



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA LA
GESTIÓN DE CARTERA DE LA EMPRESA IBERCONSA APLICANDO LA
METODOLOGÍA SWIRL.

ROMERO MACHARE BARBARA BRIGITTE
INGENIERA DE SISTEMAS

MACHALA
2021



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA LA
GESTIÓN DE CARTERA DE LA EMPRESA IBERCONSA
APLICANDO LA METODOLOGÍA SWIRL.

ROMERO MACHARE BARBARA BRIGITTE
INGENIERA DE SISTEMAS

MACHALA
2021



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

TRABAJO TITULACIÓN
PROPUESTAS TECNOLÓGICAS

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA LA GESTIÓN DE
CARTERA DE LA EMPRESA IBERCONSA APLICANDO LA METODOLOGÍA
SWIRL.

ROMERO MACHARE BARBARA BRIGITTE
INGENIERA DE SISTEMAS

CÁRDENAS VILLAVICENCIO OSCAR EFRÉN

MACHALA, 28 DE SEPTIEMBRE DE 2021

MACHALA
2021

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA LA GESTIÓN DE CARTERA DE LA EMPRESA IBERCONSA APLICANDO LA METODOLOGÍA SWIRL

INFORME DE ORIGINALIDAD

0%

INDICE DE SIMILITUD

0%

FUENTES DE INTERNET

0%

PUBLICACIONES

0%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1

www.ind.unitru.edu.pe

Fuente de Internet

<1%

Excluir citas

Activo

Excluir coincidencias < 10 words

Excluir bibliografía

Activo

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

La que suscribe, ROMERO MACHARE BARBARA BRIGITTE, en calidad de autora del siguiente trabajo escrito titulado DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA WEB PARA LA GESTIÓN DE CARTERA DE LA EMPRESA IBERCONSA APLICANDO LA METODOLOGÍA SWIRL., otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

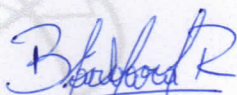
La autora declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

La autora como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 28 de septiembre de 2021



ROMERO MACHARE BARBARA BRIGITTE
0706262748

DEDICATORIA

Quiero dedicar el presente trabajo especialmente a mi madre, quien ha sido un pilar fundamental en mi vida, enseñándome a nunca rendirme ante las adversidades, y siempre luchar por mis sueños. Me ha apoyado a lo largo de mi vida personal y académica, brindándome lo necesario para salir adelante.

También va dedicado a mis hermanos y padre, quienes día a día me han brindado su apoyo y alentado a esforzarme para cumplir mis metas planteadas. A mis abuelos, quienes siempre me han inculcado a estudiar y prepararme profesionalmente, animándome siempre en las dificultades que se presentaban.

Y finalmente, a mis familiares, quienes con su cariño y dedicación me han aconsejado, ayudado, y acompañado a lo largo de la vida.

Srta. Romero Macharé Bárbara Brigitte

AGRADECIMIENTO

Quiero agradecer principalmente a Dios, quien me ha dado fortaleza todos los días para superar los retos y adversidades, quien me ha dado salud y sabiduría para alcanzar todas las metas y proyectos que me he propuesto en la vida.

A mi madre, quien me ha apoyado siempre y ha estado conmigo en los momentos que más he necesitado, enseñándome a luchar y perseverar y siendo un gran ejemplo a seguir; a mi hermano menor quien es mi motor para seguir adelante y esforzarme en cada paso que doy. A mi hermano mayor y mi padre quienes han estado apoyándome en mis estudios.

Finalmente, quiero agradecer a mi tutor, quien me ha guiado y ayudado a culminar con éxito el proceso de titulación, y a mis docentes, quienes me han impartido su conocimiento en cada semestre cursado.

Srta. Romero Macharé Bárbara Brigitte

RESUMEN

Debido a la implementación de los sistemas de información en el ámbito empresarial, se ha logrado optimizar en gran medida el desempeño de las actividades de negocio de cada empresa, convirtiéndose en un recurso imprescindible a medida que el negocio o empresa incrementa. El control adecuado de cartera dentro de una organización es de utilidad, ya que permite mantener estabilidad dentro de los balances contables y ganancias que pueda obtener la organización. Por lo tanto, el seguimiento debe ser efectuado de manera continua y meticulosa, evitando posibles pérdidas en la empresa. Los sistemas de información han logrado optimizar los procesos de control de las empresas, permitiendo llevar un seguimiento de los créditos otorgados para cada cliente. Sin embargo, muy pocos sistemas de esta naturaleza son adquiridos por PYMEs ya sea por su dependencia de otros módulos del sistema o su complejidad de entendimiento de interfaz. La necesidad de estar conectado a internet ha impulsado el consumo de sistemas orientados a la web, brindando la facilidad del acceso a la información desde múltiples puntos. Es por ello, que para su desarrollo e implementación se emplean metodologías de desarrollo orientadas a la web, con las cuales se pueda garantizar la producción de un sistema de calidad, además de características relevantes como la accesibilidad, usabilidad y en varios casos modularidad.

En base al análisis expuesto, la propuesta tecnológica se centra en diseñar e implementar un sistema administrativo modular web para la gestión de cartera en la empresa constructora IBERCONSA mediante los procesos inmersos en la metodología de desarrollo SWIRL. El sistema de información realizará el seguimiento y control de la cartera de crédito de cada uno de los clientes que posee la empresa, mediante apertura de nuevos créditos, registro de pagos de cuotas mensuales, creación de contratos para validar el crédito, visualización de los adeudos e históricos de pago por cliente, y finalmente llevar un control de las cuotas vencidas y los pagos realizados de manera general. Los requerimientos del sistema son emitidos por la empresa, siendo imprescindible el uso de un enfoque administrativo durante su diseño. De esta manera, se establecen las funciones según los roles delegados en la empresa y los permisos asignados para cada uno de ellos. Cabe recalcar, que el sistema cuenta con la gestión de usuarios, roles, y asignación de permisos según las políticas de cada empresa.

El proceso de desarrollo se llevó a cabo según los procedimientos establecidos por la metodología SWIRL; por lo que se realizó el análisis y planificación debido del proceso, identificando los requerimientos del sistema, usuarios, y funcionalidades de cada módulo a desarrollar; seguido del diseño de interfaces de usuarios, diagramas UML, y diseño de la base de datos mediante el uso de herramientas CASE respectivas para cada modelo. Para la implementación o codificación de la propuesta se han seleccionado tecnologías para el lado del cliente y para el lado del servidor. En el lado del servidor se empleó Node.js y Express.js como tecnología de desarrollo, y MongoDB, como gestor de base de datos. Mientras que en el lado del cliente se empleó el lenguaje JavaScript, y el Framework Angular.js para el desarrollo de las interfaces de usuario, incluyendo estándares de CSS, Bootstrap, y HTML para su respectivo diseño.

Una vez desarrollado, se ejecutaron las pruebas considerando las normas de la metodología, evaluando de esta manera la calidad según la norma ISO 9126; y la usabilidad y accesibilidad mediante el uso de herramientas de análisis. El sistema se puso en producción en un host en la Nube, en donde se realizaron las pruebas de funcionamiento pertinente, para posteriormente ser introducido en los procesos de negocio de la empresa.

Palabras clave: metodología SWIRL, Desarrollo web, Sistema de información, Control de cartera.

ABSTRACT

Due to the implementation of information systems in the business environment, it has been possible to optimize to a great extent the performance of the business activities of each company, becoming an essential resource as the business or company grows. Adequate portfolio control within an organization is useful, since it allows maintaining stability within the accounting balances and profits that the organization can obtain. Therefore, the follow-up must be carried out in a continuous and meticulous way, avoiding possible losses in the company. Information systems have been able to optimize the control processes of companies, allowing them to keep track of the loans granted to each client. However, very few systems of this nature are acquired by SMEs, either because of their dependence on other modules of the system or because of the complexity of understanding the interface. The need to be connected to the Internet has driven the consumption of web-oriented systems, providing easy access to information from multiple points. That is why, for its development and implementation, web-oriented development methodologies are used to guarantee the production of a quality system, in addition to relevant features such as accessibility, usability and, in several cases, modularity.

Based on the above analysis, the technological proposal focuses on designing and implementing a modular web administrative system for portfolio management in the construction company IBERCONSA through the processes immersed in the SWIRL development methodology. The information system will monitor and control the credit portfolio of each of the company's clients, by opening new credits, registering monthly installment payments, creating contracts to validate the credit, visualizing debts and payment history by client, and finally keeping track of overdue installments and payments made in general. The system requirements are issued by the company, essentially the use of an administrative approach during its design. In this way, the functions are established according to the roles delegated in the company and the permissions assigned to each of them. It should be noted that the system has the management of users, roles, and assignment of permissions according to the policies of each company.

The development process was carried out according to the procedures established by the SWIRL methodology; therefore, the analysis and due planning of the process was

performed, identifying the system requirements, users, and functionalities of each module to be developed; followed by the design of user interfaces, UML diagrams, and database design through the use of the respective CASE tools for each model. For the implementation or coding of the proposal, client-side and server-side technologies were selected. On the server side, Node.js and Express.js were used as development technology, and MongoDB as database manager. On the client side, JavaScript language and the Angular.js Framework were used for the development of the user interfaces, including CSS, Bootstrap and HTML standards for their respective design.

Once developed, the tests were executed considering the methodology standards, thus evaluating the quality according to ISO 9126; and the usability and accessibility through the use of analysis tools. The system was put into production in a Cloud host, where the relevant functional tests were carried out, to be subsequently introduced in the company's business processes.

Key words: SWIRL methodology, Web development, Information system, Portfolio control.

ÍNDICE GENERAL

DEDICATORIA	1
AGRADECIMIENTO	2
RESUMEN.....	3
ABSTRACT.....	5
INTRODUCCIÓN.....	13
1 CAPÍTULO I. DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS.	15
1.1 ÁMBITO DE APLICACIÓN: DESCRIPCIÓN DEL CONTEXTO Y HECHOS DE INTERÉS.....	15
1.2 ESTABLECIMIENTO DE REQUERIMIENTOS.	16
1.3 JUSTIFICACIÓN DEL REQUERIMIENTO A SATISFACER.....	17
2 CAPÍTULO II. DESARROLLO DEL PROTOTIPO	19
2.1 DEFINICIÓN DEL PROTOTIPO TECNOLÓGICO.	19
2.1.1 Arquitectura del software.	19
2.1.2 Definición del sistema.....	20
2.1.3 Identificación de interesados	30
2.1.4 Requerimientos	30
2.2 FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO.....	32
2.2.1 Desarrollo de software.....	32
2.2.2 Gestión de cartera.	37
2.2.3 Plataforma web.	38
2.3 OBJETIVOS DEL PROTOTIPO.....	40
2.3.1 Objetivo General	40
2.3.2 Objetivos Específicos	40
2.4 DISEÑO DEL PROTOTIPO.....	41
2.4.1 Fase de planificación.....	41
2.4.2 Fase de modelado.....	49

2.5	EJECUCIÓN Y/O ENSAMBLAJE DEL PROTOTIPO.....	69
2.5.1	Página principal.....	69
2.5.2	Páginas para la gestión de usuario.....	71
2.5.3	Páginas para la gestión de clientes	73
2.5.4	Páginas para la gestión de contratos.....	74
2.5.5	Páginas para la gestión de pagos.....	77
2.5.6	Páginas para la gestión de cartera	79
2.5.7	Diseño según tendencia.	81
3	Capítulo III. Evaluación del prototipo.....	82
3.1	PLAN DE EVALUACIÓN.....	82
3.1.1	Evaluación de calidad.....	82
3.1.2	Evaluación mediante herramientas SEO	82
3.2	RESULTADOS DE LA EVALUACIÓN.....	83
3.2.1	Evaluación de calidad.....	83
3.2.2	Evaluación mediante herramientas SEO	85
3.3	CONCLUSIONES.	89
3.4	RECOMENDACIONES.....	90
	BIBLIOGRAFÍA.....	91
	ANEXO A: ANÁLISIS DE RIESGOS.....	94
	ANEXO B: DESGLOSE DE REQUERIMIENTOS	97
	ANEXO C: ESTRUCTURA DE DESGLOSE DE TRABAJO	101
	ANEXO D: MODELADO DE BASE DE DATOS	102
	ANEXO E: DIAGRAMAS DE CASO DE USO	109

ÍNDICE DE TABLAS

Tabla 1: Usuarios del sistema: U. Root	22
Tabla 2: Usuarios del sistema: U. Final.....	23
Tabla 3: Beneficios esperados	24
Tabla 4: Factibilidad operativa	26
Tabla 5: Factibilidad económica.....	27
Tabla 6: Módulos del sistema.	28
Tabla 7: Análisis cualitativo y cuantitativo de riesgos.	29
Tabla 8: Matriz general de interesados.	30
Tabla 9: Matriz general de interesados.	31
Tabla 10: Matriz general de interesados.	31
Tabla 11: Actividades fase de análisis.	35
Tabla 12: Historia de usuario #1.	41
Tabla 13: Historia de usuario #2.	42
Tabla 14: Historia de usuario #3.	42
Tabla 15: Historia de usuario #4.	43
Tabla 16: Historia de usuario #5.	43
Tabla 17: Historia de usuario #6.	44
Tabla 18: Historia de usuario #7.	44
Tabla 19: Historia de usuario #8.	45
Tabla 20: Historia de usuario #9.	45
Tabla 21: Historia de usuario #10.	46
Tabla 22: Historia de usuario #11.	46
Tabla 23: Historia de usuario #11.	47
Tabla 24: Escala de Likert en evaluación de calidad.....	82
Tabla 25: Análisis de herramientas SEO.....	83
Tabla 26: Escala de Likert en evaluación de calidad.....	83
Tabla 27 Análisis de riesgos: RG-1	94
Tabla 28 Análisis de riesgos: RG-2.....	94
Tabla 29 Análisis de riesgos: RG-3.....	95
Tabla 30 Análisis de riesgos: RG-4.....	95
Tabla 31 Análisis de riesgos: RG-5.....	96
Tabla 32: Especificación de requerimiento funcional: RF-01	97

Tabla 33: Especificación de requerimiento funcional: RF-02.....	97
Tabla 34: Especificación de requerimiento funcional: RF-03.....	98
Tabla 35: Especificación de requerimiento funcional: RF-04.....	98
Tabla 36: Especificación de requerimiento funcional: RF-05.....	98
Tabla 37: Especificación de requerimiento funcional: RF-06.....	99
Tabla 38: Especificación de requerimiento funcional: RF-07.....	99
Tabla 39: Especificación de requerimiento funcional: RF-08.....	99
Tabla 40: Especificación de requerimiento funcional: RF-10.....	100

ÍNDICE DE ILUSTRACIONES

Figura 1 Arquitectura del software	20
Figura 2 Estructura de fundamentación teórica del prototipo	32
Figura 3 Estructura de fundamentación teórica del prototipo	34
Figura 4 Cronograma de actividades	48
Figura 5 Diagrama de Gantt.....	48
Figura 6 Modelo relacional de la base de datos	50
Figura 7 Modelo navegacional: usuario administrador	51
Figura 8 Modelo navegacional: usuario final	52
Figura 9 Prototipo de interfaz: Inicio de sesión.....	53
Figura 10 Prototipo de interfaz: menú principal	54
Figura 11 Prototipo de interfaz: perfil de usuario	54
Figura 12 Prototipo de interfaz: listado de usuario	55
Figura 13 Prototipo de interfaz: ingreso y edición de usuario	56
Figura 14 Prototipo de interfaz: ingreso y edición de usuario	56
Figura 15 Prototipo de interfaz: listado de clientes	57
Figura 16 Prototipo de interfaz: ingreso de nuevo cliente	58
Figura 17 Prototipo de interfaz: visualización de cliente	58
Figura 18 Prototipo de interfaz: listado de créditos	59
Figura 19 Prototipo de interfaz: visualización de detalle de crédito	60
Figura 20 Prototipo de interfaz: apertura de nuevo crédito.....	61
Figura 21 Prototipo de interfaz: vinculación con contrato	61
Figura 22 Prototipo de interfaz: panel de control.....	62
Figura 23 Prototipo de interfaz: ingreso de nuevo pago	63

Figura 24 Prototipo de interfaz: ingreso de nuevo abono	63
Figura 25 Prototipo de interfaz: Reporte de pagos diarios	64
Figura 26 Prototipo de interfaz: Histórico de pagos por cliente	64
Figura 27 Prototipo de interfaz: Reporte de cartera general vencida.....	65
Figura 28 Prototipo de interfaz: Reporte de cartera vencida por cliente	65
Figura 29 Prototipo de interfaz: listado de roles	66
Figura 30 Prototipo de interfaz: Ingresar nuevo rol	66
Figura 31 Prototipo de interfaz: configuración de cartera	67
Figura 32 Prototipo de interfaz: configuración del sistema	67
Figura 33 Caso de uso generalizado	68
Figura 34 Implementación del prototipo: Página principal	69
Figura 35 Implementación del prototipo: Inicio de sesión	70
Figura 36 Implementación del prototipo: Página home	70
Figura 37 Implementación del prototipo: Lista de usuarios	71
Figura 38 Implementación del prototipo: Habilitar usuario.....	71
Figura 39 Implementación del prototipo: Visualizar información de usuario	72
Figura 40 Implementación del prototipo: Ingreso de nuevo usuario	72
Figura 41 Implementación del prototipo: Listado de clientes	73
Figura 42 Implementación del prototipo: Información del cliente	73
Figura 43 Implementación del prototipo: Ingresar nuevo cliente	74
Figura 44 Implementación del prototipo: Listado de contratos	74
Figura 45 Implementación del prototipo: Aperturar nuevo crédito	75
Figura 46 Implementación del prototipo: Detalles de nuevo contrato	76
Figura 47 Implementación del prototipo: Confirmación de contrato.....	76
Figura 48 Implementación del prototipo: Ingresar abono	77
Figura 49 Implementación del prototipo: Ingresar pago	78
Figura 50 Implementación del prototipo: Ejemplo de comprobante de pago	78
Figura 51 Implementación del prototipo: Reporte de pagos diarios.....	79
Figura 52 Implementación del prototipo: Histórico de pagos por cliente.....	79
Figura 53 Implementación del prototipo: Panel administrativo	80
Figura 54 Implementación del prototipo: Reporte general de cartera vencida	80
Figura 55 Implementación del prototipo: Reporte de cartera vencida por cliente ...	81
Figura 56 Implementación del prototipo: Diseño responsivo	81
Figura 57 Prueba del sistema: HTML Validator.....	85

Figura 58 Prueba del sistema: GTMetrix.....	86
Figura 59 Prueba del sistema: Functional Accessibility Evaluator	87
Figura 60 Prueba del sistema: Website Grader.....	88
Figura 61 Prueba del sistema: SEO Site Checkup.....	88
Figura 62 Estructura de desglose de trabajo.....	101
Figura 63 Esquema base de datos: Usuarios.....	102
Figura 64 Esquema base de datos: Clientes.....	103
Figura 65 Esquema base de datos: Contratos	104
Figura 66 Esquema base de datos: Creditos	105
Figura 67 Esquema base de datos: Entradas	106
Figura 68 Esquema base de datos: Pagos	107
Figura 69 Esquema base de datos: Notificaciones	108
Figura 70 Esquema base de datos: Roles	108
Figura 71 Esquema base de datos: Permisos.....	108
Figura 72 Diagrama de caso de uso: Gestión de usuarios.....	109
Figura 73 Diagrama de caso de uso: Gestión de clientes	109
Figura 74 Diagrama de caso de uso: Gestión de roles y permisos.....	110
Figura 75 Diagrama de caso de uso: Configuración del sistema.....	110
Figura 76 Diagrama de caso de uso: Configuración de cartera.....	111
Figura 77 Diagrama de caso de uso: Gestión de pagos.....	111
Figura 78 Diagrama de caso de uso: Gestión de cartera	112

INTRODUCCIÓN

El auge del desarrollo de sistemas de información orientados a la web, y su inclusión en los diferentes ámbitos ha generado una gran demanda dentro del desarrollo de software. La inclusión de los sistemas web dentro del área empresarial se ha visto en crecimiento gracias a los beneficios generados dentro de las diferentes operaciones del negocio, desde el área operativa hasta el área administrativa y gerencial [1]. Las necesidades actuales de las diferentes organizaciones han incrementado el uso de las TIC [2] y principalmente de los sistemas de información orientados a la web, generando un desafío a los desarrolladores para diseñar sistemas capaces de satisfacer las necesidades para las cuales fueron solicitados, incluyendo las características propias de las aplicaciones web, como la usabilidad, accesibilidad, seguridad de información, y estándares de diseño dentro de las interfaces de usuario.

Dentro del ámbito empresarial, el control de cartera es un proceso que requiere un seguimiento constante, y su optimización mediante la tecnología es crucial para el negocio [3]. El control adecuado de los adeudos de los clientes dentro de una organización, permite que esta se mantenga estable en cuanto a los balances contables. IBERCONSA al ser una empresa que efectúa créditos, o montos a pagar por los clientes a largo plazo, requiere llevar un control y seguimiento de las cancelaciones o expiraciones que se efectúen al adeudo, dicho control debe ser optimizado, lo cual evite que se generen errores de cálculo en los balances mensuales, y sobre todo que la empresa en determinado momento pueda llegar a tener pérdidas.

En base a lo mencionado, el presente trabajo tiene como finalidad diseñar e implementar un sistema administrativo modular web para la gestión de cartera de la empresa IBERCONSA mediante el proceso de desarrollo de la metodología SWIRL. Para el diseño y análisis del sistema se ha considerado las actividades relevantes dentro de la metodología de desarrollo SWIRL. Por otro lado, para la implementación se consideraron diversas tecnologías: para el lado del cliente se empleó Angular.js y estándares HTML, y CSS; mientras que para el lado del servidor se empleó Node.js, express.js, y como gestor de base de datos MongoDB. El resto del documento se encuentra estructurado de la siguiente manera:

Capítulo I: Centra su enfoque en el análisis realizado al objeto de estudio, detallando el ámbito de aplicación bajo el cual se va a implementar, la elicitación de los requerimientos a satisfacer, y el análisis y justificación de los mismos.

Capítulo II: Esta sección se centra en el proceso de desarrollo del prototipo, indicando los componentes, arquitectura y tecnologías a emplear dentro del modelo, los cuales son fundamentados de manera teórica. Dentro de este capítulo se definen los objetivos y se detallan los diseños y proceso de desarrollo del sistema, para lo cual se consideran algunas de las actividades de las fases de la metodología SWIRL, como el análisis, planificación y diseño.

Capítulo III: En este último capítulo, se realiza una evaluación del prototipo ya desarrollado, para ello se consideran las actividades inmersas en la fase de prueba y control empleando herramientas y estándares de evaluación. Finalmente, se detallan las conclusiones y recomendaciones acorde al proceso desarrollado.

1 CAPÍTULO I. DIAGNÓSTICO DE NECESIDADES Y REQUERIMIENTOS.

1.1 ÁMBITO DE APLICACIÓN: DESCRIPCIÓN DEL CONTEXTO Y HECHOS DE INTERÉS.

IBERCONSA, es una empresa constructora ubicada en la ciudad de Machala, encargada de la construcción y venta de viviendas individuales, y venta de terrenos. Los procesos de venta son realizados mediante créditos a largo plazo, asegurados por un contrato legalizado, llevando un control permanente de los adeudos que posee cada cliente al cual se le ha otorgado un crédito. La cartera de clientes es el registro mediante el cual se controla la actividad crediticia de los mismos permitiendo evidenciar el avance de los pagos del adeudo a lo largo del tiempo acordado, generados por un crédito emitido por la empresa gracias a la venta de un bien o servicio [4].

La empresa lleva una gestión de la cartera de clientes mediante dos áreas, el área de finanzas, que es la encargada de generar y consolidar el contrato del crédito, y por otro lado el área de cobranzas, quienes son los encargados de registrar los pagos de las cuotas de la deuda y realizar un balance mensual para estipular los clientes en mora durante el periodo analizado. El proceso de forma general contempla la generación de contratos, el almacenamiento de la deuda por cada cliente según las cuotas y los tiempos acordados en el contrato y tabla de amortización, registro de pagos mensuales, análisis histórico de pagos por clientes para su respectiva notificación, análisis de pagos diarios, y análisis de cartera vencida general y por cliente,

Bajo este punto es importante considerar varios posibles escenarios, como son la cancelación completa del crédito en un menor tiempo que lo acordado; mora de pago de uno o varios adeudos generando un valor agregado de interés por mora, y finalmente el registro de abonos en los pagos mensuales, y la emisión de comprobantes de pagos. Es importante mencionar, que es requerida la gestión de campos adicionales, como son, la generación de contratos, y la gestión de la información del cliente para lograr un correcto control de los créditos.

