones de política no da fiado, no obstante, cuentan con un beneficio de pago a 3 meses de los productos. Los clientes que usan este beneficio se apuntan en una libreta.	Al realizar un sistema se puede tener un mejor registro de los clientes que tienen una cuenta por pagar con la empresa, lo que beneficia en gran medida a la compañía, ya que se aseguran mantener los datos al día y correctamente guardados.	Se busca observar la forma en la que la empresa registra y da seguimiento a los montos pendientes de pago por parte de los clientes.

No	Aspectos por observar	Cumple	No cumple	Oportunidad de mejora	Detalle de observación
7	Control del presupuesto del negocio	Cuentan con un libro de Microsoft Excel que ayuda a monitorear el presupuesto mensual de la empresa.	No cumplen con una seguridad adecuada al realizar el presupuesto, por lo que lleva trabajo extra para no cometer errores.	Se puede realizar un apartado, el cual ayude a llevar a cabo el presupuesto mensual de la empresa, en este presupuesto. Ayudando, de manera que se puede ver la cantidad de dinero que se gastó de ese presupuesto y el total que sobró.	Se busca observar si la empresa establece algún control o planificación sobre sus gastos e ingresos mediante un presupuesto, con el fin de conocer el método que se utiliza y considerar esta funcionalidad en el sistema.

Fuente: Elaboración propia