La empresa actualmente no cuenta con un sistema de información, de tipo escritorio o web, que permita gestionar tanto la información de clientes, créditos y pagos que se

realicen. La falta de optimización en los procesos de crédito puede generar inconsistencias dentro de las decisiones financieras y administrativas de la empresa [5], afectando indirecta o directamente los beneficios y solvencia de la organización.

Bajo lo mencionado, se propone el diseño de un sistema de información modular y administrable que permita el control de la cartera de clientes de la empresa IBERCONSA. El sistema se constituirá por módulos, permitiendo la posibilidad de un escalamiento y optimización de nuevas áreas en la organización. En la actualidad los procesos y actividades del negocio no son ejecutadas según los cargos que corresponda, debido a falta de personal. Es por ello que se diseña una estructura administrativa, mediante la cual sea posible asignar varios roles y permisos a cada usuario, de manera que se mantengan los recursos de la empresa dando posibilidad a inclusión de nuevos cargos a futuro.

Los módulos que abarcará el sistema incluyen la gestión de cartera, gestión de créditos (contratos, y clientes), gestión de usuarios, roles y permisos. Mediante la propuesta se busca optimizar los procesos, agilizando el ingreso de los pagos y verificación de los saldos pendientes bajo los escenarios provistos, además de la visualización de históricos de pagos, y reportes de cartera vencida, identificando los clientes en mora, el monto no cobrado por crédito durante el mes, y un listado con información personal de los clientes con adeudos para sus respectivas notificaciones.

1.2 ESTABLECIMIENTO DE REQUERIMIENTOS.

Basándose en las necesidades y procesos de negocio que se ejecutan dentro de la organización, IBERCONSA establece como requerimientos clave el registro de pagos y abonos de los adeudos, el análisis histórico de pagos por clientes, apertura de nuevos créditos y contratos, análisis de cartera vencida, el estado de crediticio según la tabla de amortización y de forma adicional y necesaria, la gestión de clientes.

Uno de los requerimientos solicitados por la empresa, es la importancia de expansión del sistema, permitiendo en un futuro agregar nuevas funcionalidades para la optimización de procesos de negocio de otras áreas. Es por ello, que se ha considerado propicio el diseño de un sistema modular, desarrollando en la presente propuesta los módulos requeridos. El uso de software modular ayuda a que el sistema sea flexible, adaptable y reutilizable, en alguna de sus funciones [6]. Para lograr la

identificación correcta de los requerimientos, se emplean los principios básicos de la ingeniería de requisitos, la cual se centra en analizar las actividades necesarias para el cliente y para el usuario final [7], considerando la conducta externa del sistema y las restricciones delimitadas por el cliente.

La propuesta tecnológica se diseña con la finalidad de satisfacer los requerimientos establecidos acorde a sus respectivos módulos: Dentro de la gestión de clientes, el sistema debe ser capaz de crear y editar la información de un cliente, y al ser eliminado se eliminarán todos los contratos y pagos relacionados al mismo, esta acción solo es ejecutada mediante permisos. Para la visualización de la información del cliente, se detallará tanto la información de contacto como un listado de todos los contratos otorgados y su estado actual acorde a la tabla de amortización.

Los requerimientos inmersos para el proceso de gestión de cartera son la apertura de nuevos créditos y la generación respectiva del contrato en formato pdf; el ingreso de los pagos y abonos a las cuotas de forma mensual, incluyendo la impresión del comprobante de pago; visualización del histórico de pagos por cliente con el respectivo almacenamiento de las notificaciones realizadas; visualización de cartera vencida (general y por cliente) y listado de pagos diarios. El sistema permite visualizar un panel para el área administrativa con un resumen mensual de los créditos que incluyen monto por cobrar en el mes, ingreso mensual en las cuentas de banco y caja, listado de cuotas a pagar durante el mes (informando las cuotas vencidas y las vigentes) y el listado de cuotas vencidas hasta la fecha.

Adicionalmente, se incluye como requerimiento la gestión de campos relacionados en los procesos y que son definidos por la empresa cargos, roles, formas de pago, entidades bancarias, entre otros. Para la especificación, análisis y documentación de los requerimientos, se consideran las actividades y herramientas propuestas por la metodología de desarrollo SWIRL, y las necesidades emitidas por la empresa IBERCONSA, estableciendo de esta forma los requerimientos funcionales y no funcionales en los próximos apartados.

1.3 JUSTIFICACIÓN DEL REQUERIMIENTO A SATISFACER.

La información es el bien más preciado que toda organización posee, siendo necesario implementar tecnología y mecanismos que permitan su seguridad,

recolección y análisis. Dentro de la empresa IBERCONSA, el control de cartera es un proceso principal, ya que la prestación de los servicios de construcción genera contratos con adeudos en los clientes a largo plazo (superiores a los 5 años), motivo por el cual, el seguimiento del pago mensual de las cuotas es un punto clave para llevar un orden administrativo y contable dentro de la misma. Empleando un sistema administrativo modular para el control de cartera basado en la arquitectura cliente-servidor [8], es posible llevar un seguimiento de los, pagos, cuotas vencidas y por pagar que tiene cada uno de los clientes, a los cuales en cierto momento se les han efectuado un crédito.

El proceso de cartera conlleva una relación directa con otros procesos administrativos, como son la generación de contratos, la gestión de clientes, y la gestión de campos independientes necesarios para un control específico de los pagos mediante las diferentes entidades bancarias en casos de depósitos. El desarrollo del sistema mediante módulos, beneficia directamente a la empresa que lo empleará, ya que será posible la inclusión de nuevas funcionalidades para optimizar procesos de negocio de otras áreas sin la necesidad de modificar completamente el sistema de información ya diseñado. En base a lo mencionado y a la necesidad de información relevante de los clientes, y usuarios para gestionar adecuadamente el control de los créditos, se ha estructurado el sistema de información en los siguientes módulos: gestión de créditos, control de créditos, gestión de clientes y gestión de usuarios.

Dentro de la gestión de clientes, es relevante analizar el detalle crediticio actual al igual que el análisis de factores básicos para la cobranza, como son listado de clientes pendientes de pago, la información de contacto, y el monto recaudado para su respectivo análisis.

Al analizar los requerimientos de la empresa, y las condiciones bajo las cuales el personal realizan los procesos de negocio, se diseña un sistema administrativo, basado en roles y permisos, de manera que se puedan agregar nuevos roles según la organización y asignar los permisos que cada rol va a poseer dentro del sistema. Gracias a lo mencionado, el sistema es capaz de ser empleado en diferentes áreas de la empresa. Debido a las funciones a implementar, es importante definir los mecanismos, tecnologías, y herramientas que permitan satisfacer los procesos de

manera eficiente, por lo que se consideran las herramientas y fases que la metodología SWIRL propone, y se detallan en el siguiente capítulo.

2 CAPÍTULO II. DESARROLLO DEL PROTOTIPO

2.1 DEFINICIÓN DEL PROTOTIPO TECNOLÓGICO.

En la presente sección se define la estructura base para desarrollar la propuesta, incluyendo la arquitectura del software, la fundamentación teórica, la definición del sistema, los interesados, requerimientos funcionales y otras actividades incluidas en la fase de análisis de la metodología.

2.1.1 Arquitectura del software.

La arquitectura del proyecto a desarrollar se basa en la arquitectura cliente-servidor, permitiendo una adecuada comunicación bidireccional entre dos aplicaciones destinadas para el front-end y el back-end [9]. Se considera el lado del cliente los dispositivos mediante los cuales se accederá a la información, mientras que en el lado del servidor se establece el procesamiento de la información y su comunicación con la base de datos. Para facilitar el diseño del sistema se divide la arquitectura según el modelo de tres capas: presentación, procesos y de datos [10] como se muestra en la *Figura 1*

En el primer nivel se encuentra la capa de presentación, implementada dentro del lado del cliente y la cual es accedida por los diferentes usuarios mediante un navegador web y realiza las respectivas peticiones de datos y procesamiento de información al segundo nivel. En el segundo nivel se encuentra la capa de procesos el cual se encarga de recibir las peticiones del cliente y brindar acceso a la información solicitada, para ello hace uso de la capa de datos. El ultimo nivel es la capa de datos, en la cual se encuentra el gestor de base de datos que almacenará la información y la emitirá al nivel dos para ser visualizada en la capa de presentación.

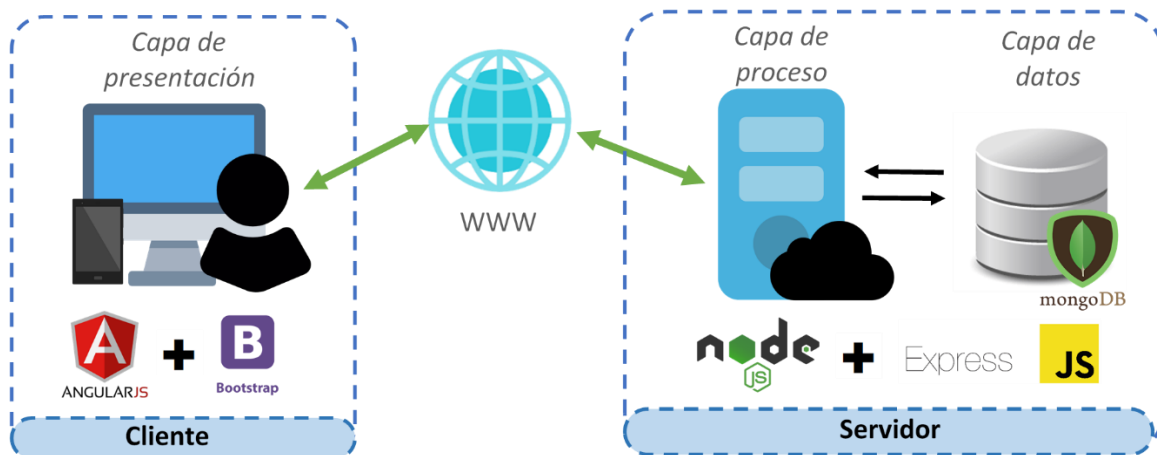


Figura 1 Arquitectura del software
Fuente: Elaboración propia

Capa de presentación: Esta capa incluye las interfaces del programa que el usuario utilizará para la visualización y gestión de la información, así como el uso de las funcionalidades del sistema. La interfaz es accesible mediante un navegador web, y el diseño de las mismas son desarrolladas mediante el framework de Angular.js, y estándares Bootstrap y CSS.

Capa de procesos: Dentro de este se reciben las peticiones emitidas por el cliente, y devuelve la información según sea el proceso solicitado. Esta capa también se comunica con la tercera capa, la cual es la capa de datos. El desarrollo del api-rest se realiza mediante el lenguaje Node.js, y para la comunicación entre las aplicaciones cliente-servidor se emplea Express.js.

Capa de datos: Esta capa es la encargada de gestionar los registros contenidos dentro de la base de datos, respondiendo a las peticiones que emite el api-rest. Para la presente propuesta se emplea como gestor de base de datos Mongo DB, debido a la facilidad de almacenamiento de archivos, grande cantidad de datos, funcionalidad asíncrona, entre otras características que posee el proyecto.

2.1.2 Definición del sistema.

Una vez que se ha establecido la arquitectura de desarrollo del proyecto, se procede a definir las funcionalidades principales, los requerimientos funcionales y no funcionales del sistema, y demás componentes necesarios para establecer las características relevantes.

2.1.2.1 Reconocimiento general del sistema.

El sistema tiene como funcionalidad principal la gestión de la cartera de clientes de la empresa IBERCONSA, implementando los procesos de pago de cuotas, ingreso de clientes, apertura de crédito, y visualización de listados de estado de cartera para las actividades esenciales de cobranza. IBERCONSA, es una empresa dedicada a la construcción de viviendas, urbanizaciones, entre otras, incluyendo el crédito como una forma de pago. El sistema a implementar dentro de la empresa, permitirá llevar un mejor control de los clientes, el estado de sus deudas, y facilidad en la detección de clientes en mora para la debida notificación.

El sistema permitirá ingresar un nuevo cliente, registrando la información básica, la información del cónyuge (en caso de poseer), y la información laboral; dicha información permitirá contactarse con el cliente en caso de cuotas vencidas. Una vez registrado el cliente el sistema permitirá crear un nuevo crédito, ingresando los valores de entrada y el monto a diferir. Cada crédito se encuentra vinculado a un contrato, dentro del cual se almacena la información de beneficiarios en caso de fallecimiento, e información relevante del terreno a adquirir. El sistema permitirá la opción de efectuar el cálculo de la tabla de amortización de un crédito sin necesidad de crear contratos, o asignar un cliente, sin embargo, el crédito no será guardado ya que es solo para impresión.

Un punto importante a resaltar, es que el sistema debe ser administrable y modular. Al ser administrable, será posible adaptarlo a diferentes empresas, que requieran un sistema para el control de deudas y pagos de los clientes; además, las funcionalidades disponibles serán acorde a los permisos que otorga el administrador a cada empleado. Es considerado modular, debido a que se implementará mediante módulos, para en un futuro poder optimizar nuevos procesos dentro de la empresa sin necesidad de efectuar cambios radicales.

2.1.2.2 Estudio de factibilidad.

Dentro del estudio de factibilidad se consideran la identificación de usuarios, los beneficios esperados con la implementación y el alcance que poseerá el sistema.

➤ Identificación de Usuarios del Sistema

Pese a que el sistema cuenta con la creación de roles según el proceso de negocio de la organización, es necesario establecer usuarios fijos a los cuales se le asignarán actividades específicas dentro del sistema, estos tipos de roles no podrán ser modificados ni eliminados de la aplicación. En base a lo mencionado, el sistema cuenta con dos tipos de usuario: el administrador (usu-admin) y los usuarios finales (usu-final), clasificados según sus características y actividades dentro del sistema como usuarios principales y secundarios. A continuación, se detallan cada uno de ellos:

Usu-admin: Es el usuario con la posibilidad de acceder a todas las funcionalidades del sistema sin restricción alguna, visualización de todas las tablas, y ejecución de todas las actividades como eliminar o modificar campos sensibles. Es importante recalcar que a pesar que es posible agregar nuevos usuarios de tipo administrador, el programa cuenta con un usuario administrador por defecto el cual no es posible modificar, ni eliminar.

Tabla 1: Usuarios del sistema: U-Admin

Usuario:	Usu-admin
Tipo de usuario:	Principal.
Nivel:	Primer nivel.
Tipo de interacción:	Directa – Parcial
Actividades:	<ul style="list-style-type: none">- Ingresar al sistema.- Gestionar clientes.- Gestionar usuarios, asignar roles y permisos.- Gestionar roles y permisos por defecto.- Gestionar configuración de campos del sistema.- Gestionar configuración de cartera.- Gestionar créditos (Visualizar, crear, eliminar, modificar)- Gestionar área de cartera (apertura de crédito, ingreso de pagos.)- Acceso al panel administrativo.

	- Visualización de todos los reporte de gestión de cartera (carteras vencidas, históricos de pagos, y pagos diarios).
Contenido de interacción:	Acceso a todas las áreas y funcionalidades del sistema.
Descripción:	Administrador del sistema, el cual tiene acceso completo a todas las funcionalidades del sistema, para la configuración y gestión de módulos, permisos por defecto, entre otras configuraciones específicas del sistema. Este tipo de usuario no puede ser modificado ni eliminado dentro de los roles y permisos.

Fuente: Elaboración propia.

Usu-final: Este tipo de usuario puede ser clasificado independientemente por la estructura organizacional de la empresa. Debido a que el sistema a diseñar es administrativo y los cargos dentro de las organizaciones son variadas, los usuarios finales ejecutan las actividades que son asignadas según los permisos, y roles a los que pertenecen.

Tabla 2: Usuarios del sistema: U. Final

Usuario:	Usu-Final
Tipo de usuario:	Secundario.
Nivel:	Primer nivel.
Tipo de interacción:	Directa – Temporal
Actividades:	<ul style="list-style-type: none"> - Ingresar al sistema. - Gestionar clientes. - Ver perfil de usuario. - Gestionar créditos (Visualizar, crear) - Ingresar nuevos pagos y abonos. - Visualizar resultados en panel administrativo. - Visualización de reportes de cartera vencida, y pagos diarios. - Visualización de histórico de paos de clientes. - Ingresar notificaciones de clientes.

Contenido de interacción:	<ul style="list-style-type: none"> - Inicio de sesión. - Visualización de perfil. - Página de panel administrativo de cartera. - Páginas de gestión de clientes. - Página de créditos (listar y crear). -Página de ingreso de pago y abono. - Página de listado de pagos diarios. <p>Páginas de gestión de cartera (cartera vencida e histórico de pagos).</p>
Descripción:	Empleados de la empresa, las actividades asignadas para cada uno dependerán de los permisos que el administrador le habilite según el rol que desempeñen.

Fuente: Elaboración propia.

➤ **Beneficios esperados.**

Con la implementación del sistema se pretende que la empresa IBERCONSA obtenga beneficios visibles de manera monetaria (beneficios tangibles), y también ventajas en los procesos de la empresa (beneficios intangibles). En la *Tabla 3* Se presentan los beneficios tangibles e intangibles que se pretenden obtener con la implementación del sistema de control de cartera dentro de la organización.

Tabla 3: Beneficios esperados

Beneficios Tangibles	Beneficios Intangibles
<ul style="list-style-type: none"> • Reducción de tiempo en el cálculo de monto de deuda mensual. • Optimización del proceso de registro y control de pagos de deudas de los clientes. • Reducción de tiempo en la elaboración de contratos. 	<ul style="list-style-type: none"> • Rápido acceso a la información para el análisis de saldos de clientes. • Incremento en la efectividad del servicio de pago de créditos. • Incremento en el control de deudas vencidas y por pagar.

Fuente: Elaboración propia.

➤ **Alcance del proyecto.**

En base al análisis realizado se debe establecer el alcance del proyecto, indicando las funcionalidades y las limitaciones con las que el sistema cuenta. En cuanto a las funcionalidades del sistema se tienen las siguientes:

- Permite la autenticación del usuario mediante el inicio de sesión.
- Permite la gestión de usuarios (ingresar, modificar, visualizar, e inhabilitar).
- Permite la gestión de clientes (ingresar, modificar, visualizar y eliminar).
- Permite la asignación de permisos individualmente por usuario.
- Permite la gestión de roles (ingresar, modificar, visualizar, eliminar, y asignar permisos por defecto).
- Permite la creación y visualización de créditos (la eliminación solo se permite para el administrador).
- Permite la vinculación de un crédito a un nuevo contrato.
- Permite la visualización, e impresión de contratos.
- Efectuar el pago de una cuota mensual de uno o varios clientes.
- Efectuar el pago de uno o más abonos en cada cuota.
- Permitir cancelar por completo la deuda en clientes que se encuentren al día en pagos.
- Visualización de reportes de cartera vencida (general y por cliente).
- Listado de pagos diarios.
- Visualización de histórico de pagos por cliente.

Es necesario establecer de la misma forma las limitaciones que el sistema poseerá basándose en los requerimientos iniciales, funcionalidades, y objetivos del sistema. Algunas de las limitaciones que posee son:

- No genera reportes en el ámbito de cobranza, ni reportes que no se hayan especificado por la empresa.
- Las plantillas para la creación e impresión de contratos no pueden ser modificados.
- No es posible modificar un crédito, este debe ser eliminado y reingresado. (para la etapa de migración de la información y solo con permisos específicos).
- No es posible eliminar un usuario solo inhabilitarlo.

- No es posible desvincular un contrato de un crédito.

2.1.2.3 Análisis de factibilidad

Para identificar si el desarrollo e implementación del sistema es factible, se debe analizar la propuesta en el ámbito operativo,

➤ Factibilidad Operativa

El análisis de la factibilidad operativa presentado en la *Tabla 4*, se realiza con la finalidad de identificar si la empresa posee las condiciones necesarias para que el sistema pueda ser implementado.

Tabla 4: Factibilidad operativa

N°	Actividad	Priorización
1	El sistema requiere ser alojado en un host en la nube.	Medio
2	Los usuarios requieren capacitación para el manejo del sistema.	Medio
3	El sistema requiere conectividad a internet para su funcionamiento.	Alta

Fuente: Elaboración propia.

En base al análisis de la tabla presentada, se considera que la empresa se encuentra en condiciones para implementar el sistema, ya que no es necesario personal con completo entendimiento del sistema al ser fácil de operar.

➤ Factibilidad técnica

Para que el sistema sea implementado correctamente, es necesario contar con un servidor o adquirir un servicio de hosting en la nube. Existen dos tipos de hosting, los gratuitos en los cuales es posible alojar el sistema considerando que la capacidad de almacenamiento será menor y podría afectar en un futuro si los registros de la empresa incrementan exponencialmente. Por otro lado, los servicios de hosting pagados, los cuales tienen un precio aproximado de \$5 al mes. En ambas opciones, es factible técnicamente implementar el sistema en la empresa.

➤ Factibilidad económica

A continuación, se detallan los recursos tecnológicos empleados para el desarrollo del sistema. Es importante mencionar, que la propuesta al ser producto del proceso de titulación, no se consideran gastos en recursos humanos.

Tabla 5: Factibilidad económica

Recursos Tecnológicos		
Hardware		
Cant.	Descripción	Total (\$)
1	Computador Core I7 - 16RAM - 256 SSD	\$1200
Software		
Cant.	Descripción	Total (\$)
1	Visual Studio Code	\$0
1	Mongo en la nube	\$0
1	Host en la nube gratuito	\$0
Costo total de recursos tecnológicos		\$1200

Fuente: Elaboración propia.

2.1.2.4 Perspectiva del producto

El sistema web diseñado mediante la propuesta tecnológica será implementado dentro de la empresa IBERCONSA para el control y análisis de la cartera de crédito de clientes. La inclusión del sistema como herramienta para el proceso de negocio mencionado, optimizará los resultados obtenidos en cuanto al seguimiento de los adeudos de los clientes, al permitir el registro de los pagos de cuotas, la creación de nuevos contratos, y la visualización del seguimiento y estado del crédito que cada cliente posee, los cuales pueden ser empleados para análisis administrativo por parte del gerente.

Se estima que con la implementación del sistema las actividades se realicen en menor tiempo, y principalmente el control del ingreso a caja y los montos de las deudas por cobrar se lleve de manera ordenada y segura, con la mínima posibilidad de errores. Debido a que la empresa cuenta con sus cargos y roles específicos para cada

empleado, se diseña un sistema administrativo, mediante el cual el administrador es capaz de asignar los roles y permisos a cada usuario/empleado, según el papel que desempeñan en la organización.

➤ **Funcionalidad del sistema**

Al ser un sistema modular es necesario explicar cada uno de los módulos del sistema de control de cartera y las actividades que efectuará cada uno, como se presenta en la *Tabla 6*.

Tabla 6: Módulos del sistema.

#	Módulo	Actividades
1	Usuarios	Gestión de usuarios del sistema, lo cual incluye creación, visualización, modificación e inhabilitación. Dentro del mismo se maneja la asignación de permisos y roles propios de cada usuario, así como la habilitación e inhabilitación de las cuentas. La habilitación solo puede ser realizada por el administrador.
2	Clientes	Gestión de clientes, incluyendo creación, visualización, modificación y eliminación de clientes (este último elimina los contratos pagos vinculados).
3	Cartera	<p>Gestión de crédito: el cual incluye la creación, visualización, impresión de tabla de amortización, y eliminación de un crédito.</p> <p>Gestión de contratos: incluye la visualización e impresión. La creación se realiza mediante la vinculación al crear un nuevo crédito, y la eliminación solo se ejecuta al momento de eliminar un crédito.</p> <p>Gestión de pagos: incluye el ingreso de un nuevo pago, ingreso de abonos, impresión de comprobante de pago, cancelación completa de la deuda, y visualización de reporte de pagos diarios.</p> <p>Dashboard: Visualización de listado de clientes en mora, clientes por cobrar el mes actual, el ingreso mensual de los adeudos, y el valor estimado a cobrar dentro del mes.</p>

#	Módulo	Actividades
		Gestión de cartera: Incluye la visualización de la cartera vencida general y por cliente, y visualización del histórico de pagos del cliente.
4	Roles y Permisos	Gestión de roles, incluyendo creación, visualización, modificación y eliminación de roles. Asignación de permisos por defecto en los roles, no se permite ingresar, editar o eliminar los permisos.

Fuente: Elaboración propia.

El módulo de cartera, en el que se basa principalmente el programa, es el más extenso debido a la inclusión de actividades para diversas áreas como son los créditos, contratos, pagos, y panel administrativo.

2.1.2.5 Análisis de riesgos

Para lograr minimizar el impacto y ejecución de los riesgos durante el desarrollo del proyecto, se realiza una definición y análisis de los posibles riesgos, considerando la etapa bajo la cual pueden ser provocados, y el grupo del riesgo al cual pertenece. La presenta un análisis cualitativo y cuantitativo de los riesgos, los cuales han sido definidos y analizados en el **ANEXO A**.

Tabla 7: Análisis cualitativo y cuantitativo de riesgos.

Grupo de riesgos	Riesgo No	Análisis cuantitativo		Análisis cualitativo	
		Probabilidad	Impacto	Magnitud	Prioridad
Riesgos técnicos, de calidad o rendimiento	RG-3	50%	0.4	Leve	Baja
	RG-4	77.5%	0.8	Severo	Muy Alta
Riesgos en la gerencia de proyectos.	RG-2	52.5%	0.63	Moderado	Alta
	RG-5	45%	0.68	Moderado	Baja
Riesgos del sistema en relación con el usuario	RG-1	35%	0.78	Severo	Baja

Fuente: Elaboración propia.

2.1.3 Identificación de interesados

Los interesados o también denominados stakeholders, son las personas afectadas directa o indirectamente por el desarrollo del proyecto [11]. Dentro de la propuesta se definen tres interesados los cuales se detallan en la *Tabla 8*

Tabla 8: Matriz general de interesados.

Matriz de interesados			
Nombre	Rol	Profesión	Responsabilidad
Ing. Oscar Cárdenas	Gerente del proyecto	Ing. de Sistemas	Revisión de documentación, entregables, y pruebas.
Bárbara Romero Macharé	Desarrollador	Alumno	Realizar el diseño, planificación, implementación y pruebas del sistema.
Francisco Medina	Propietario y gerente de IBERCONSA	Ingeniero civil	Emisión de requerimientos y proceso del negocio.

Fuente: Elaboración propia.

2.1.4 Requerimientos

Para la elicitación de requerimientos se emplea la técnica de entrevista y conversación con el gerente de la empresa IBERCONSA, además de obtener plantillas y documentación manual de cómo se efectúan las actividades en los procesos a optimizar.

2.1.4.1 Requerimientos funcionales.

La definición de requerimientos funcionales (RF) presentada en la *Tabla 9* se realiza según los módulos y procesos de negocio, sin embargo cada módulo cuenta con especificaciones diferentes, ya que no todas las actividades cuentan con un proceso de CRUD básico (Crear, Seleccionar, Actualizar, Eliminar) [12]. Es por ello que se realiza un desglose de los requerimientos en el **ANEXO B**, presentado las actividades a incluir en el sistema para cada módulo.

Tabla 9: Matriz general de interesados.

Código	Requerimiento
RF-01	Autenticación del usuario mediante el inicio de sesión (validación de usuario y contraseña).
RF-02	Gestión de los usuarios, ingreso, asignación de permisos, modificación, visualización, y bloqueo de cuenta.
RF-03	Gestión de clientes (ingresar, modificar, eliminar y visualizar).
RF-04	Gestión de roles y permisos del sistema.
RF-05	Gestión de crédito, creación, eliminación e impresión.
RF-06	Gestión de contratos, visualización e impresión.
RF-07	Ingreso de pagos, ingreso de abonos a la cuota mensual, cancelación completa de la deuda, visualización de reporte de pagos diarios.
FR-08	Visualización de un panel administrativo para la gestión de cartera de los clientes (visualización de cuentas en mora, cuentas al día, etc).
RF-09	Visualización de reportes de gestión de cartera vencida e históricos de pago.

Fuente: Elaboración propia.

2.1.4.2 Requerimientos no funcionales.

Los requerimientos no funcionales (RNF) son los que proveen las especificaciones del producto, características de seguridad, calidad, y otros aspectos relevantes para que el sistema se ejecute de manera satisfactoria [13] y cumpla con lo requerido por el cliente.

Tabla 10: Matriz general de interesados.

Código	Requerimiento
RNF-01	Las modificaciones podrán ser realizadas según los permisos que se le asigne a cada usuario.
RNF-02	Solo deben ser visibles las funcionalidades vinculadas a los permisos de cada usuario.
RNF-03	Los registro por defecto en las tablas de roles, permisos, usuarios, entre otros solo serán gestionados por el administrador.

Código	Requerimiento
RNF-04	La base de datos debe ser escalable para los posibles nuevos módulos.
RNF-05	El sistema web debe poseer una estructura modular, y administrable.
RNF-06	Los datos sensibles deben almacenarse de manera encriptada.
RNF-07	El diseño de la interfaz de usuario debe ser de fácil aprendizaje y cumpliendo los estándares básicos de usabilidad.

Fuente: Elaboración propia.

2.2 FUNDAMENTACIÓN TEÓRICA DEL PROTOTIPO.

En este apartado se define de manera teórica las tecnologías, herramientas y metodología seleccionada para desarrollar la propuesta tecnológica y que se desglosan en la *Figura 2*.

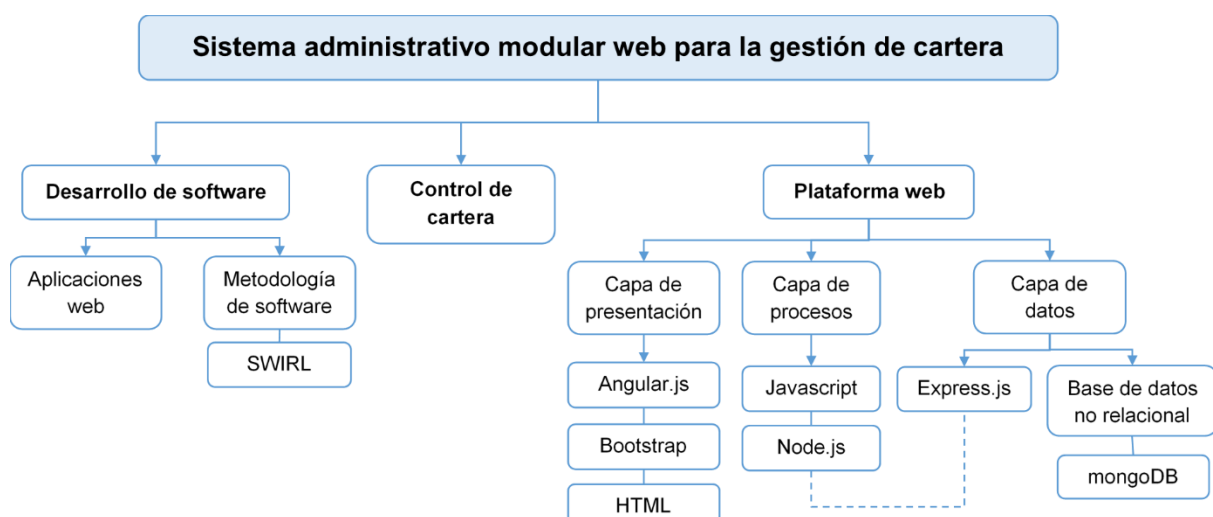


Figura 2 Estructura de fundamentación teórica del prototipo

Fuente: Elaboración propia.

2.2.1 Desarrollo de software.

El desarrollo de software se ha convertido en un factor de relevancia dentro del ámbito tecnológico, permitiendo la creación de sistemas y aplicaciones ya sean para plataformas, móviles, web o programas de escritorio. Dentro del proceso de desarrollo, se consideran varios factores y herramientas tecnológicas, sin embargo, tienen como principal objetivo entregar un producto de calidad que satisfaga las necesidades del cliente o usuario final.

Para lograr un correcto desarrollo de software y evitar fallas dentro durante la gestión de un proyecto es necesario incluir una metodología de desarrollo [14], estas deben ser consideradas según las necesidades del proyecto y del equipo de desarrollo. El uso de metodologías o modelos durante el desarrollo ayuda a un mejor entendimiento de las actividades y a una producción de calidad [15], incrementando las probabilidades de éxito del proyecto.

La ingeniería de software es la encargada de los procesos de desarrollo de un programa, sin embargo, con el pasar del tiempo y la importancia que han generado las aplicaciones web en el ámbito tecnológico, se ha especificado una rama separada para el desarrollo de aplicaciones basadas en la web, denominada ingeniería web [16].

2.2.1.1 Aplicaciones web.

Las aplicaciones con enfoque a la web presentan características específicas, como la rápida retroalimentación del usuario, la facilidad de cambios, la accesibilidad y fácil navegabilidad, y sobre todo una interfaz de usuario amigable y responsiva.

Las aplicaciones web son sistemas de carácter complejo, debido a las características que presentan, el contenido de las mismas puede ser estático o dinámico. Las aplicaciones emplean protocolos HTTP para la comunicación entre sus páginas, vistas e incluso la interacción con otras aplicaciones.

2.2.1.2 Metodología de software.

Un aspecto de relevancia dentro del desarrollo del software es el uso de metodologías o modelos [17], los cuales facilitan la guía y desarrollo de diferentes actividades a través del ciclo de vida del proyecto. En la actualidad existe una gran variedad de metodologías, las cuales poseen propiedades y características específicas, sin embargo centran su enfoque y objetivo en la misma finalidad, considerando etapas bases como: la planificación, elicitación de requerimiento, diseño, desarrollo o codificación, pruebas de calidad [18].

Las metodologías ágiles han constituido un proceso que beneficia tanto al producto como a los desarrolladores, las metodologías con enfoques ágiles se enfocan principalmente en el desarrollo a corto plazo, entregando en cada iteración un

producto de valor y así mantener el éxito del proyecto [19], además de contrastar los resultados y procedimientos de proyectos anteriores con los futuros [20].

➤ Metodología SWIRL.

SWIRL es una metodología híbrida, enfocada principalmente en el desarrollo de aplicaciones basadas en la web, mediante la combinación de un enfoque híbrido e iterativo [21]. El enfoque iterativo permite la entrega de resultados constantes a lo largo del proceso de desarrollo obteniendo como resultado final un producto eficiente y de calidad.

El ciclo de vida del proceso de desarrollo del proyecto se presenta en la *Figura 3* siendo las fases: análisis, planificación, modelado, implementación, verificación/pruebas y lanzamiento.

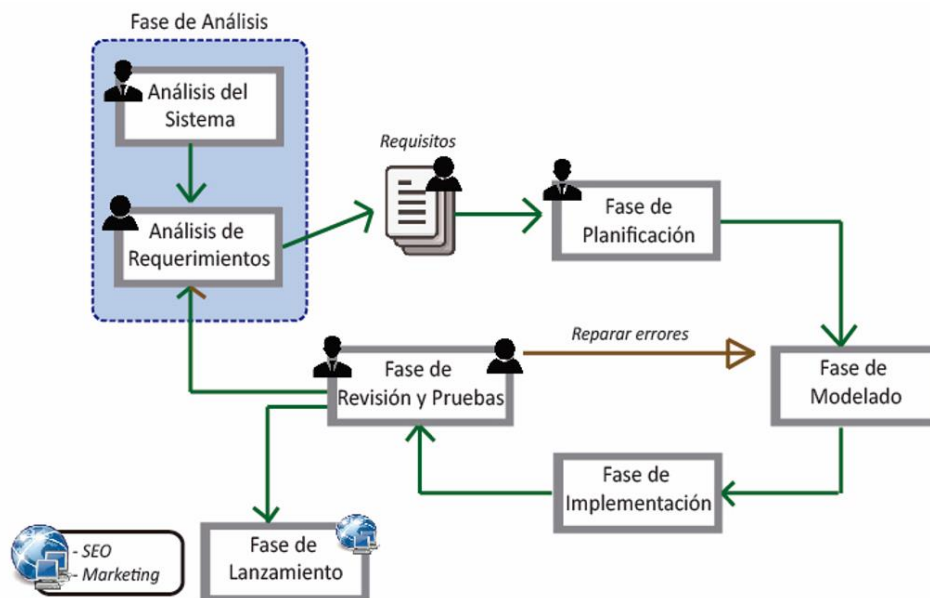


Figura 3 Estructura de fundamentación teórica del prototipo

Fuente: [21]

Características:

La metodología SWIRL se centra en un proceso iterativo incluyendo criterios de costo, tiempo, alcance, calidad y comunicaciones. Las iteraciones incluyen los diversos procesos de las fases, y su estructuración depende de las necesidades de cada proyecto.

Una de las características que presenta el modelo SWIRL es la inclusión constante del usuario, y su validación dentro de las iteraciones. Otras características son las siguientes:

- Acoplamiento dentro de proyectos grandes o pequeños.
- Reducción de recurso humano, de costo, y tiempo.
- Los interesados dentro de los procesos deben estar bien definidos.

Fases:

A continuación, se describen las fases que contempla la metodología SWIRL, indicando el objetivo y las actividades de cada una de ellas.

1) Fase de análisis.

Esta fase contempla una visión y análisis del sistema a desarrollar, con lo cual se lograrán definir los elementos, funcionalidades, y consideraciones claves para el proyecto. El principal objetivo de la fase de análisis es analizar la viabilidad del proyecto mediante la identificación de procesos necesarios del modelo de negocio. Las actividades consideradas dentro de esta fase son detalladas en la *Tabla 11*.

Tabla 11: Actividades fase de análisis.

Actividad	Sub-actividad
1. Definición del sistema	<ul style="list-style-type: none"> ● Reconocimiento general del sistema. ● Estudio de la factibilidad. ● Análisis de la factibilidad. ● Perspectiva del producto. ● Funcionalidad del sistema. ● Análisis de riesgos.
2. Modelado de negocio	<ul style="list-style-type: none"> ● Descripción del modelo de negocio. ● Evaluación ● Mejora/ Innovación (Opcional).
3. Identificación de interesados	<ul style="list-style-type: none"> ● Identificación de roles y stakeholders. ● Definición de las actividades.
4. Requerimientos	<ul style="list-style-type: none"> ● Elicitación. ● Análisis. ● Validación.

Fuente: Elaboración propia.

2) Fase de planificación.

Esta etapa se centra en la organización de las actividades conforme a un tiempo o cronograma determinado. El objetivo de esta fase es definir actividades y recursos que permitan desarrollar un producto de calidad. Las actividades inmersas dentro de la misma son:

- Historias de usuario
- Definición de entregables
- Gestión de cronograma, riesgos y comunicación
- EDT (Estructura de desglose de trabajo)
- Velocidad del proyecto y estimación de esfuerzo

3) Fase de modelado.

Dentro de esta fase se diseñan los diferentes modelos para el correcto desarrollo del sistema, aplicando los diagramas UML en los casos que se consideren necesarios. Las actividades dentro de la misma son:

- Diseño de modelo conceptual
- Diseño de modelo navegacional
- Diseño de interfaz de usuario
- Modelado de diagramas UML

4) Fase de implementación.

Durante la fase de implementación se realiza la codificación de cada una de las funcionalidades del sistema, dentro de esta etapa es importante considerar estándares de programación, los cuales no son obligatorios, pero ayudan a una interpretación adecuada del código fuente. Las actividades son:

- Codificación
- Estándares
- Definición e implementación de módulos
- Diseño y codificación de páginas principales
- Interconexiones

5) Fase de revisión y pruebas.

Se evalúan las funcionalidades desarrolladas en la fase anterior de manera que se asegure la calidad del producto final. El objetivo de esta fase es analizar fallos

o posibles errores en el funcionamiento del sistema. Las actividades de esta fase son:

- Control integrado de cambios
- Pruebas de integración
- Detección y corrección de errores
- Pruebas del sistema y de aceptación
- Control de alcance
- Cierre

6) Fase de lanzamiento y Marketing.

Esta etapa es realizada al finalizar el proyecto, es importante que se ejecute siempre y cuando el producto sea aprobado por el cliente. Las actividades dentro de esta fase son:

- Selección y alojamiento del hosting
- Preparación del dominio
- Configuración del certificado SSL (Opcional)
- Campaña de Marketing / SEO

2.2.2 Gestión de cartera.

Dentro de una organización el aseguramiento de su sustentabilidad en el tiempo es de suma importancia, garantizando las políticas de dividendos y liquidez [22]. Según la gestión de créditos es una actividad riesgosa dentro de la gestión de fondos que posee la organización para asegurar la liquidez y solvencia del negocio.

Bajo lo mencionado, es importante definir el término crédito, como una inversión de la organización a un cliente por la venta de un producto o servicio, con la finalidad de incrementar las ventas [23].

La cartera de crédito de un cliente, se basa en la gestión de las deudas ligadas a los créditos otorgados por un bien o servicio. El control de cartera incluye la gestión de los pagos, el análisis de los mismos para estimar los riesgos de liquidez, o de clientes en mora, también se emplean sistemas y modelos para evaluar el riesgo antes de la otorgación de un crédito.

2.2.3 Plataforma web.

Para implementar la propuesta es necesario considerar diversas plataformas y herramientas de desarrollo, tanto para la estructuración de la comunicación en las aplicaciones, como para la codificación de las funcionalidades del sistema. Al emplear una arquitectura de tres capas, es necesario diseñar dos aplicaciones, una que trabaje como cliente (capa de presentación) y visualice las interfaces de usuario para mostrar la información de la base de datos; y otra aplicación que trabajará como servidor (capa de procesos y datos), la cual efectuará las llamadas y será un intermediario para la comunicación de los datos de la base de datos hacia las interfaces de usuario.

2.2.3.1 Modelo tres capas.

Una arquitectura tres capas es de gran ayuda al momento de implementar una aplicación web y que requiere protocolos de comunicación y un almacenamiento de archivos para la data, de manera que la información se encuentra accesible para las diferentes peticiones [24]. Esta arquitectura está organizada de manera jerárquica de manera que las capas posteriores generen servicios a las capas superiores [25]. La arquitectura 3 capas está compuesta por la capa de presentación, de procesos y de datos.

➤ Capa de presentación.

En esta capa se almacena la interfaz gráfica del usuario empleada para la comunicación con las otras capas de la arquitectura [26]. Las GUI deben emplear peticiones HTTP a la aplicación que se encuentra en la capa de procesos o también conocida como API-REST [27].

➤ Capa de procesos.

También denominada capa de contenido, y se centra la programación que establece comunicación con la capa de presentación y la capa de datos [10], es decir, procesa las peticiones del cliente que provienen de la capa de presentación, ejecuta el proceso en la capa de datos, y envía la respuesta de vuelta a la capa de presentación.

➤ **Capa de datos.**

En esta capa se almacena la data dentro de gestores de base de datos relacionales o no relaciones de manera que se puedan acceder a ellos mediante los diferentes clientes [28].

2.2.3.2 JavaScript.

JavaScript es un lenguaje de programación basado en script creado principalmente para la elaboración de páginas dinámicas y el más empleado para el desarrollo de aplicaciones web [29]. Una de las ventajas que posee este lenguaje es la adaptabilidad en código HTML [30].

➤ **Framework Angular.js**

Angular.js es uno de los framework para el desarrollo de software basado en javascript. Combinar el framework de Angular con Bootstrap ayuda a la reducción de tiempo de diseño de la interfaz gráfica de usuario, mejorando de esta forma la UI [31]. Uno de los principales objetivos de Angular.js es la facilidad de creación de contenido en una sola página, generando un desarrollo flexible [32].

➤ **Node.js**

Node.js ayuda en el manejo de eventos y facilita la comunicación con el servidor [33], diseñado para un entorno asíncrono y basado en eventos de manera que se empleen de manera eficaz los recursos que brinda el servidor. Node.js se emplea principalmente en el lado del servidor generando una aplicación escalable y con la posibilidad de conexiones simultáneas [34].

➤ **Express.js**

Express.js es un marco empleado en el lado del servidor para su configuración en el procesamiento de solicitudes HTTP mediante el API-REST [35]. Entre los beneficios que ofrece están la organización de rutas, la gestión de sesiones y de manera principal manipulación de consultas HTTP.

2.2.3.3 Base de datos no relacional.

Las bases de datos NoSQL han sido implementadas a mayor escala debido a la necesidad de procesamientos rápidos de grandes cantidades de datos que no se encuentran estructurados [36]. Una de las diferencias notables entre el uso de bases de datos relacionales y las no relacionales es la forma de organización de la información, ya que no se encuentran en tablas sino almacenadas en forma de documentos, que almacena su propio esquema, y propios datos.

➤ Mongo DB.

Mongo DB es un gestor de base de datos no relacional de tipo documento, en donde la información se almacena en formato JSON, con claves y valores propios para cada documento [37]. Los documentos son almacenados en colecciones, los cuales no necesariamente deben coincidir en cuanto a su esquema.

2.3 OBJETIVOS DEL PROTOTIPO.

2.3.1 Objetivo General

Implementar un sistema administrativo modular web para la gestión de cartera de la empresa IBERCONSA mediante la metodología SWIRL.

2.3.2 Objetivos Específicos

- Investigar las etapas de la metodología SWIRL para su aplicación dentro del desarrollo del proyecto.
- Analizar el proceso de gestión de cartera de la empresa IBERCONSA para la definición de los requerimientos funcionales y no funcionales.
- Diseñar los prototipos de las interfaces de usuario mediante el uso de herramientas CASE.
- Desarrollar el sistema web mediante el uso del Angular.js, node.js y MongoDB.

2.4 DISEÑO DEL PROTOTIPO.

Para diseñar el prototipo se consideran las actividades dentro de las fases de planificación y modelado, analizando los tiempos mediante el cronograma, los requerimientos, mediante las historias de usuario, y definiendo un modelo base de la interfaz gráfica de usuario, y la estructura navegacional y el diseño de la base de datos a desarrollar.

2.4.1 Fase de planificación

Dentro de la fase de planificación se establecen las historias de usuario, y las actividades a efectuar dentro del cronograma.

2.4.1.1 Historias de usuario

Mediante la entrevista con los interesados para la elicitación de requerimientos, y después del análisis de los procesos de negocio efectuados para el control de cartera, se definen las historias de usuario con una breve descripción del funcionamiento del sistema bajo dicho requerimiento. Cada historia de usuario se encuentra vinculada con un requerimiento funcional, por cada requerimiento puede existir una o más historias de usuario.

Tabla 12: Historia de usuario #1.

Historia de usuario	
Número: 1	Usuario: Todos
Nombre de historia: Login o inicio de sesión	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-01	
Programador responsable: Bárbara Romero	
Descripción: El usuario debe ingresar al sistema mediante autenticación, validando su usuario (número de cédula), y contraseña, y así visualizar la página principal con las funcionalidades según el rol y permisos que posea.	
Observaciones: Las funcionalidades restringidas de la página principal no deben ser visibles.	

Fuente: Elaboración propia.

- El sistema debe permitir el ingreso solo al personal autorizado, identificando su rol dentro del sistema y habilitando solo las funcionalidades permitidas para dicho usuario.

Tabla 13: Historia de usuario #2.

Historia de usuario	
Número: 2	Usuario: administrador
Nombre de historia: Gestión de usuarios (crear, modificar, visualizar, bloquear)	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-02	
Programador responsable: Bárbara Romero	
Descripción: Como administrador debe poder crear nuevos usuarios, modificar los datos de usuarios ya existentes, dar de baja a algún usuario y habilitarlo posteriormente, ya que no es posible eliminar un usuario.	
Observaciones: Los usuarios no pueden ser eliminados, sólo se modifica su estado de activo a bloqueado. Se pueden modificar todos los datos de los demás usuarios a excepción de la contraseña.	

Fuente: Elaboración propia.

Tabla 14: Historia de usuario #3.

Historia de usuario	
Número: 3	Usuario: administrador
Nombre de historia: Asignar roles y permisos al usuario	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-02	
Programador responsable: Bárbara Romero	
Descripción: Como administrador al crear o modificar un usuario debe poder asignar roles y permisos.	
Observaciones: Los permisos son visualizados por defecto según el rol que se seleccione, sin embargo, es posible otorgar nuevos permisos o quitar algunos ya establecidos por defecto.	

Fuente: Elaboración propia.

- Es importante que dentro de la gestión de usuarios se incluya la asignación de permisos ya sea al modificar o al agregar un nuevo usuario, sin embargo, esta actividad solo es desarrollada por el administrador.

Tabla 15: Historia de usuario #4.

Historia de usuario	
Número: 4	Usuario: Todos
Nombre de historia: gestionar clientes (ingresar, modificar, eliminar y visualizar)	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-03	
Programador responsable: Bárbara Romero	
Descripción: El usuario final o el de tipo administrador debe ser capaz de ingresar, modificar, y visualizar un cliente, siempre y cuando posea los permisos necesarios. Solo como administrador es posible eliminar un cliente.	
Observaciones: Al eliminar un cliente se eliminan, los contratos y pagos relacionados al mismo.	

Fuente: Elaboración propia.

- La gestión de clientes, (ingresar, editar, visualizar), debe ser realizada por todos los tipos de usuario, siempre y cuando posean los permisos adecuados. Mientras que la opción de eliminar solo será posible para el tipo administrador.

Tabla 16: Historia de usuario #5.

Historia de usuario	
Número: 5	Usuario: administrador
Nombre de historia: gestionar roles y permisos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-04	
Programador responsable: Bárbara Romero	
Descripción: Como administrador debe ser posible crear, editar, eliminar nuevos roles, y asignar los permisos por defecto de cada uno.	
Observaciones: El sistema contará con un administrador por defecto el cual no podrá ser eliminado, modificado, ni visualizado. Es posible crear nuevos usuarios y asignarles el rol de administrador.	

Fuente: Elaboración propia.

- El sistema debe poseer un formulario para el administrador, mediante el cual se puedan gestionar los roles, así como asignar los permisos por defecto de cada rol.

Tabla 17: Historia de usuario #6.

Historia de usuario	
Número: 6	Usuario: Todos
Nombre de historia: gestión de crédito	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-05	
Programador responsable: Bárbara Romero	
Descripción: El sistema debe permitir la apertura de nuevos créditos, asignando el monto para entrada, el monto para crédito y los plazos para cada uno. Una vez creado se vincula a un nuevo contrato.	
Observaciones: No hay observaciones	

Fuente: Elaboración propia.

Tabla 18: Historia de usuario #7.

Historia de usuario	
Número: 7	Usuario: Todos
Nombre de historia: gestión de crédito	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-06	
Programador responsable: Bárbara Romero	
Descripción: El sistema debe permitir la vinculación de un contrato a un crédito inmediatamente después de su apertura, solicitando la información relevante para la generación del mismo.	
Observaciones: Un contrato puede ser generado solo después de la apertura de un nuevo crédito, y requiere información de código de lotes, detalles del terreno, y beneficiario(s).	

Fuente: Elaboración propia.

- Dentro de la gestión de créditos, es importante la vinculación con contratos, ya que mediante este es posible validar el préstamo otorgado al cliente.

Tabla 19: Historia de usuario #8.

Historia de usuario	
Número: 8	Usuario: Usuario-final
Nombre de historia: gestión de pagos de cuotas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-07	
Programador responsable: Bárbara Romero	
Descripción: El usuario final debe ingresar un nuevo pago de la cuota del adeudo de un cliente. En caso de abonos se debe indicar el periodo al cual se va a abonar. En caso de cancelación completa el adeudo se requiere el análisis de no tener cuotas vencidas.	
Observaciones: solo se puede realizar un pago de cuota a la vez, las cuotas se deben cancelar de forma secuencial (no se elige el mes). Una vez ingresado el pago se procede a la impresión del comprobante de ingreso.	

Fuente: Elaboración propia.

- El usuario final que posea los permisos adecuados debe ser capaz de ingresar los pagos de las cuotas, según sea el orden en la tabla de amortización. Es posible realizar el pago de más de una cuota de forma secuencial, sin embargo, deben primero ser canceladas las cuotas vencidas. Bajo el último escenario, se considera el ingreso de un porcentaje de mora.

Tabla 20: Historia de usuario #9.

Historia de usuario	
Número: 9	Usuario: administrador
Nombre de historia: Visualizar panel administrativo	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-08	
Programador responsable: Bárbara Romero	
Descripción: El sistema debe mostrar un panel con los montos recolectados en el mes, así como una lista con los clientes en mora, y un listado con las cuotas que se deben cancelar en el mes.	
Observaciones: Se debe especificar el monto de ingreso de caja y de banco.	

Fuente: Elaboración propia.

- El panel administrativo o dashboard es implementado para conocer el estado actual de la empresa en cuanto a las deudas de crédito, además de identificar un listado con los clientes en mora y las cuotas por cancelar en el mes para el área de cobranza. A esta funcionalidad pueden acceder los demás usuarios siempre y cuando se les otorgue dichos permisos.

Tabla 21: Historia de usuario #10.

Historia de usuario	
Número: 10	Usuario: administrador
Nombre de historia: gestión de valores de cartera	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-09	
Programador responsable: Bárbara Romero	
Descripción: El sistema debe permitir la gestión (ingresar modificar, y eliminar) de los valores necesarios para la cartera como, entidades bancarias, formas de pagos, y tipos de créditos, entre otros.	
Observaciones: No hay observaciones.	

Fuente: Elaboración propia.

Tabla 22: Historia de usuario #11.

Historia de usuario	
Número: 11	Usuario: administrador
Nombre de historia: gestión de valores del sistema	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-09	
Programador responsable: Bárbara Romero	
Descripción: El sistema debe permitir la gestión (ingresar modificar, y eliminar) de los valores de campos necesarios para algunos formularios, como el estado civil, estado laboral, y cargo los cuales son propuestos por la empresa.	
Observaciones: No hay observaciones.	

Fuente: Elaboración propia.

Tabla 23: Historia de usuario #12.

Historia de usuario	
Número: 12	Usuario: administrador
Nombre de historia: Reportes de gestión de cartera	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Requerimiento funcional: RF-09	
Programador responsable: Bárbara Romero	
Descripción: El administrador y los usuarios con permisos deben ser capaces de visualizar los reportes de carteras vencidas general y por cliente, así como visualizar el histórico de pagos de cada cliente.	
Observaciones: Dentro de la visualización de cartera vencida por cliente se realiza el control de las notificaciones, ingresando la fecha y motivo del momento en que se notificó por cuota vencida.	

Fuente: Elaboración propia.

2.4.1.2 Gestión del cronograma

En base a las fases consideradas de la metodología SWIRL para el desarrollo de la propuesta, en la *Figura 4* se define el cronograma detallando las actividades a ejecutar dentro del proyecto para cada una de las fases, indicando su duración, la fecha de comienzo y la fecha de finalización. En la *Figura 5* se ilustra el respectivo diagrama de Gantt.

Nombre de tarea	Duración	Comienzo	Fin
◀ SISTEMA WEB DE CONTROL DE CARTERA	67 días	jue 10/6/21	vie 10/9/21
◀ Fase I: Análisis	8 días	jue 10/6/21	lun 21/6/21
Definición del sistema	2 días	jue 10/6/21	vie 11/6/21
Estudio de factibilidad	1 día	lun 14/6/21	lun 14/6/21
Identificación de riesgos	1 día	mar 15/6/21	mar 15/6/21
Elicitación de requerimientos	4 días	mié 16/6/21	lun 21/6/21
◀ Fase II: Planificación	4 días	mar 22/6/21	vie 25/6/21
Historias de usuario	2 días	mar 22/6/21	mié 23/6/21
Gestión de cronograma	1 día	jue 24/6/21	jue 24/6/21
EDT	1 día	vie 25/6/21	vie 25/6/21
◀ Fase III: Modelado	16 días	lun 28/6/21	lun 19/7/21
Diseño de la base de datos	3 días	lun 28/6/21	mié 30/6/21
Diseño de interfaz abstracta de usuario	1 día	jue 1/7/21	jue 1/7/21
Diseño navegacional	2 días	vie 2/7/21	lun 5/7/21
Diseño de diagramas UML	2 días	mar 6/7/21	mié 7/7/21
Redacción del informe final (Cap 1 y 2)	8 días	jue 8/7/21	lun 19/7/21
◀ Fase IV: Implementación	33 días	jue 8/7/21	lun 23/8/21
Diseño de interfaces de usuario	5 días	jue 8/7/21	mié 14/7/21
Codificación de las funcionalidades y módulos	25 días	jue 15/7/21	mié 18/8/21
Validación de restricciones	3 días	jue 19/8/21	lun 23/8/21
◀ Fase V: Revisión y pruebas	4 días	mar 24/8/21	vie 27/8/21
Plan de evaluación de calidad	2 días	mar 24/8/21	mié 25/8/21
Análisis de resultados de pruebas	2 días	jue 26/8/21	vie 27/8/21
◀ Fase VI: Lanzamiento	6 días	lun 30/8/21	lun 6/9/21
Elección de dominio	1 día	lun 30/8/21	lun 30/8/21
Preparación de host	2 días	mar 31/8/21	mié 1/9/21
Implementación del sistema en el host	3 días	jue 2/9/21	lun 6/9/21
Redacción del informe final (Cap. 3)	4 días	mar 24/8/21	vie 27/8/21
Revisión del informe de titulación	10 días	lun 30/8/21	vie 10/9/21

Figura 4 Cronograma de actividades

Fuente: Elaboración propia

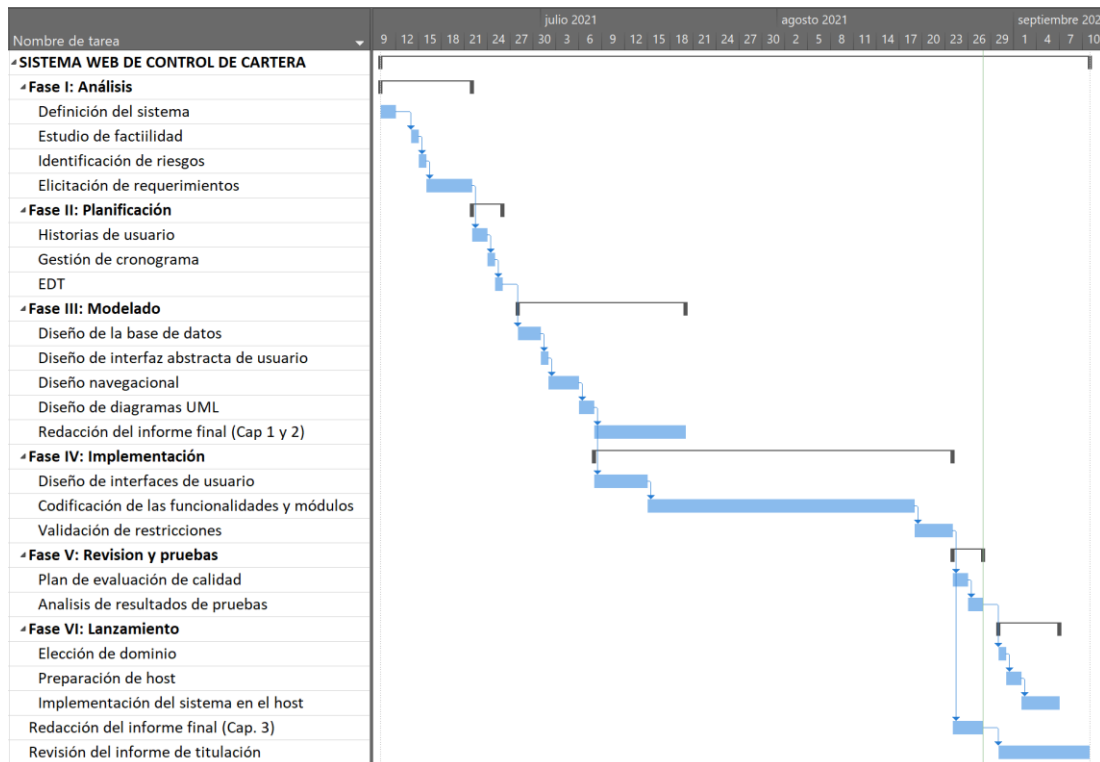


Figura 5 Diagrama de Gantt

Fuente: Elaboración propia

2.4.1.3 EDT (Estructura de desglose de trabajo)

Para que el desarrollo del sistema no sea complejo, se descomponen sus funcionalidades en tareas más sencillas, según los módulos a los cuales pertenece, presentado en el **ANEXO C**. Los módulos distribuidos para el sistema son los siguientes:

- Usuarios
- Clientes
- Cartera
- Roles y permisos
- Entidades Independientes

2.4.2 Fase de modelado

Dentro de esta fase se establecen los modelos bases para el desarrollo del sistema y la estructuración adecuada de las funciones, como el diseño de la base de datos, el modelo navegacional, las interfaces de usuario, y los diagramas UML pertinentes.

2.4.2.1 Diseño de la base de datos

Al emplear un gestor de base de datos NoSql como MongoDB, la estructura del modelado de la misma se realiza mediante los esquemas empleados para almacenar cada registro. La Figura 6 detalla los esquemas a emplear y las conexiones entre los mismos en caso de ser necesarias. Las líneas entre las conexiones de los esquemas se interpretan de la siguiente manera:

- Las líneas con doble flecha indican que los dos documentos poseen una referencia de ambos, pudiendo ser un arreglo o un solo campo.
- Las líneas con una sola flecha indican que el documento posee una referencia del otro esquema mediante el id propio.
- Las líneas entre-cortadas indican que ambos esquemas poseen relación, sin embargo, no se establece como referencia dentro del modelo sino como un campo estático.

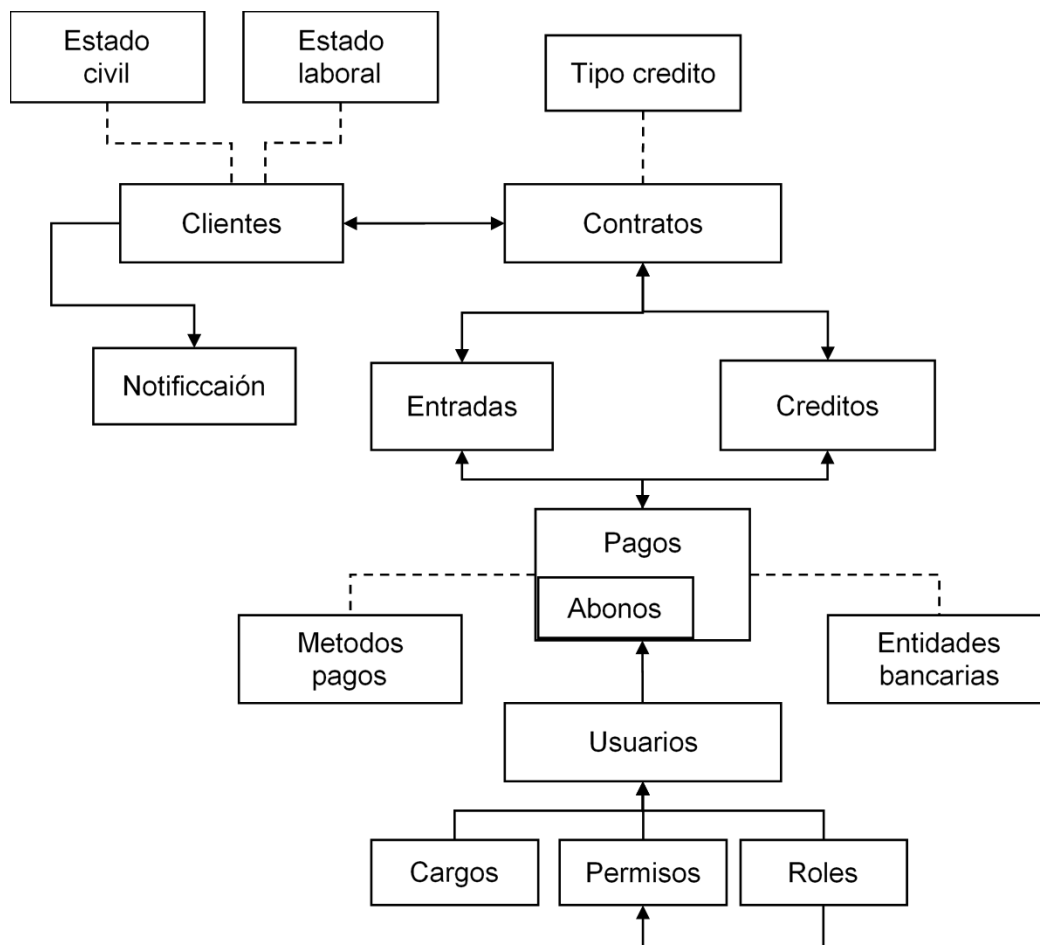


Figura 6 Modelo relacional de la base de datos
Fuente: Elaboración propia

Los modelos de cada esquema son detallados en el **ANEXO D**, en el cual se especifican el nombre de los campos, el tipo de datos, las restricciones (en caso de existir). La lógica inmersa dentro del modelo relacional de la base de datos se centra en el esquema *Contratos* los cuales dentro de cada documento ingresa como referencia un campo para el crédito y otro para la entrada, ya que cada crédito otorgado cuenta con una entrada, se crean otros esquemas para el almacenamiento de la data de *Créditos* y *Entradas*. Para los pagos, se crea un esquema separado, el cual se relaciona al crédito o entrada, según un código y concepto, el esquema de pagos tiene a su vez una lista de subdocumentos que hacen referencia a los abonos realizados dentro del mismo, cada pago hace referencia a una cuota (periodo); dentro de los pagos se incluyen los nombres de las *Entidades bancarias* y los *Métodos de pagos*, y una referencia al id del usuario que ingresó el pago al sistema.

Se crea también un esquema para los *Clientes*, los cuales se relacionan bidireccionalmente con el contrato, es decir, un cliente posee como campo una lista

con los id de los contratos, mientras que en cada documento de contrato se incluye un campo de referencia con el id del cliente. Las notificaciones se realizan de forma general, es decir, no por contrato sino por cliente, por lo que cada notificación posee el id del cliente.

Por otro lado, el esquema *Usuarios* almacena la información básica de cada usuario, un campo de referencia del cargo, una lista con los id de los permisos y otra lista con los id de los roles asignados. Finalmente se crean tablas independientes para el *estado civil*, *estado laboral*, *entidades bancarias*, y *métodos de pago*, de manera que la empresa pueda gestionar sus contenidos para ser visualizados en las interfaces de usuarios.

2.4.2.2 Diseño del modelo navegacional

Para identificar el funcionamiento del sistema en relación a los re-direccionamientos a otras páginas, se presentan los siguientes modelos navegacionales en relación a los diferentes usuarios del sistema, ya que cada usuario posee permisos a diversas funcionalidades. El modelo navegacional del administrador detallado en la *Figura 7* muestra el funcionamiento completo del sistema, ya que este tipo de usuario posee acceso a todas las páginas y funcionalidades del mismo.

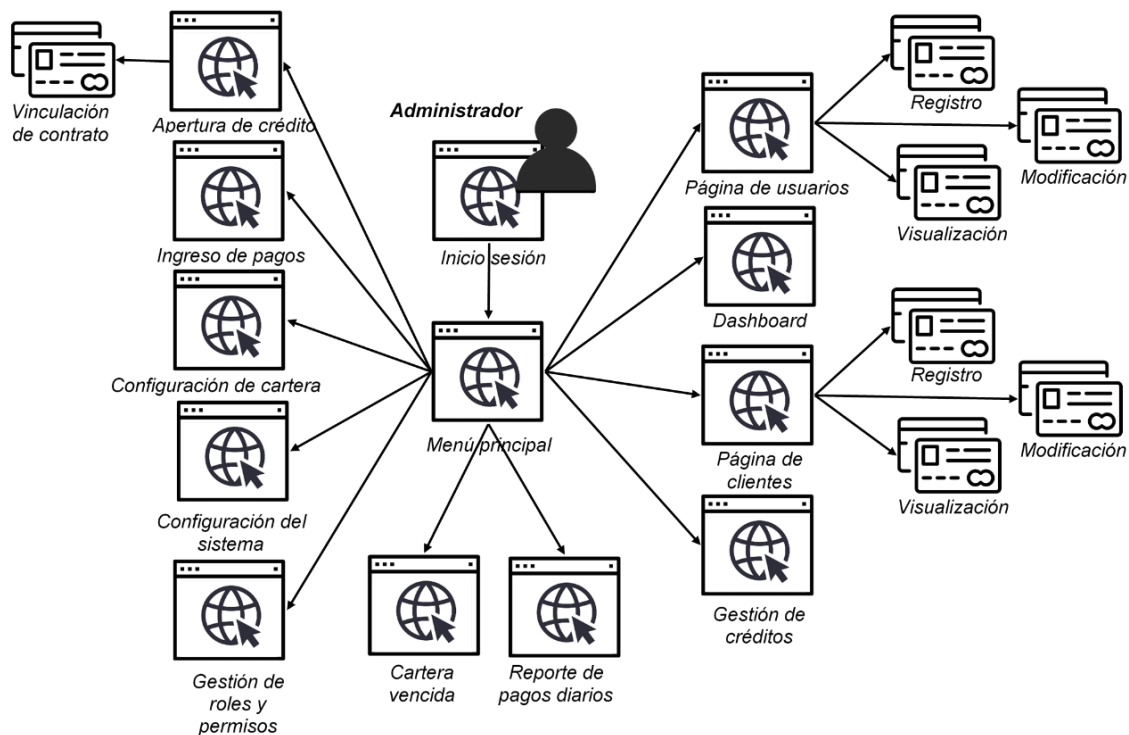


Figura 7 Modelo navegacional: usuario administrador

Fuente: Elaboración propia

El usuario final, el cual es considerado como un empleado de la empresa, posee permisos cambiantes, esto se debe a la propiedad del sistema de ser administrable. Las páginas y funcionalidades a las cuales tienen acceso son designadas según los permisos que posee cada uno por lo que el modelo navegacional para este tipo de usuarios puede variar. No obstante, se define un modelo navegacional presentado en la *Figura 8* con todas las páginas y funciones a las cuales puede acceder un usuario final.

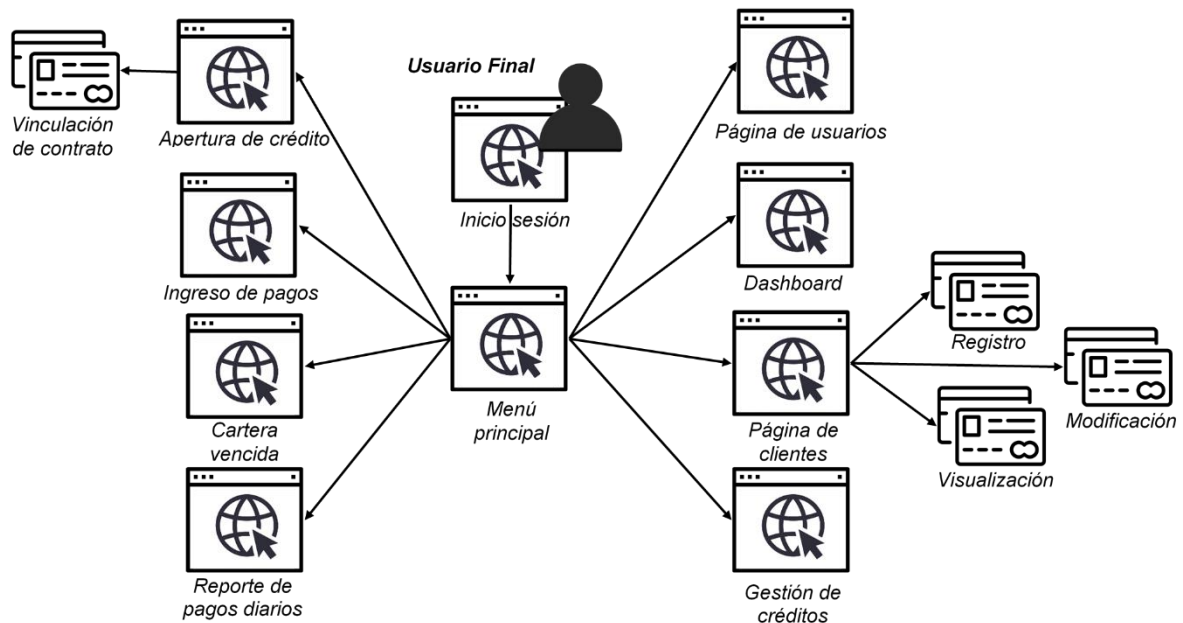


Figura 8 Modelo navegacional: usuario final
Fuente: Elaboración propia

2.4.2.3 Diseño de interfaz abstracta de usuario

Para el diseño de los prototipos de la interfaz abstracta de usuario se emplea la herramienta Balsamiq Mockups, además de contemplar una tendencia de diseño responsivo, permitiendo que sea adaptable a la pantalla de cualquier dispositivo.

➤ Tendencia de diseño

La tendencia de diseño empleada en cada proyecto de software está estrechamente vinculado a las características que este posee y al entorno bajo el cual se va a desenvolver, por lo que su selección es propia por cada proyecto [38].

Sin embargo, la metodología SWIRL propone la inclusión de normas de tendencias de diseño en el proceso de creación y codificación de las interfaces

de usuario. Una de las necesidades que presenta el sistema es la adaptabilidad a diferentes dispositivos los cuales contengan conexión a internet, como un Smartphone, Tablet u ordenador. Es por ello, que se ha considerado a Responsive Design (Diseño adaptable o responsivo) como la tendencia de diseño para la presente propuesta.

➤ **Prototipado de interfaces de usuario**

Mediante el diseño de los prototipos de las interfaces de usuario, es posible obtener una guía para la creación de la misma durante la etapa de codificación, así como la identificación de posibles fallas de funcionalidad o lógica de diseño dentro de las interfaces. A continuación, se describen los prototipos de interfaces según los módulos a los cuales pertenecen.

La página inicial es el login o inicio de sesión la cual se accede mediante la validación de la cédula y contraseña se muestra en la *Figura 9*.



Figura 9 Prototipo de interfaz: Inicio de sesión
Fuente: Elaboración propia

Una vez validados los datos, se presenta el menú principal como se muestra en la *Figura 10*, el cual contiene un menú lateral con las opciones y funcionalidades a las cuales el usuario tiene acceso según los permisos que se le han otorgado.

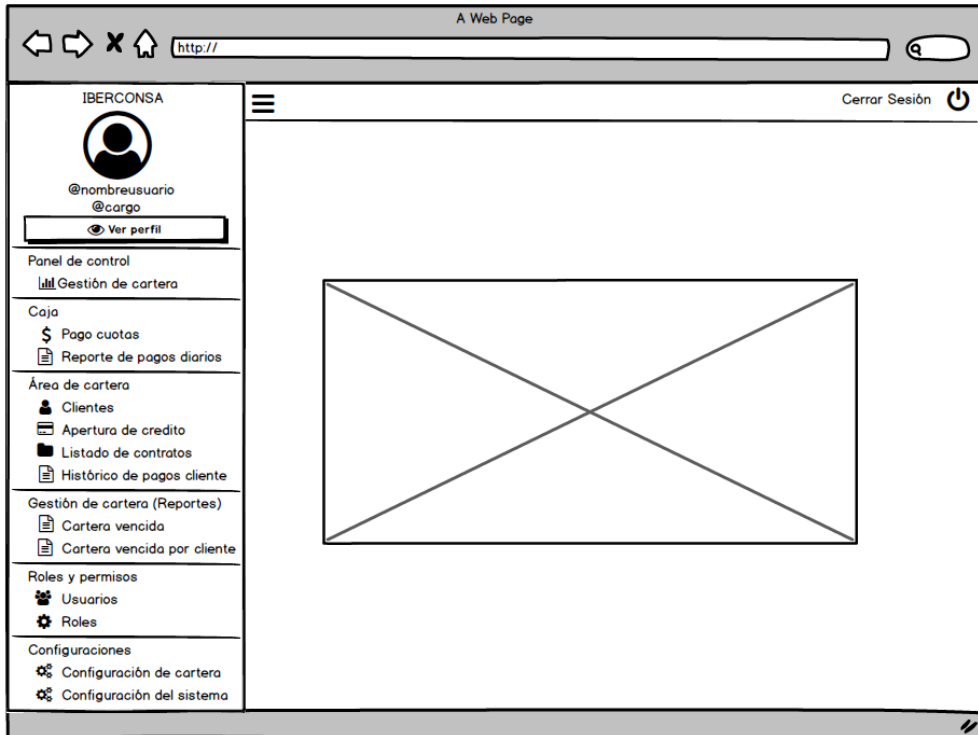


Figura 10 Prototipo de interfaz: menú principal
Fuente: Elaboración propia

Todos los tipos de usuario, sin excepción alguna, pueden visualizar su perfil como se detalla en la *Figura 11*, el cual contiene solo la información personal del usuario que ha iniciado sesión actualmente.

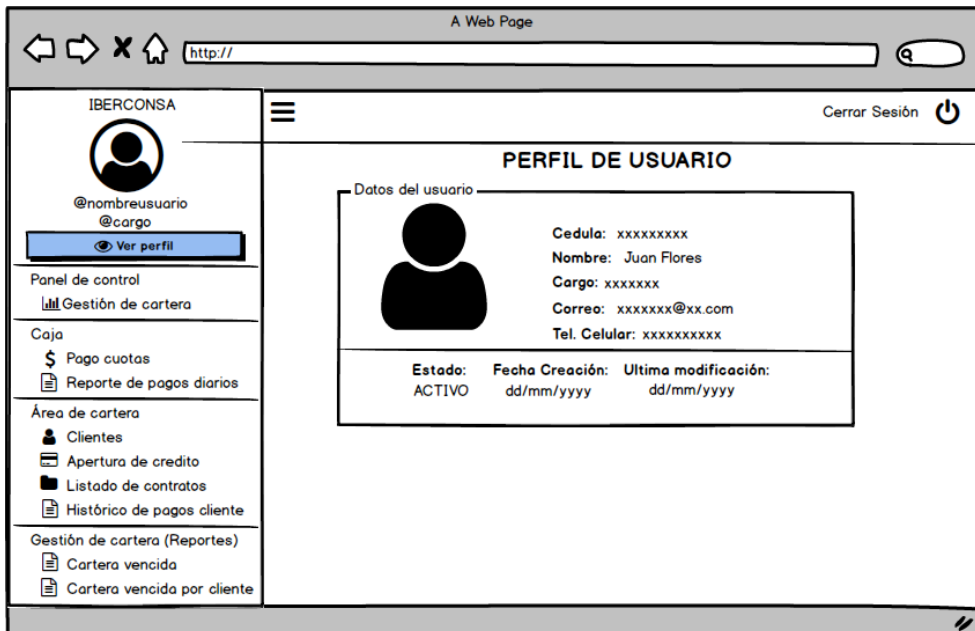


Figura 11 Prototipo de interfaz: perfil de usuario
Fuente: Elaboración propia

Gestión de usuarios

La gestión de usuarios inicia con el listado de todos los usuarios del sistema, indicando en la tabla las acciones para la eliminación, visualización y modificación de los mismos como se muestra en la *Figura 12*. Es importante mencionar que un usuario no puede ser eliminado, en su lugar su estado pasa a estar bloqueado.

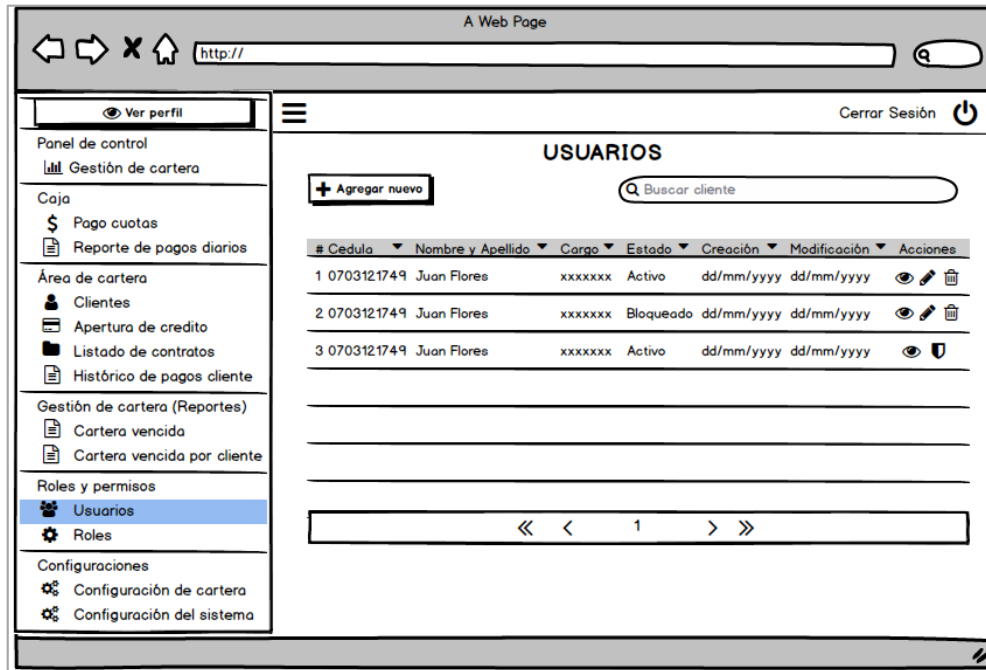


Figura 12 Prototipo de interfaz: listado de usuario

Fuente: Elaboración propia

La *Figura 13* muestra los formularios flotantes para el ingreso de un nuevo usuario, así como la modificación del mismo en caso de ser solicitado.

Al ingresar un nuevo usuario se debe seleccionar el rol al que pertenece, pueden ser más de dos en caso de empresas pequeñas, además de marcar y desmarcar los permisos que posee dicho usuario dentro del sistema. Al seleccionar un rol, se marcarán los permisos del usuario que se encuentren por defecto, sin embargo, el administrador puede asignar o eliminar los permisos para ese usuario.

Figura 13 Prototipo de interfaz: ingreso y edición de usuario
Fuente: Elaboración propia

La Figura 15 Muestra la página empleada en caso que se desee ver la información de cada usuario, en ella se detallan los datos básicos, una lista de roles a los cuales pertenece, y una lista con los permisos que se le han otorgado.

Figura 14 Prototipo de interfaz: ingreso y edición de usuario
Fuente: Elaboración propia

Gestión de clientes

En la *Figura 15* se detalla la página para la gestión de clientes, la cual cuenta con una tabla listando todos los clientes que posee la empresa, incluyendo información relevante para conocer su estado crediticio actual.

La tabla cuenta con acciones similares a la de los usuarios, visualizar, editar, y eliminar, la última opción solo será para el usuario de tipo administrador.

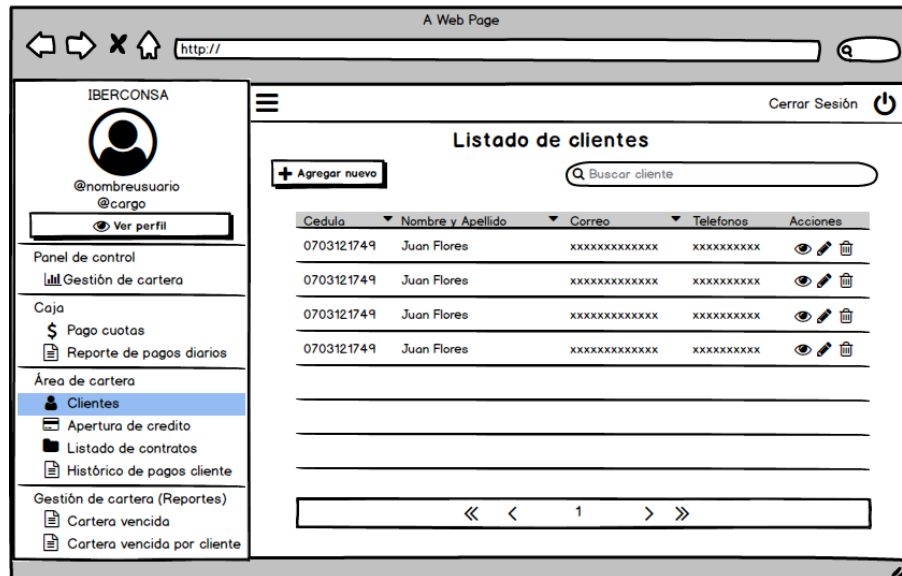


Figura 15 Prototipo de interfaz: listado de clientes

Fuente: Elaboración propia

Al crear un nuevo cliente o editar la información de alguno existente, se visualiza el formulario presentado en la *Figura 16*, en la cual se deben completar la información básica del cliente, los datos de su cónyuge (en caso de poseer), y los datos de la empresa donde labora (en caso de trabajar).

Al momento de visualizar la información del cliente, se presenta una nueva ventana con la información básica del cliente, un breve detalle de los saldos del cliente, y un listado de los créditos que se les ha otorgado a lo largo del tiempo como se muestra en la *Figura 17*.

Ingresar nuevo cliente (Editar información de cliente)

Datos del cliente

Cédula: Nombre: Apellido:

Dirección:

Ciudad: Estado civil: Correo:

Estado laboral: Telefono 1: Telefono 2:

Datos del cónyuge

Cédula:

Nombre: Apellido:

Correo: Tel. Celular: Estado laboral:

Datos del trabajo/ negocio

Nombre de empresa o negocio: Dirección:

Cargo: Tipo de empresa: Telefono:

Descripción del negocio:

Figura 16 Prototipo de interfaz: ingreso de nuevo cliente
Fuente: Elaboración propia

A Web Page
http://

☰
Cerrar Sesión

◀ Listado de clientes
Detalle del cliente

Datos del cliente

Cédula: xxxxxxxxxx

Dirección: xxxxxxxxxxxxxxxxxxxxxxxx

Ciudad: Machala

Estado Civil: Xxxxxxx

Nombre: Juan Juan Flores Flores

Correo: correo@example.com

Telefonos: xxxxxxxxxx - xxxxxxxx

Estado Laboral: Xxxxxxx

Datos del trabajo/ negocio

Nombre: xxxxxxxxxx

Tipo de empresa: Xxxxxxxx

Cargo: Xxxxxxxx

Telefono: xxxxxxxxxx

Descripción del negocio:

Datos del cónyuge

Cédula: xxxxxxxxxx

Nombre: Juan Juan Flores Flores

Estado Laboral: Xxxxxxx

Telefono: xxxxxxxxxx

Correo: correo@example.com

Listado de Creditos

#	Código de Contrato	Fecha de apertura	Fecha de culminación	Plazo	Monto total crédito	Estado	Acciones
1	x	01/01/2010	01/01/2020	120 meses	\$0.00	Cancelado	<input type="button" value="🔍"/>
2	x	01/01/2020	01/01/2030	120 meses	\$0.00	Vigente	<input type="button" value="🔍"/>

Figura 17 Prototipo de interfaz: visualización de cliente
Fuente: Elaboración propia

Gestión de créditos

En la *Figura 18* se visualiza página principal para la gestión de créditos, esta página lista todos los créditos existentes indicando los detalles básicos como el cliente, el tipo de crédito y las acciones de visualizar, y eliminar (restringidas según los permisos que posee el usuario que ha iniciado sesión).

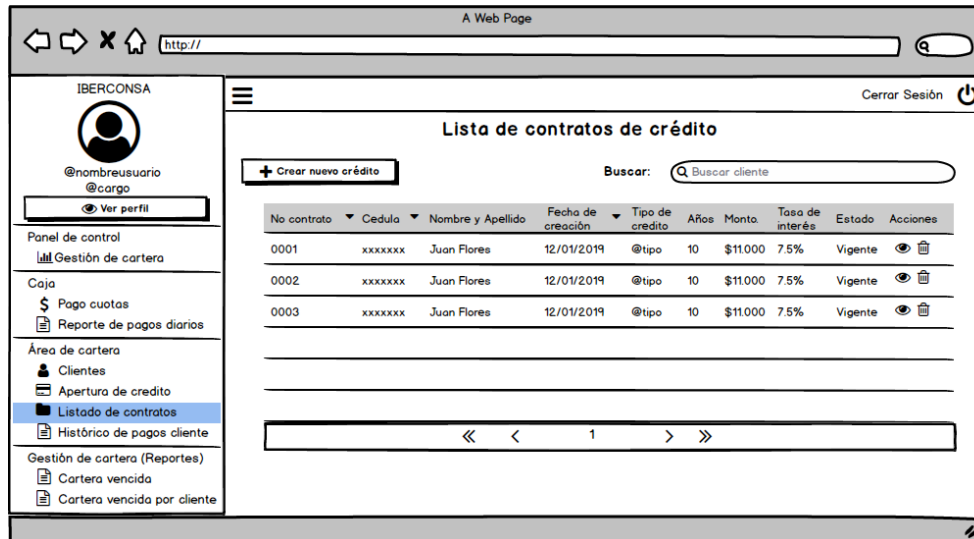


Figura 18 Prototipo de interfaz: listado de créditos

Fuente: Elaboración propia

En esta página es posible también la creación de un nuevo crédito, a manera de acceso directo, ya que esta funcionalidad también puede ser ejecutada en la opción del menú lateral “*Apertura de nuevo crédito*”.

El proceso de eliminación se emplea para el proceso de migración de datos, en los cuales pueden existir errores en el traspaso de la información.

Al visualizar los datos del crédito se presenta la información básica como se muestra en la *Figura 19*. Es posible visualizar la tabla de amortización actual, identificando los montos cancelados al igual que el monto de la deuda y cuotas pendientes.

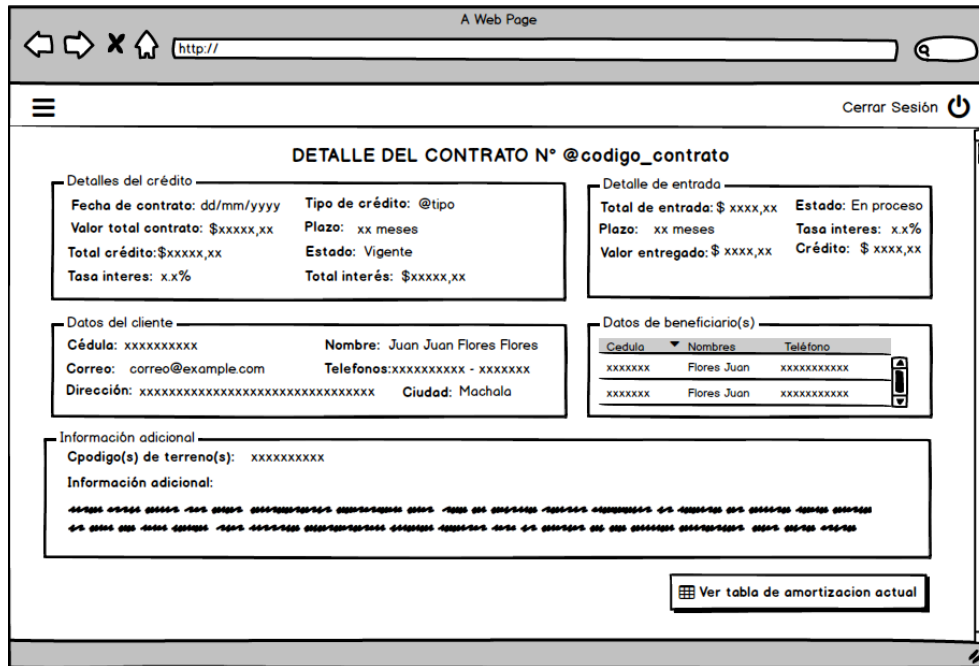


Figura 19 Prototipo de interfaz: visualización de detalle de crédito
Fuente: Elaboración propia

Para la apertura de un nuevo crédito, se puede acceder desde la gestión de créditos (lista de créditos), o desde la opción en el menú lateral. La *Figura 20* muestra la ventana presentada en la creación de un nuevo crédito, siendo requerido campos como el monto del crédito, el valor de la entrada y los datos del cliente, para posteriormente mostrar las tablas de amortizaciones tanto de la entrada como del crédito.

Dentro de este punto se crea un campo separado para entrada, debido a que, dentro del proceso del negocio de la empresa, para la apertura del crédito es necesario cancelar una entrada la cual puede ser en efectivo al corriente o diferido a un corto plazo (en este último caso es necesario crear una tabla de amortización).

Una vez guardado y verificado los datos del crédito, se procede a la creación del contrato. Sin embargo, existe la opción de imprimir la amortización, en caso que el cliente requiera simplemente de una cotización y no de un nuevo contrato.

Para el proceso de migración de datos se crea un botón para el ingreso de contratos existentes, este solo es visible mediante la asignación de los permisos requeridos.

IBERCONSA Cerrar Sesión

APERTURAR NUEVO CRÉDITO Código contrato: C-001

Datos del cliente

Cédula:

Nombre: Flores Flores Juan Juan

Dirección: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx Ciudad: Machala

Correo: correo@example.com

Detalle del crédito

Fecha: / / Tipo de crédito: - Seleccione -

Valor total: \$ Tasa de interés: %

Plazo: meses Monto interés: \$

Total Entrada: \$ Crédito: \$

Detalle de entrada

Entregado: \$

Credito: \$

Plazo: meses

Tasa de interés: %

Tabla de amortización de la Entrada

N° Periodos	Fecha Vencimiento	Valor dividendo	Interes	Capital pagado	Saldo del capital
0	--	-	-	-	\$xxxx.xx
1	dd/mm/yy	\$xxxx.xx	\$x.xx	\$xxxx.xx	\$xxxx.xx
2	dd/mm/yy	\$xxxx.xx	\$x.xx	\$xxxx.xx	\$xxxx.xx

Tabla de amortización del crédito

N° Periodos	Fecha Vencimiento	Valor dividendo	Interes	Capital pagado	Saldo del capital
0	--	-	-	-	\$xxxx.xx
1	dd/mm/yy	\$xxxx.xx	\$x.xx	\$xxxx.xx	\$xxxx.xx
2	dd/mm/yy	\$xxxx.xx	\$x.xx	\$xxxx.xx	\$xxxx.xx
3	dd/mm/yy	\$xxxx.xx	\$x.xx	\$xxxx.xx	\$xxxx.xx

Figura 20 Prototipo de interfaz: apertura de nuevo crédito
Fuente: Elaboración propia

Para crear un nuevo contrato es necesario llenar los campos del formulario que se presenta en la *Figura 21*. Se ingresan los datos de las personas beneficiarias en caso de fallecimiento y la información adicional como el número de terrenos y los códigos con los que la empresa los identifica.

IBERCONSA Cerrar Sesión

DETALLE PARA EL CONTRATO

Datos del cliente

Cédula: xxxxxxxxxx Nombre: Juan Juan Flores Flores

Dirección: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx Ciudad: Machala

Correo: correo@example.com

Información adicional

Cant. de terrenos: 2 terrenos

Código(s):

Escriba aquí la información y datos sobre el(los) terreno(s)

Beneficiario(s) en caso de fallecimiento

En esta sección se establecen el(los) beneficiario(s) en caso del fallecimiento del cliente bajo el cual se realiza el contrato inicial.

Cédula:

Nombre: Apellido:

Correo: Telefono:

Cedula	Nombres	Teléfono	Acciones
xxxxxxx	Flores Juan	xxxxxxxxxxxx	<input type="button" value="Q"/> <input type="button" value="X"/>
xxxxxxx	Flores Juan	xxxxxxxxxxxx	<input type="button" value="Q"/> <input type="button" value="X"/>

Figura 21 Prototipo de interfaz: vinculación con contrato
Fuente: Elaboración propia

Para la visualización del panel de cobranza se diseña una ventana que permita la visualización de los valores ingresados durante el mes gracias a los pagos de los adeudos, así como la lista de los clientes en mora, y las cuotas por cobrar el presente mes como se muestra en la *Figura 22*.

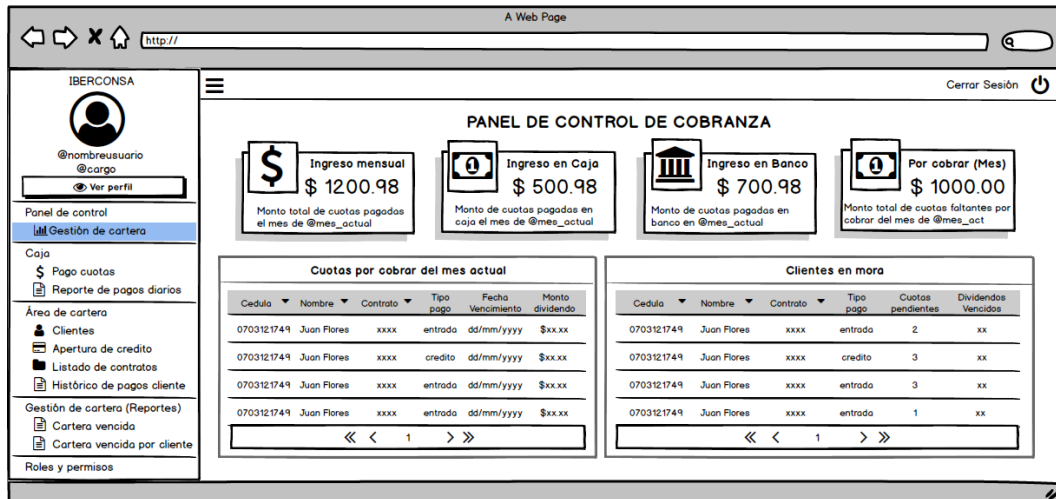


Figura 22 Prototipo de interfaz: panel de control

Fuente: Elaboración propia

Gestión de pagos.

La *Figura 23* muestra los campos necesarios para ingresar un nuevo pago de una cuota por parte del cliente. El sistema solicitará inicialmente el número de cédula, con lo cual se visualizará una lista de contratos vinculados, y la opción de ingresar el delimitante de la fecha hasta la cual se desean consultar las cuotas. El pago de las cuotas puede realizarse de dos formas, pagar o abonar, y solo se pueden realizar de manera secuencial, es decir, debe estar pagada la cuota anterior para proceder con el mes actual. El código del comprobante de ingreso se genera únicamente al seleccionar el periodo a cancelar.

Al pagar por completo una cuota se deben completar los campos detallados en la *Figura 23*. En la cual se muestra información relevante de manera solo lectura, y las casillas para el ingreso del monto cancelado, la forma de pago, y demás información, se encuentran habilitada. En caso de que dicha cuota posea abonos anteriores, se mostrará una lista detallada de los mismos. Una vez completada la información se selecciona la opción guardar e imprimir comprobante, con lo que se generará un comprobante de ingreso en formato pdf.

INGRESO DE PAGO
COMPROBANTE DE INGRESO

Comprobante de ingreso: N° de comprobante interno:

Ingresar existente

Datos del cliente:
Cédula:
Nombre: Flores Flores Juan Juan
Correo: correo@example.com

Ver cuotas hasta:
Código contrato: Seleccione

Lista de cuotas

Tipo	N°	Fecha Vencimiento	Valor a cancelar	Acciones
Entrada	16	dd/mm/yy	\$xxxx.xx	<input type="button" value="Pagar"/> <input type="button" value="Abonar"/>
Credito	1	dd/mm/yy	\$xxxx.xx	<input type="button" value="Pagar"/> <input type="button" value="Abonar"/>

« < 1 > »

Detalles del pago

Tipo de cuota: Entrada Período: ## Fecha vencimiento: #/##/##
 Valor capital: \$ Valor interes: \$ Valor dividido: \$
 Interes mora(%): % Interes mora(\$): \$
 Total a pagar: \$ Valores abonados: \$ Saldo capital: \$
 Fecha de emisión: Valor cancelado: \$ Método de pago: Seleccione
 Entidad bancaria: Seleccione No recibo Banco: Fecha depósito (Pago):

Detalles de abonos

#	Fecha	Valor	Forma de pago
1	dd/mm/yy	\$xx.xx	xxxxxx

Figura 23 Prototipo de interfaz: ingreso de nuevo pago
Fuente: Elaboración propia

Para realizar el abono de una cuota se deben completar los campos de la Figura 24 siendo menos la cantidad de campos visualizados a diferencia de un pago completo. Al igual que la apertura de crédito, los pagos tienen una opción para el proceso de migración de información, habilitando los campos autogenerados, como códigos y fechas.

INGRESO DE PAGO
COMPROBANTE DE INGRESO

Comprobante de ingreso: N° de comprobante interno:

Ingresar existente

Datos del cliente:
Cédula:
Nombre: Flores Flores Juan Juan
Correo: correo@example.com

Ver cuotas hasta:
Código contrato: Seleccione

Lista de cuotas

Tipo	N°	Fecha Vencimiento	Valor a cancelar	Acciones
Entrada	16	dd/mm/yy	\$xxxx.xx	<input type="button" value="Pagar"/> <input type="button" value="Abonar"/>
Credito	1	dd/mm/yy	\$xxxx.xx	<input type="button" value="Pagar"/> <input type="button" value="Abonar"/>

« < 1 > »

Detalles del Abono

Tipo de cuota: Entrada Período: ## Fecha vencimiento: #/##/##
 Valor dividido: \$ Valor interes: \$ Valor capital: \$
 Fecha de emisión: Valor del abono: \$ Método de pago: Seleccione
 Entidad bancaria: Seleccione No recibo Banco: Fecha depósito (Pago):

Figura 24 Prototipo de interfaz: ingreso de nuevo abono
Fuente: Elaboración propia

Los comprobantes de ingresos generados, tanto los pagos como los abonos, son visualizados en el reporte de pagos diarios como se detalla en la Figura 25, en el cual se visualiza una lista con todos los pagos realizados en un día en específico (por defecto el día actual).

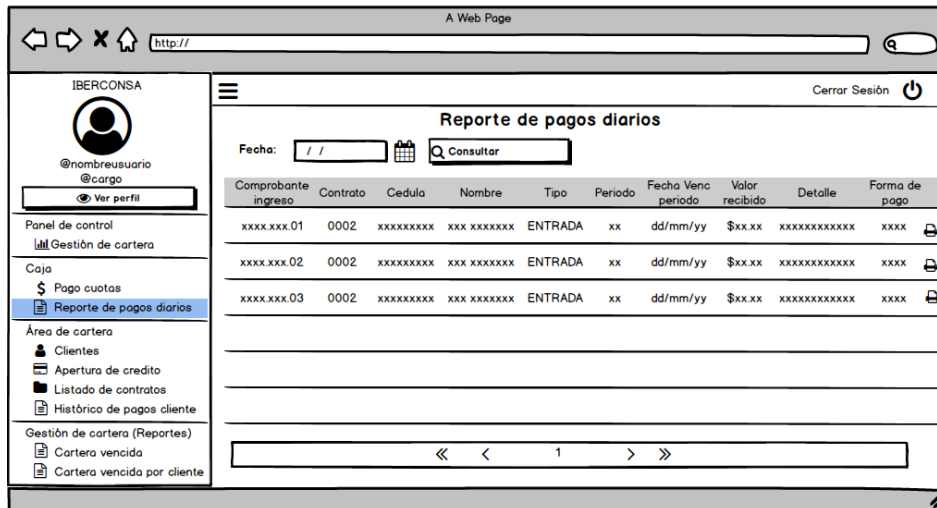


Figura 25 Prototipo de interfaz: Reporte de pagos diarios

Fuente: Elaboración propia

De igual forma es posible visualizar el histórico de pagos del cliente como se muestra en la Figura 26, en el cual se debe ingresar el número de cédula, luego se visualizarán los contratos vinculados y al seleccionar la opción de ver pagos en el contrato deseado, se generará una lista con todos los pagos que se han realizado a dicho contrato.

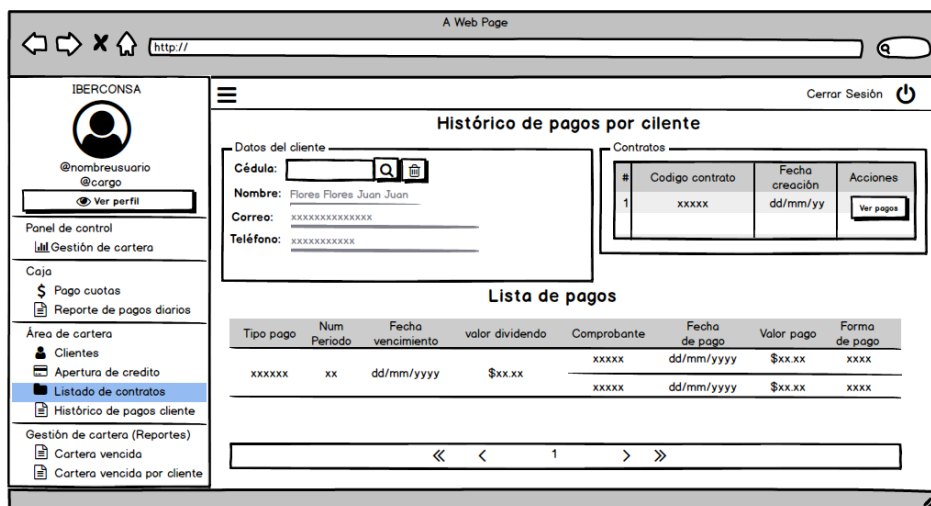


Figura 26 Prototipo de interfaz: Histórico de pagos por cliente

Fuente: Elaboración propia

Gestión de cartera.

Dentro de la gestión de cartera se lleva el control de la cartera vencida para ello se visualizan dos páginas, la primera muestra la cartera vencida general, indicando cuales son los clientes y cuotas que se mantienen en mora (Figura 27).

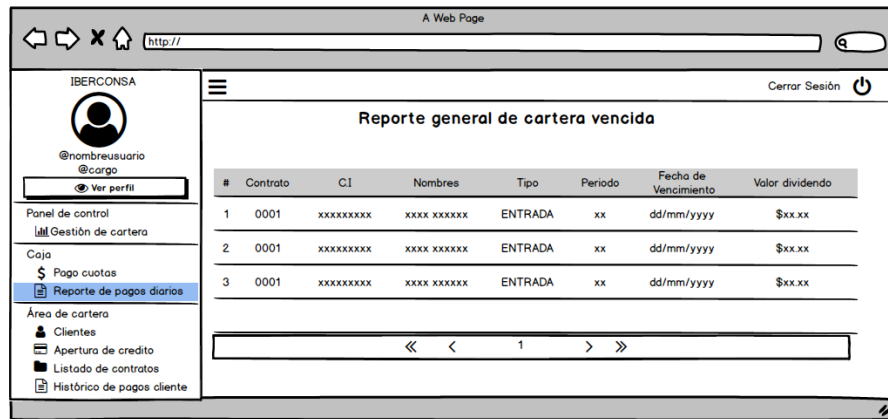


Figura 27 Prototipo de interfaz: Reporte de cartera general vencida
Fuente: Elaboración propia

La Figura 28 muestra la cartera vencida por cliente, en donde es necesario ingresar el número de cédula y automáticamente se visualizan las cuotas vencidas que posea el mismo. Dentro de esta página se gestionan también las notificaciones en caso de existir cuentas en mora.

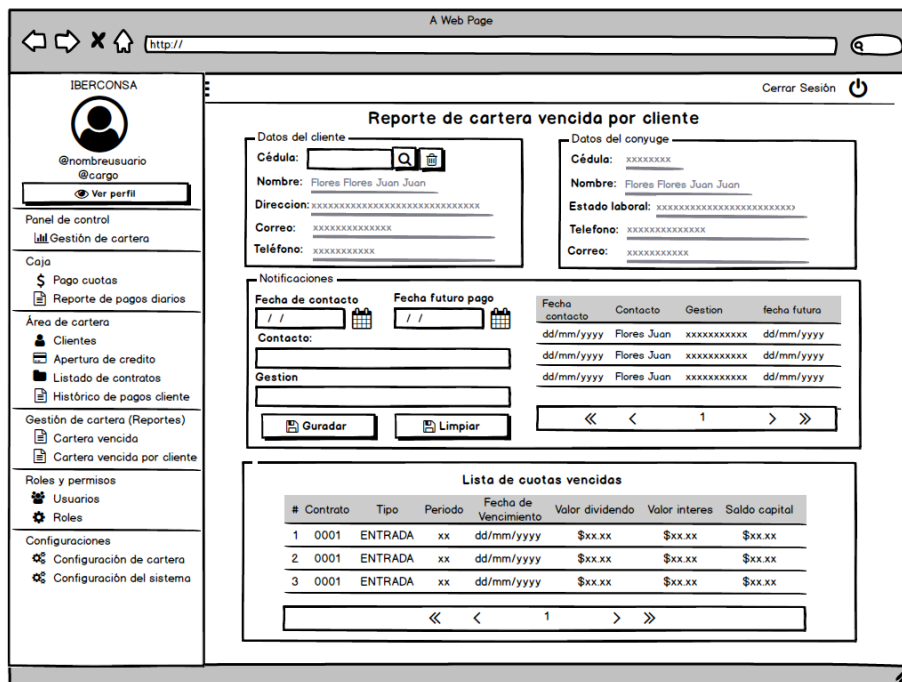


Figura 28 Prototipo de interfaz: Reporte de cartera vencida por cliente
Fuente: Elaboración propia

Gestión de roles y permisos.

Dentro de la gestión de roles y permisos, solo se establecen las pantallas necesarias para gestionar los roles, y asignar permisos por defecto a los mismos. La Figura 29 muestra la página principal de los roles y permisos, en la cual se listan cada uno de ellos con las opciones de visualizar, editar y eliminar. Es importante mencionar que el rol del administrador se encuentra por defecto, por lo que no podrá ser eliminado ni modificado.

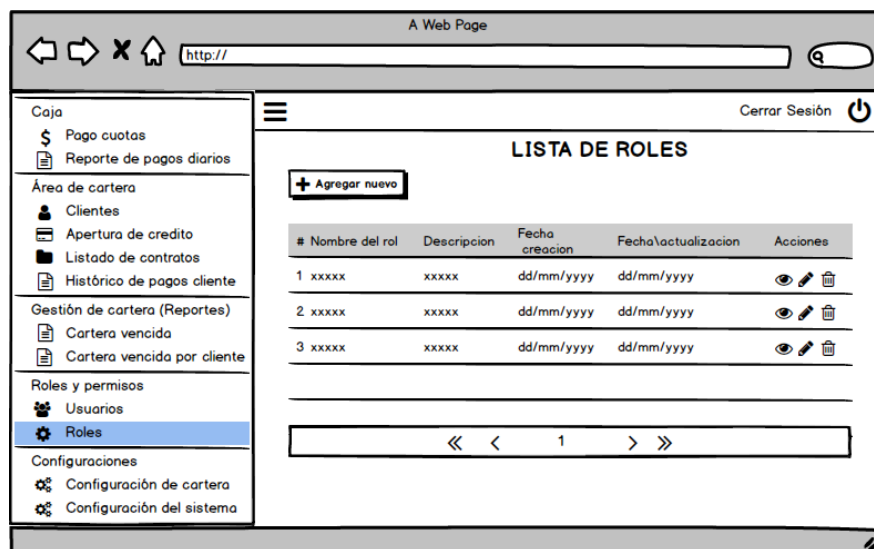


Figura 29 Prototipo de interfaz: listado de roles

Fuente: Elaboración propia

Para crear un nuevo rol o editar la información del mismo, se emplea una ventana flotante, en la que se ingresa el nombre y la descripción, y se asignan los permisos que el rol va a tener por defecto como muestra la Figura 30.

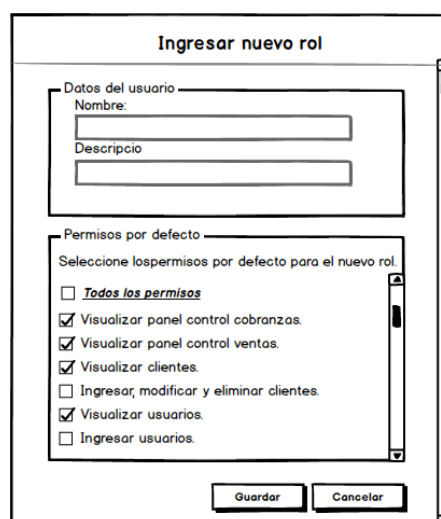


Figura 30 Prototipo de interfaz: Ingresar nuevo rol

Fuente: Elaboración propia

Configuraciones.

Para la configuración de los campos requeridos para el control de cartera y que son propiamente de la empresa, se diseña la ventana presentada en la *Figura 31* con la posibilidad de gestionar (ingresar, eliminar, visualizar) los métodos de pago, y las entidades bancarias.

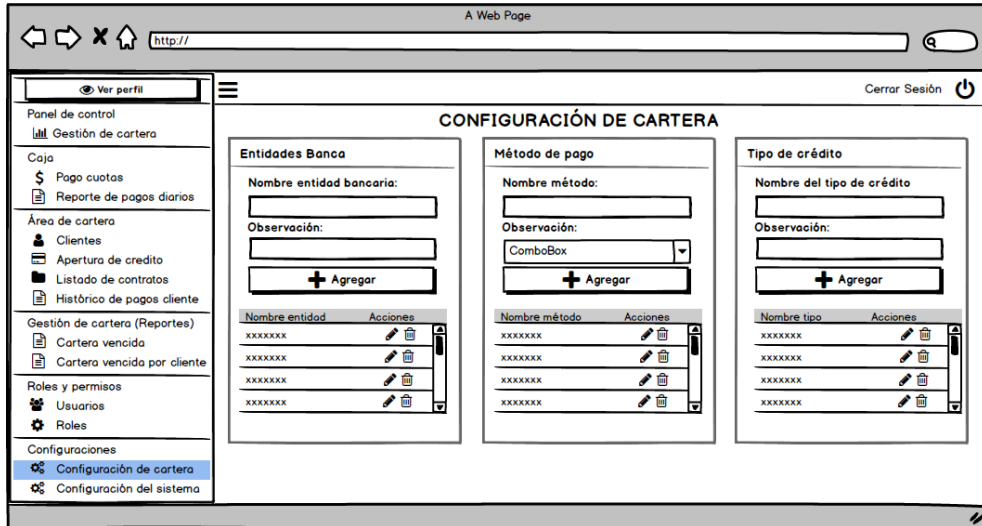


Figura 31 Prototipo de interfaz: configuración de cartera
Fuente: Elaboración propia

Gestión de configuración del sistema.

Dentro de la *Figura 32* se visualiza la página para la gestión de campos básicos para el sistema, como son el ingreso, eliminación, y visualización de estado civil y estado laboral.

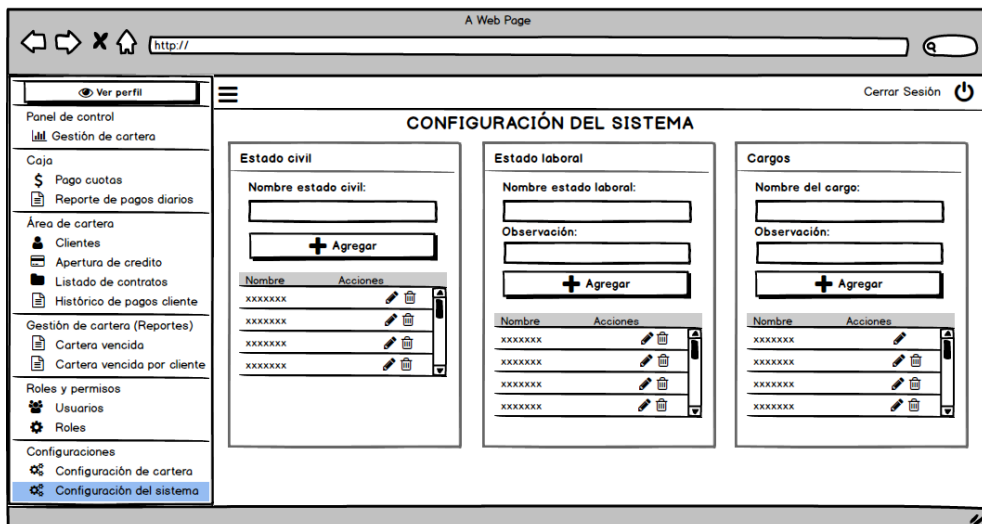


Figura 32 Prototipo de interfaz: configuración del sistema
Fuente: Elaboración propia

2.4.2.4 Diseño de diagramas UML

Mediante el uso de diagramas UML es posible conocer la estructuración de la información y la forma en que funciona el sistema a diseñar. Para la presente propuesta se ha considerado relevante el uso de los diagramas de caso de uso, de manera que se pueda identificar las funcionalidades del sistema en cuanto a los diferentes tipos de usuarios.

➤ Diagrama de casos de uso

El diagrama de casos de uso generalizado presentado en la *Figura 72*, muestra los módulos del sistema de manera global, indicando su acceso según el tipo de usuario. Los módulos pueden poseer sub-módulos, y estos a su vez actividades.

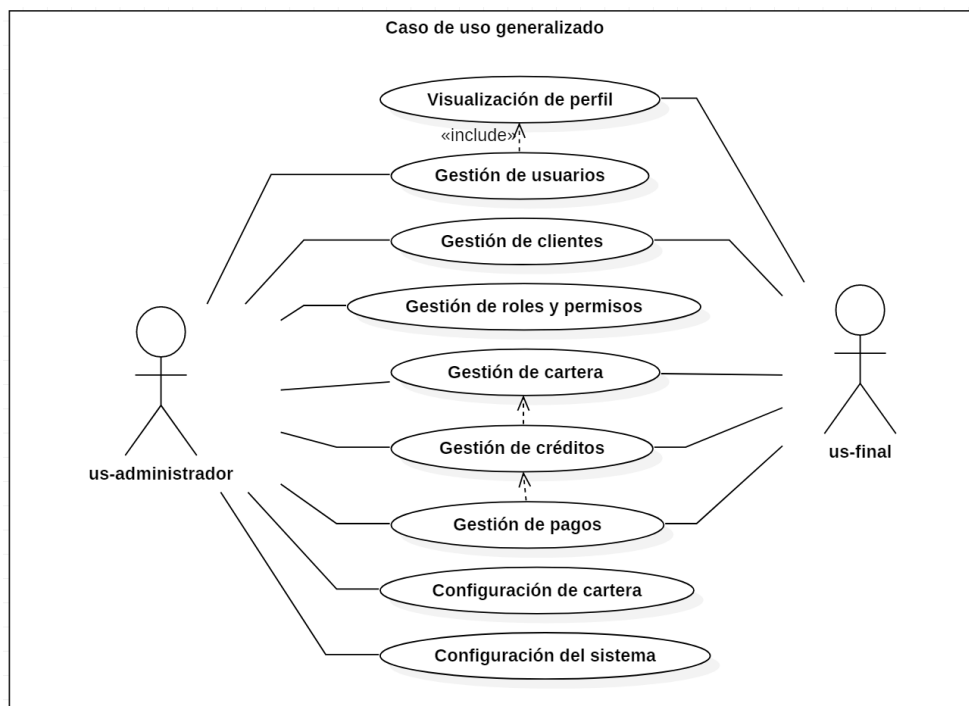


Figura 33 Caso de uso generalizado

Fuente: Elaboración propia

Para detallar el procedimiento y funcionalidades de cada módulo se detallan casos de uso especializados, presentados en el **ANEXO E**.

2.5 EJECUCIÓN Y/O ENSAMBLAJE DEL PROTOTIPO.

La etapa de desarrollo dentro de la metodología SWIRL, se basa en la codificación de las funcionalidades del sistema, considerando los prototipos de interfaces de usuarios diseñados, las tecnologías seleccionadas dentro de la arquitectura de software, y los patrones de diseños seleccionados en las etapas anteriores. A continuación, se presentan las páginas desarrolladas para cada uno de los procesos de la propuesta.

2.5.1 Página principal

La primera página que se visualiza al iniciar el sistema es la página de bienvenida en la cual el usuario podrá iniciar sesión, haciendo clic en el botón superior.



Figura 34 Implementación del prototipo: Página principal

Fuente: Elaboración propia

Luego de esto se redirige al usuario a la pantalla del login, en la cual se debe verificar que el usuario (número de cédula), y contraseña sean correctos.

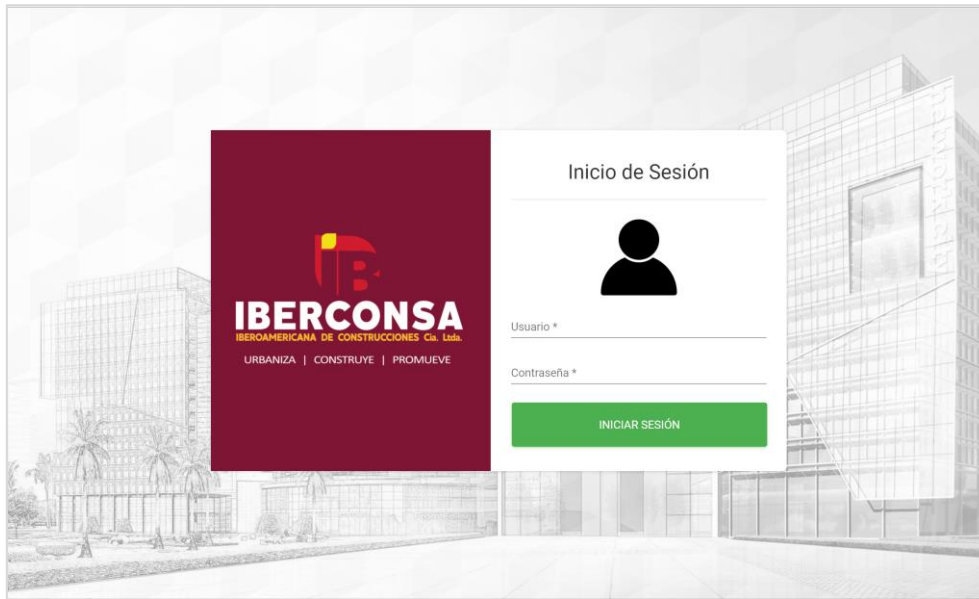


Figura 35 Implementación del prototipo: Inicio de sesión
Fuente: Elaboración propia

Una vez validado el usuario se accede a la página principal, en la cual se desplegará un menú con las opciones del sistema, según los permisos que se le han otorgado al momento de su creación.



Figura 36 Implementación del prototipo: Página home
Fuente: Elaboración propia

2.5.2 Páginas para la gestión de usuario

Las páginas relacionadas a la gestión de usuario solo son accesibles para el tipo administrador, y usuario con permisos asignados. La página principal de los usuarios muestra un listado con todos los usuarios registrados (activos y bloqueados), además de permitir ejecutar las acciones de gestión para cada uno, modificar, dar de baja, o desbloquear.

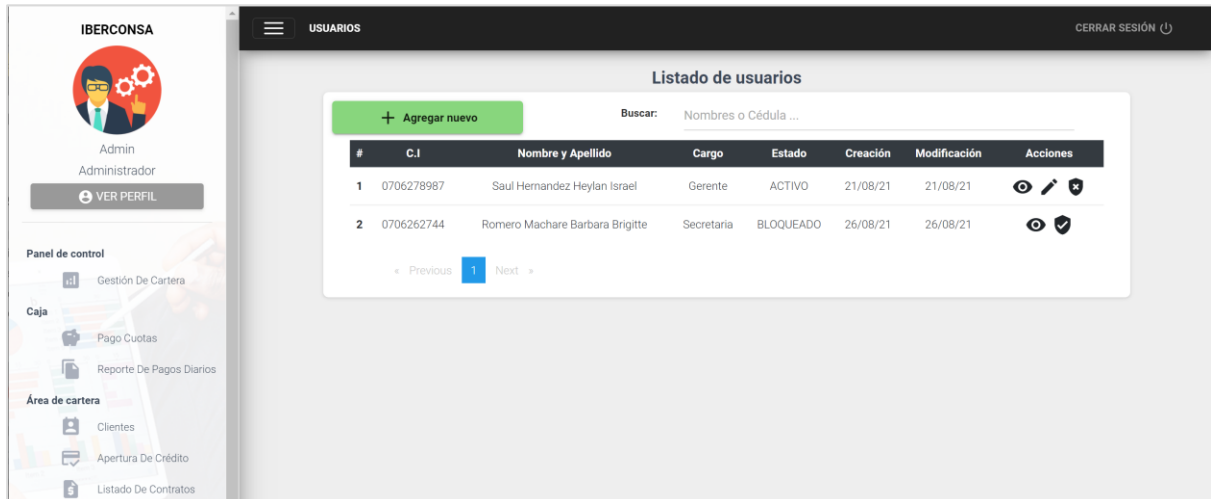


Figura 37 Implementación del prototipo: Lista de usuarios
Fuente: Elaboración propia

Al momento de habilitar un usuario, se emite una advertencia de que se le asignan los permisos que poseía anteriormente.

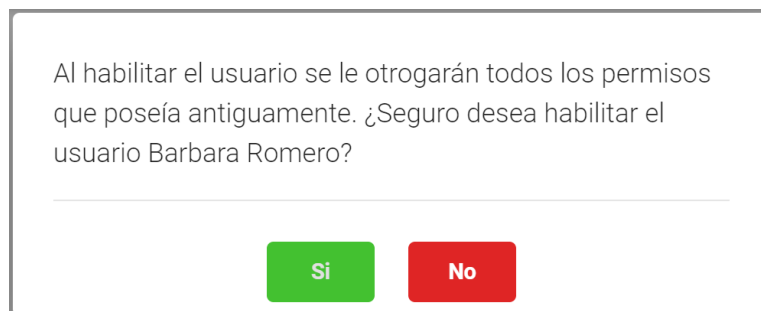


Figura 38 Implementación del prototipo: Habilitar usuario
Fuente: Elaboración propia

Al visualizar la información de cada usuario se detallan los datos personales, y se listan los roles y los permisos que se le han asignado.

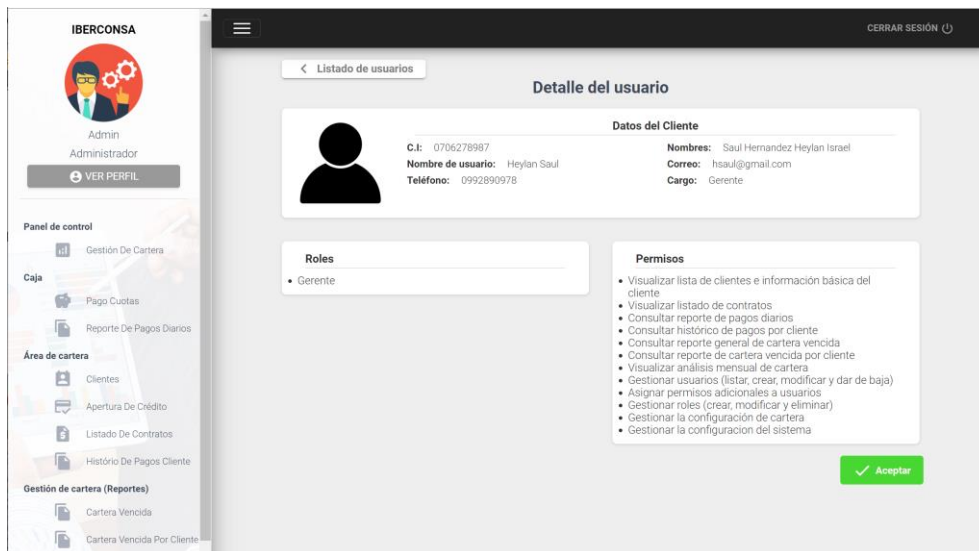


Figura 39 Implementación del prototipo: Visualizar información de usuario
Fuente: Elaboración propia

Para editar la información se muestra la misma ventana empleada al momento de ingresar un nuevo usuario, en donde es necesario ingresar los datos personales, seleccionar el o los roles a los cuales pertenece, y los permisos que se le otorgan a cada uno. Al momento de seleccionar el rol automáticamente se marcan los permisos por defecto que posee dicho rol, dando la opción de agregar o quitar algún permiso para dicho usuario.

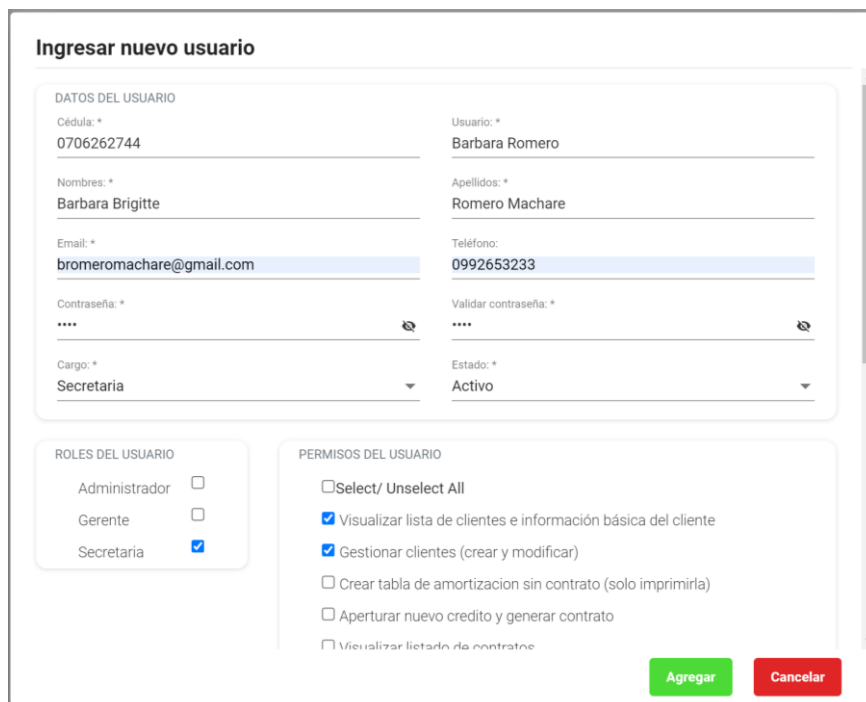


Figura 40 Implementación del prototipo: Ingreso de nuevo usuario
Fuente: Elaboración propia

2.5.3 Páginas para la gestión de clientes

La pantalla principal para la gestión de clientes se centra en un listado de los mismos, indicando los clientes que se encuentran registrados, y las acciones a ejecutar para cada uno de ellos, como son el visualizar información, editar y eliminar. Desde esta página se puede crear un nuevo cliente.

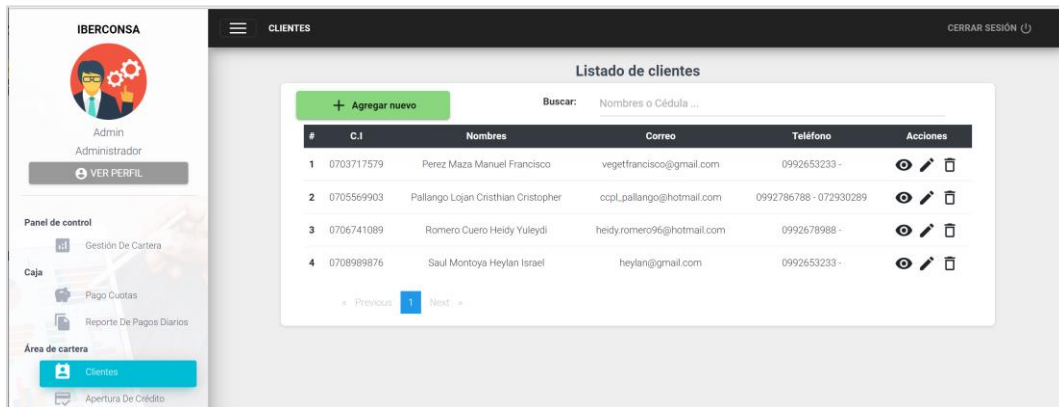


Figura 41 Implementación del prototipo: Listado de clientes
Fuente: Elaboración propia

La visualización de los detalles del cliente, incluye la información básica, los datos del cónyuge, la información laboral, y un listado de los contratos vinculados que posee.

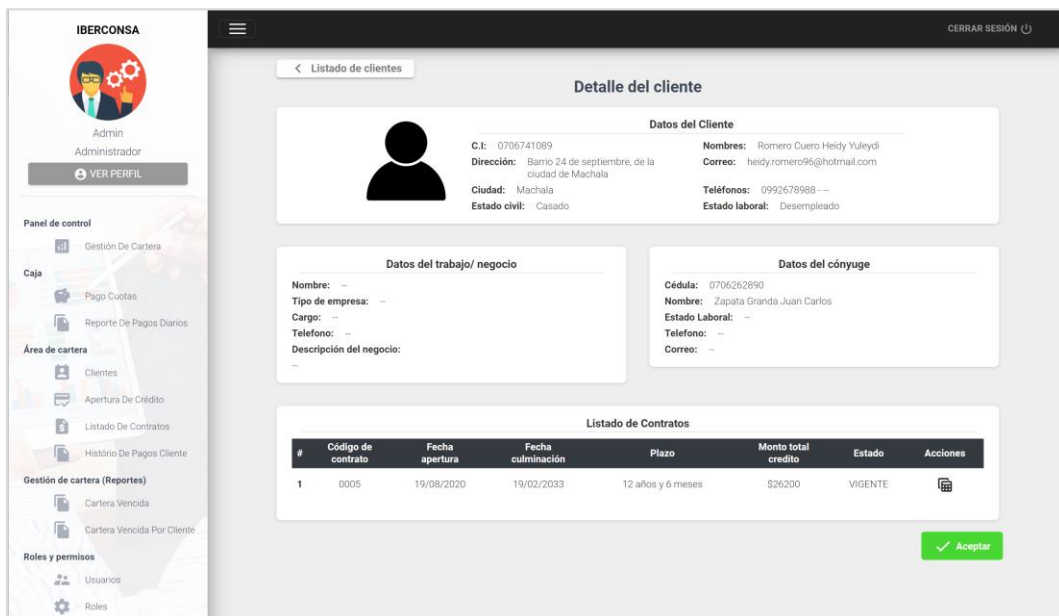


Figura 42 Implementación del prototipo: Información del cliente
Fuente: Elaboración propia

El formulario para modificar los datos es el mismo que se emplea para agregar un nuevo cliente, en el cual es requerido la información básica del cliente, los datos del cónyuge y la información laboral.

Ingresar nuevo cliente

DATOS DEL CLIENTE

Cédula * 0703079899 Nombres * Juan David Apellidos * Romero Machare

Dirección * Tarqui y 14va norte Barrio 10 de Julio

Ciudad * Machala Estado Civil * Soltero Correo jdavid@gmail.com

Estado Laboral * Desempleado Telefono 1 * 0992786722 Telefono 2 072930178

DATOS DEL CÓNYUGE

Cedula 🗑️ Limpiar datos

Nombres _____ Apellidos _____

Correo _____ Teléfono _____ Estado Laboral _____

DATOS DEL TRABAJO / NEGOCIO


Agregar
Cancelar

Figura 43 Implementación del prototipo: Ingresar nuevo cliente
Fuente: Elaboración propia

2.5.4 Páginas para la gestión de contratos

La página principal para la gestión de contratos muestra un listado de todos los contratos generados, siendo filtrados por cliente, y permitiendo las acciones de visualizar y eliminar en cada uno de ellos.

IBERCONSA



Admin
Administrador

VER PERFIL

Panel de control

- Gestión De Cartera
- Caja**
- Pago Cuotas
- Reporte De Pagos Diarios
- Área de cartera**
- Cientes
- Apertura De Crédito
- Listado De Contratos**
- Historio De Pagos Cliente

LISTADO DE CONTRATOS
CERRAR SESIÓN

Listado de contratos de crédito

Buscar: Nombres o Cédula ...

+ Agregar nuevo

No. contrato	C.I.	Nombres	Fecha de creación	Tipo de crédito	Años	Monto	Tasa de interés	Estado	Acciones
0001	0708989876	Saul Montoya Heylan Israel	23/08/2021	Vivienda	11 años y 3 meses	\$ 13100	7.5%	VIGENTE	👁️ 🗑️
0003	0708989876	Saul Montoya Heylan Israel	15/01/2021	Vivienda	11 años y 3 meses	\$ 13100	7.5%	VIGENTE	👁️ 🗑️
0004	0703717579	Perez Maza Manuel Francisco	15/07/2020	Vivienda	12 años y 1 mes	\$ 13100	7.5%	VIGENTE	👁️ 🗑️
0005	0706741089	Romero Cuero Heidy Yuleydi	20/08/2020	Vivienda	12 años y 6 meses	\$ 26200	7.5%	VIGENTE	👁️ 🗑️

◀ Previous 1 Next ▶

Figura 44 Implementación del prototipo: Listado de contratos
Fuente: Elaboración propia

La opción de eliminar solo se encuentra visible para el administrador, y para los usuarios que posean el permiso asignado. Dentro de la página de listado de contratos es posible crear un nuevo crédito, lo cual redirigirá a la página de “Aperturar nuevo crédito”, el cual también es posible acceder mediante la opción en el menú lateral.

Para la apertura de un nuevo crédito es necesario ingresar el número de cédula y una vez que encuentre los datos del cliente, se procede a completar los valores del crédito y de la entrada. Una vez completado todos los campos se procede a crear la tabla de amortización, para posteriormente habilitarse el botón de vincular a contrato. Para la migración de datos, se diseña un botón de ingreso de contrato ya existente el cual permite modificar los campos autogenerados como código, y fechas.

No. periodo	Fecha vencimiento	Valor dividendo	Interés	Capital pagado	Saldo del capital
0		\$	\$	\$	\$ 6000.00
1	2020-08-20	\$ 3000.00	\$ 0.00	\$ 100.00	\$ 3000.00
2	2020-09-20	\$ 100.00	\$ 0.00	\$ 100.00	\$ 2900.00
3	2020-10-20	\$ 100.00	\$ 0.00	\$ 100.00	\$ 2800.00
4	2020-11-20	\$ 100.00	\$ 0.00	\$ 100.00	\$ 2700.00
5	2020-12-20	\$ 100.00	\$ 0.00	\$ 100.00	\$ 2600.00
6	2021-01-20	\$ 100.00	\$ 0.00	\$ 100.00	\$ 2500.00

Figura 45 Implementación del prototipo: Aperturar nuevo crédito
Fuente: Elaboración propia

Una vez completado los datos para el cálculo del crédito y de la entrada, se muestra la página del detalle del contrato, en el cual se ingresa la información adicional requerida, como número de terrenos, código de lotes, y beneficiarios en caso de fallecimiento.

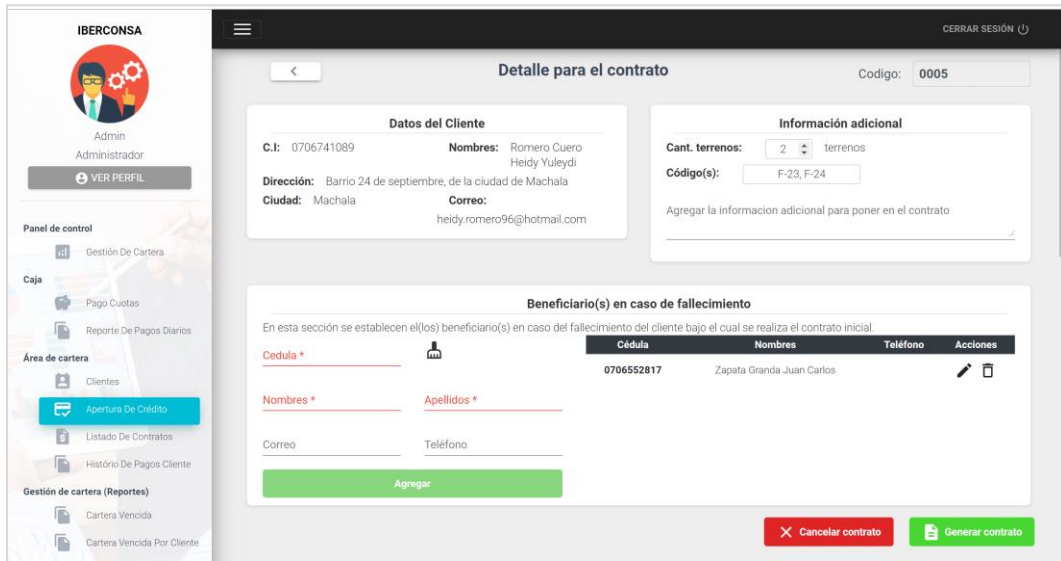


Figura 46 Implementación del prototipo: Detalles de nuevo contrato

Fuente: Elaboración propia

Una vez completado los campos necesarios se procede a generar el contrato, en donde se visualizará la redacción del mismo, con la opción del ingreso de la imagen del lote comprado. Al guardar el contrato se genera un archivo pdf con la información detallada del mismo.



Figura 47 Implementación del prototipo: Confirmación de contrato

Fuente: Elaboración propia

2.5.5 Páginas para la gestión de pagos

Al momento de realizar un pago, es posible seleccionar dos métodos, abonos o pago total de la cuota. Es necesario ingresar el número de cédula del cliente ya que, mediante la consulta de los datos, se visualizará el código de los contratos que tenga vinculados. La fecha estipulada para la consulta de las cuotas es ajustable, siendo por defecto la fecha actual, esto se realiza en caso de querer abonar o pagar cuentas de meses próximos. Una vez seleccionada la fecha y el contrato, se da clic en consultar, con lo cual se visualizará una tabla con todas las cuotas dentro de ese periodo e indicando las cuotas vencidas (con color rojo), y las vigentes (con color blanco).

Luego se selecciona la cuota a cancelar, y se da clic en abonar o en pagar, siendo el caso respectivo. Para realizar un abono se visualiza la información por defecto de la cuota, y requiriendo el ingreso del valor del abono, el método de pago, la entidad bancaria y el número del recibo del banco en caso de ser mediante documento contable.

The screenshot displays the 'PAGO CUOTAS' interface. On the left is a sidebar with the user profile 'Admin Administrador' and navigation options like 'Gestión De Cartera', 'Pago Cuotas', and 'Reporte De Pagos Diarios'. The main area is titled 'Ingreso de pago' and contains a 'Comprobante de ingreso' (C.I. AGOSTO21.002) and 'N° de comprobante interno' (2021 F-23, F-24 ENT-3). Below this is the 'Datos del Cliente' section with fields for C.I. (0706741089), Nombres (Romero Cuero Heidy Yuleydi), and Correo (heidy.romero96@hotmail.com). A 'Lista de cuotas' table shows four entries with columns for Tipo, Periodo, Fecha de vencimiento, Valor a cancelar, and Acciones (Pagar, Abonar). The 'Detalles del abono' section includes fields for Tipo de cuota (ENTRADA), Periodo (3), Fecha Vencimiento (2020-10-20), Valor dividendo (\$ 100), Valor interés (\$ 0), Valor capital (\$ 100), Fecha de emisión (26/08/2021), Valor del abono (\$ 50), Método de pago (Efectivo), Entidad bancaria, and N° recibo banco. A green 'Guardar pago' button is at the bottom.

Figura 48 Implementación del prototipo: Ingresar abono

Fuente: Elaboración propia

En caso de que se realice un pago completo, la información a ingresar consta del valor cancelado, el método de pago, la entidad bancaria, y el número de recibo del banco. Adicional a esto se incrementa un campo de interés de mora, en caso de que la cuota se encuentre vencida.

Figura 49 Implementación del prototipo: Ingresar pago
Fuente: Elaboración propia

Una vez ingresado los datos, se guarda el pago y se genera un comprobante de pago el cual puede ser impreso.

Figura 50 Implementación del prototipo: Ejemplo de comprobante de pago
Fuente: Elaboración propia

Es posible visualizar un reporte de pagos diarios, en el cual se listan los pagos realizados en un día en específico, por defecto el día actual.

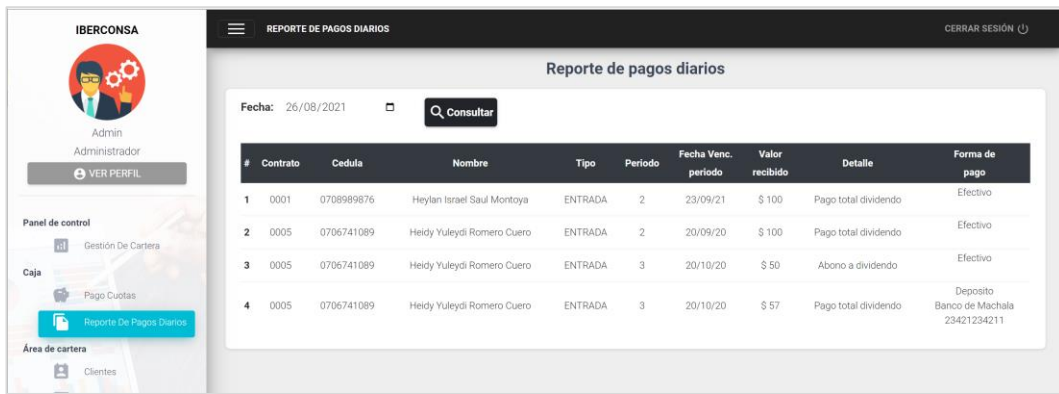


Figura 51 Implementación del prototipo: Reporte de pagos diarios
Fuente: Elaboración propia

De igual forma se visualiza el histórico de pagos por cliente, al ingresar el número de cédula se realizará una búsqueda listando todos los contratos que tiene vinculados. Al seleccionar la opción de “ver pagos” se detallarán todos los pagos realizados a dicho contrato, incluyendo los abonos realizados.

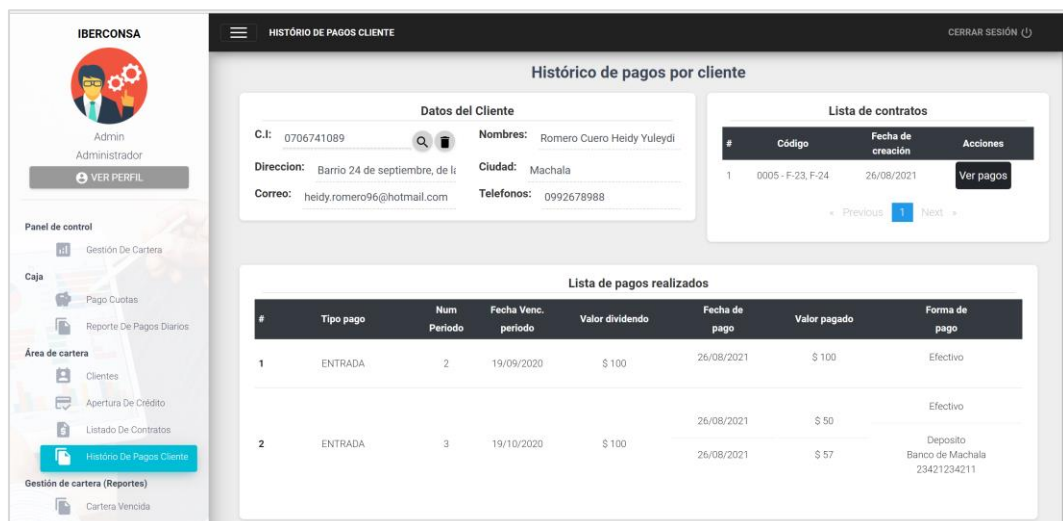


Figura 52 Implementación del prototipo: Histórico de pagos por cliente
Fuente: Elaboración propia

2.5.6 Páginas para la gestión de cartera

Dentro de la gestión de cartera se incluye el panel administrativo, en donde se muestra el ingreso mensual de las cuotas cobradas, separando las cuotas que ingresan a las cuentas de caja, y las que ingresan a las cuentas de banco; además de mostrar el monto de cuotas por cobrar del mes actual. Adicional a esto, se muestra un listado con las cuotas a cobrar el mes actual, y otra con los clientes con cuotas en mora.

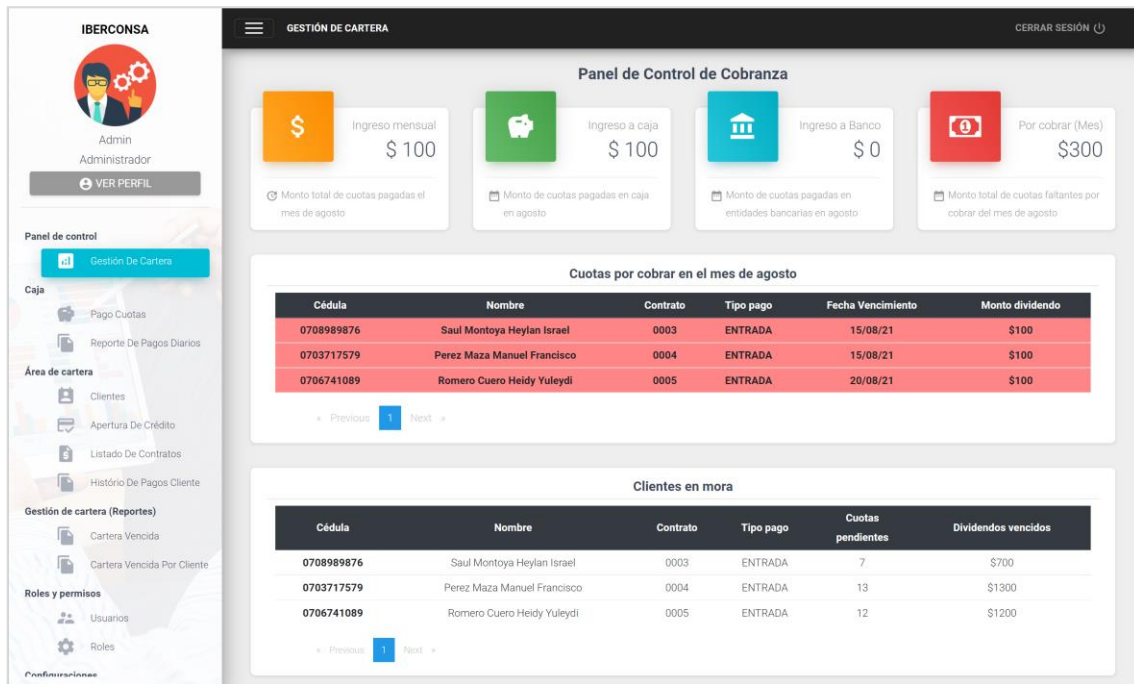


Figura 53 Implementación del prototipo: Panel administrativo
Fuente: Elaboración propia

La cartera vencida muestra las cuotas pendientes de pago de todos los clientes y de todos los contratos, cada una muestra la fecha de vencimiento de dicha cuota.

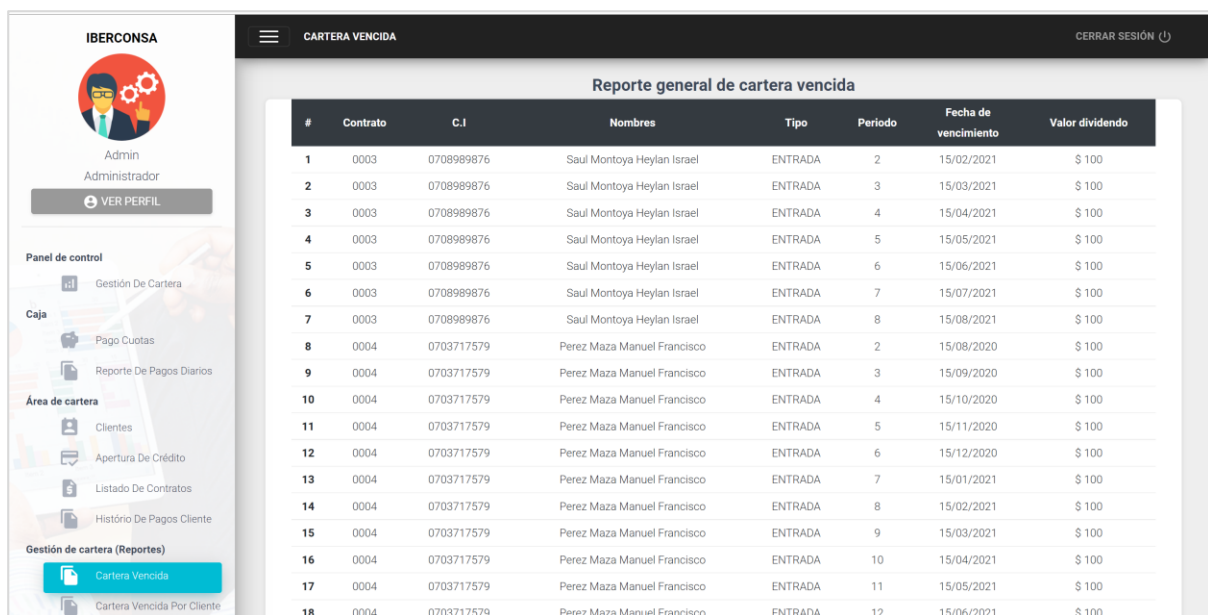


Figura 54 Implementación del prototipo: Reporte general de cartera vencida
Fuente: Elaboración propia

De igual forma se visualiza el reporte de cartera vencida por cliente, en donde se gestiona también las notificaciones realizadas al mismo en caso de atrasos en los pagos de las cuotas.

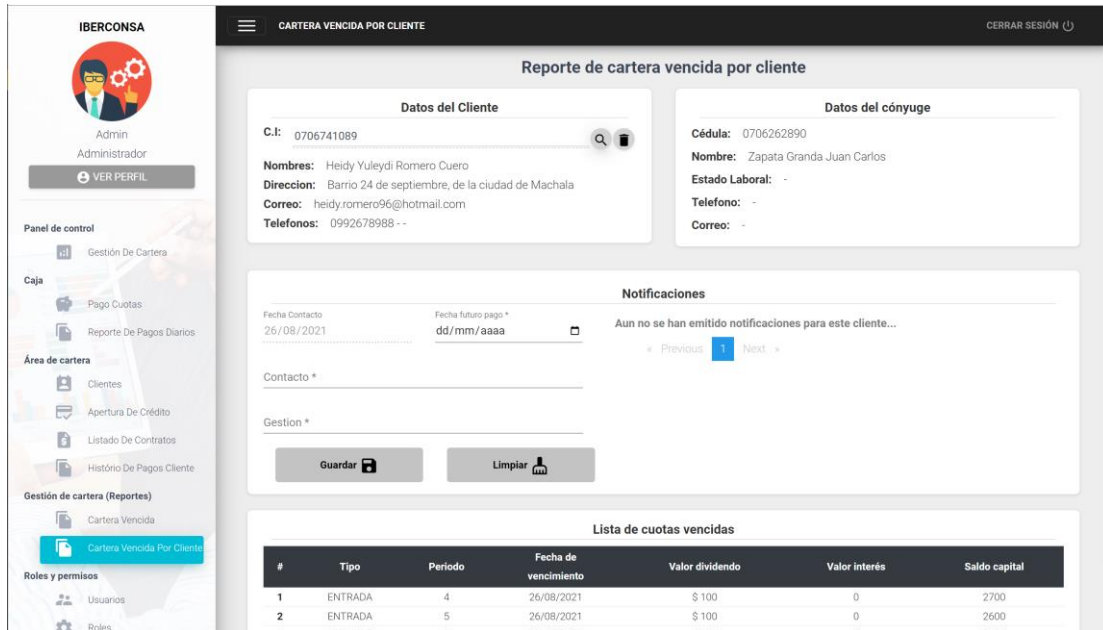


Figura 55 Implementación del prototipo: Reporte de cartera vencida por cliente
Fuente: Elaboración propia

2.5.7 Diseño según tendencia.

Dentro del sistema se implementó el diseño responsivo según las pautas establecidas en la fase de modelado beneficiando de esta manera la usabilidad del sistema y adaptabilidad a los diferentes dispositivos.

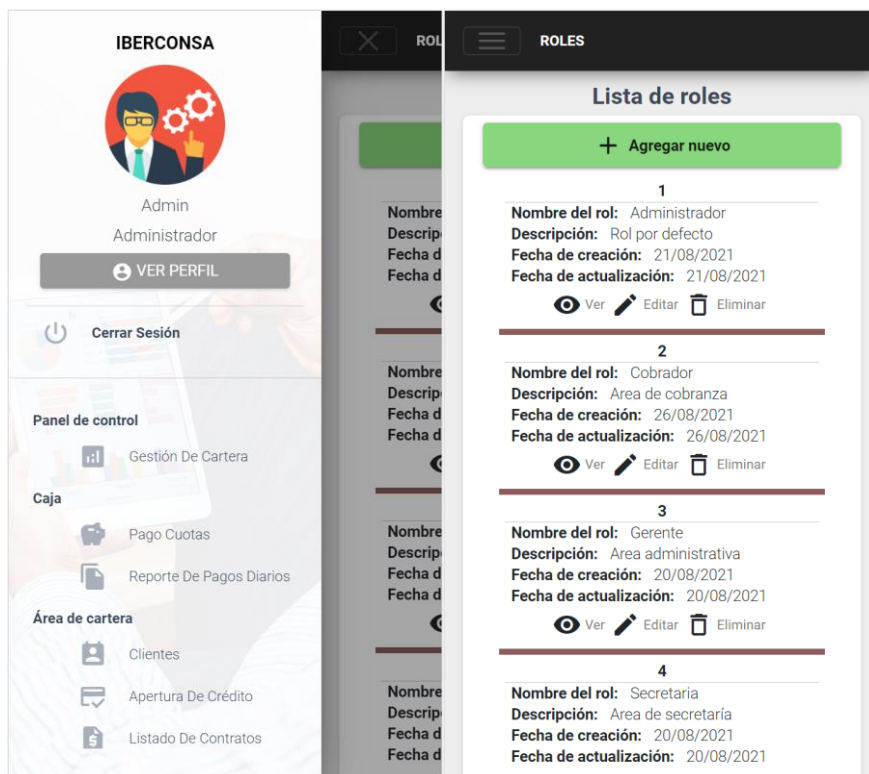


Figura 56 Implementación del prototipo: Diseño responsivo
Fuente: Elaboración propia

3 Capítulo III. Evaluación del prototipo.

3.1 PLAN DE EVALUACIÓN.

Seguir un plan de evaluación durante el proceso de desarrollo permite detectar los posibles errores y solucionarlos a tiempo, de manera que se entregue un producto funcional y de calidad. El plan de evaluación a considerar se basa en las tres pruebas de la metodología SWIRL, evaluación de calidad, evaluación de tendencia, y evaluación con herramientas SEO.

3.1.1 Evaluación de calidad

La calidad se evalúa acorde a las métricas de la norma ISO/IEC 9126, según se explica en la metodología SWIRL, empleando como herramienta un cuestionario de control con los indicadores de la métrica. Para facilitar su evaluación se emplea la escala de Likert valorando desde 1 (Excelente), hasta 5 (Pésimo) como se detalla en la *Tabla 24*.

Tabla 24: Escala de Likert en evaluación de calidad.

Escala de Likert	
Valoración	Interpretación de resultados
1	Pésimo
2	Malo
3	Regular
4	Bueno
5	Excelente

Fuente: [21]

La finalidad de la evaluación de calidad es identificar si la aplicación cumple con los criterios de funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad, y operatividad.

3.1.2 Evaluación mediante herramientas SEO

Para la evaluación de la tendencia de diseño, y en términos de características SEO, SWIRL propone la inclusión de herramientas SEO que permitan la evaluación del contenido semántico de la aplicación, y proporcionen una valoración en términos de posicionamiento, rendimiento, y en algunas ocasiones accesibilidad, y usabilidad. Dentro de la presente propuesta se harán uso de las herramientas SEO presentadas en la *Tabla 25* detallando de igual forma su enfoque principal.

Tabla 25: Análisis de herramientas SEO.

ANÁLISIS DE HERRAMIENTA SEO					
Herramienta	Parámetro				
	Accesibilidad	Funcionalidad	Eficiencia	Portabilidad	Posicionamiento
HTML Validator		X			
GtMetrix			X		
Functional Accessibility Evaluator	X				
Website Grader				X	
Seo Site Checkup				X	X

Fuente: [39]

3.2 RESULTADOS DE LA EVALUACIÓN.

3.2.1 Evaluación de calidad

En la *Tabla 26*, se detallan los resultados de la evaluación de la calidad realizada a la aplicación según las características de la norma ISO/IEC 9126, para ello se consideran las subcaracterísticas inmersas en cada una de ellas.

Tabla 26: Escala de Likert en evaluación de calidad.

Característica	Sub-característica	Criterio	Valoración
Funcionalidad	Adecuación	¿El software tiene la capacidad de cumplir con las tareas por las cuales se desarrolló?	5
	Exactitud	¿El software cumple con los resultados solicitados de manera correcta y precisión?	5
	Interoperabilidad	¿El sistema puede interactuar con otro sistema?	3
	Conformidad	¿El sistema cumple con los estándares referentes a la funcionalidad?	5
	Seguridad	¿El sistema impide el acceso no autorizado?	5
% DE CUMPLIMIENTO EN FUNCIONALIDAD			92%
Eficiencia	Comportamiento en el tiempo.	¿El tiempo de respuesta es el adecuado al usar el sistema?	4

Característica	Sub-característica	Criterio	Valoración
	Consumo de recursos.	¿Los recursos del sistema son ocupados de manera adecuada?	4
% DE CUMPLIMIENTO EN EFICIENCIA			80%
Usabilidad	Comprensibilidad	¿Es fácil de entender el funcionamiento del sistema?	5
	Facilidad de aprender	¿Es fácil de aprender a usar el sistema?	5
	Operabilidad	¿El sistema es fácil de operar y controlarlo?	4
% DE CUMPLIMIENTO EN USABILIDAD			86.67%
Portabilidad	Facilidad de instalación	¿El software es fácil de instalar?	5
	Reemplazabilidad	¿Es fácil reemplazar el software por otro con similares funciones?	4
	Conformidad	¿El software está desarrollado con algún lenguaje multiplataforma?	4
	Coexistencia	¿El software puede interactuar y compartir recursos con otro?	3
	Adaptabilidad	¿Es fácil adaptar el sistema en diversos entornos de trabajo?	5
% DE CUMPLIMIENTO EN PORTABILIDAD			84%
Mantenibilidad	Capacidad de análisis	¿Es fácil identificar fallas para ser corregidas o modificadas?	5
	Capacidad de modificación	¿Es fácil de modificar el sistema ante cualquier cambio del mismo?	4
	Estabilidad	¿El software se mantiene funcional pese a cambios?	4
	Facilidad de prueba	¿Es fácil realizar pruebas del sistema?	5
% DE CUMPLIMIENTO EN MANTENIBILIDAD			90%
Confiabilidad	Nivel de madurez	¿El software cumple con el manejo de errores en caso de percibirlos?	5
	Tolerancia a fallas	¿El software se mantiene funcionando ante la presencia de fallas del usuario o del sistema?	5
	Recuperación	¿El software es capaz de recuperar datos en caso de fallas?	3
% DE CUMPLIMIENTO EN CONFIABILIDAD			86.67%
TOTAL DE CUMPLIMIENTO EN CALIDAD			86.56%

Fuente: Elaboración propia

Los resultados obtenidos mediante la evaluación de calidad, muestra que el sistema cumple en un 86.56% las características de calidad de la norma ISO 9126, lo cual se

considera bueno; cumpliendo en funcionalidad un 92%, eficiencia 80%, usabilidad 86.67%, portabilidad 84%, mantenibilidad 90%, y finalmente confiabilidad 86.67%.

El uso de las tecnologías y tendencias seleccionadas para el desarrollo del sistema, beneficiaron al cumplimiento de los criterios de calidad evaluados, principalmente en la característica de mantenibilidad, permitiendo realizar cambios sin afectar las demás funcionalidades del sistema, así como la portabilidad y usabilidad, ya que al emplear Angular.js brinda una interfaz de usuario amigable y adaptable a los diferentes dispositivos.

Por otro lado, existen características que no son completadas en su totalidad, como la Recuperabilidad, ya que el sistema no respalda la información de la base de datos en caso de fallas. Un punto clave a considerar es la eficiencia, sin embargo, el sistema no contempla una alta satisfacción en este criterio debido a un lento tiempo de respuesta de la base de datos.

3.2.2 Evaluación mediante herramientas SEO

- **HTML VALIDATOR**

Mediante la ejecución de la herramienta HTML Validator se logró evaluar la estructura, y código HTML y CSS, consiguiendo como resultado una valoración correcta, sin errores o advertencias.

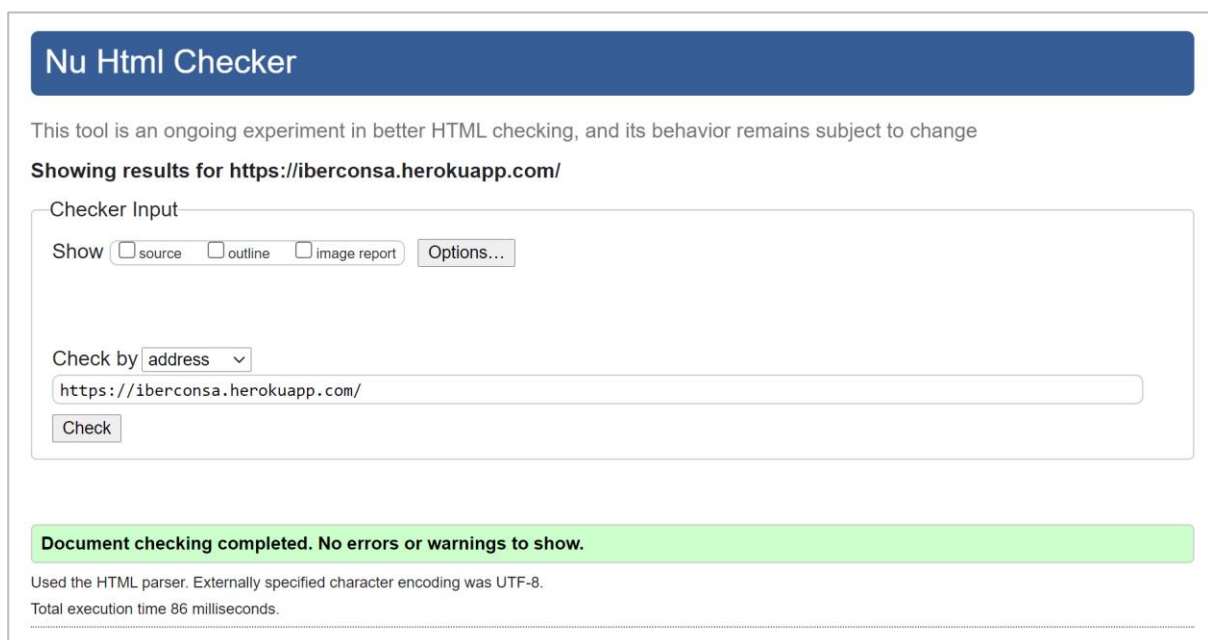


Figura 57 Prueba del sistema: HTML Validator

Fuente: Elaboración propia

- **GTmetrix**

La herramienta GTmetrix permite evaluar la aplicación en cuanto al rendimiento, estructura, y tiempo de respuesta. La aplicación tuvo resultados favorables, teniendo una valoración de B, con 86% en rendimiento, 82% en estructura, y 1.1s en tiempo de respuesta, lo cual es en promedio bueno.

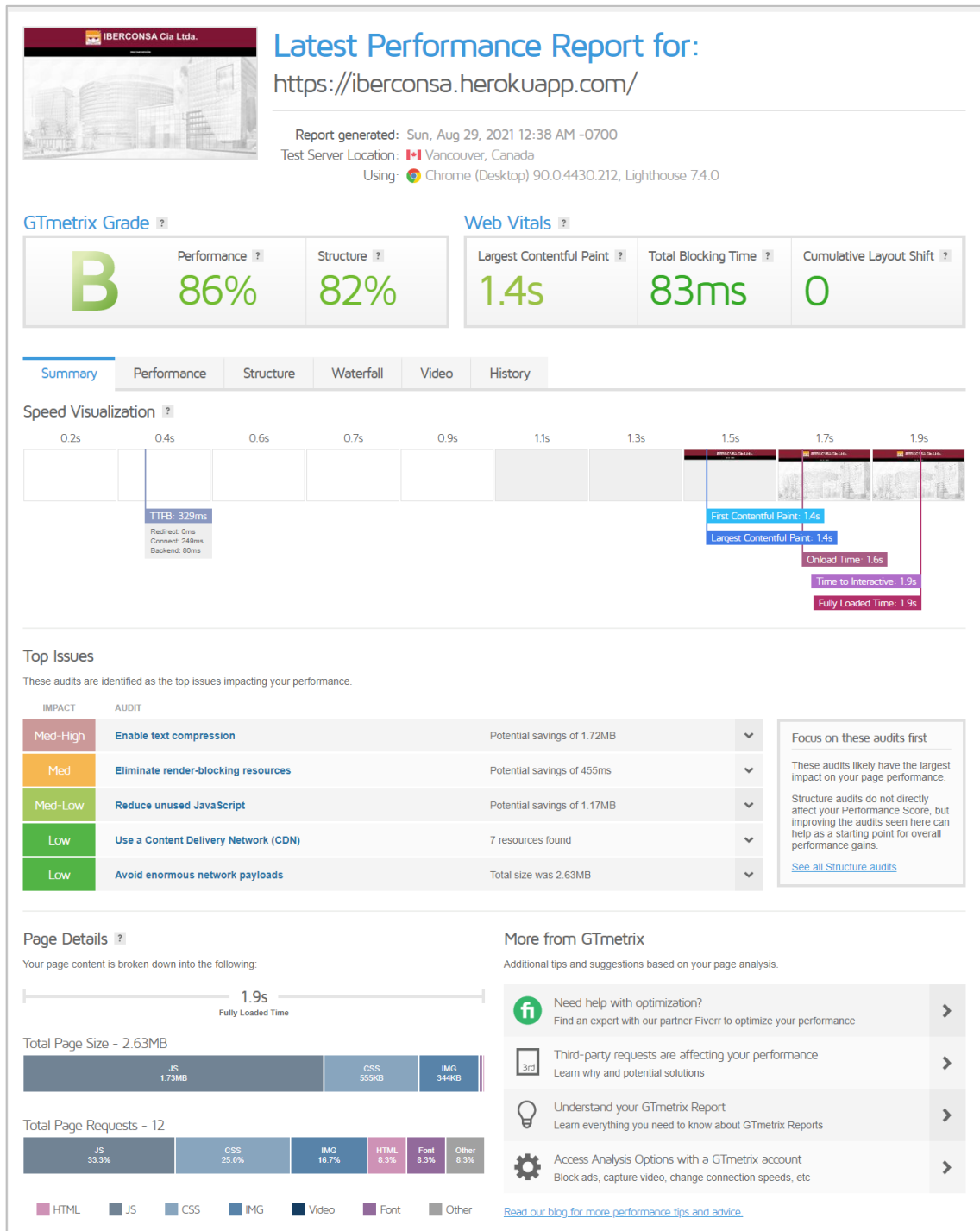


Figura 58 Prueba del sistema: GTmetrix

Fuente: Elaboración propia

- **Functional Accessibility Evaluator**

La *Figura 59* muestra los resultados de la evaluación de mediante la herramienta Functional Accessibility Evaluator (FAE), con la cual se obtuvieron resultados buenos /un puntaje de 14). Esta herramienta evalúa el sitio según los requerimientos de las “Pautas de accesibilidad al contenido Web 2.0” de la WCAG 2.0 [40] en dos niveles A, y los criterios de éxito del AA.

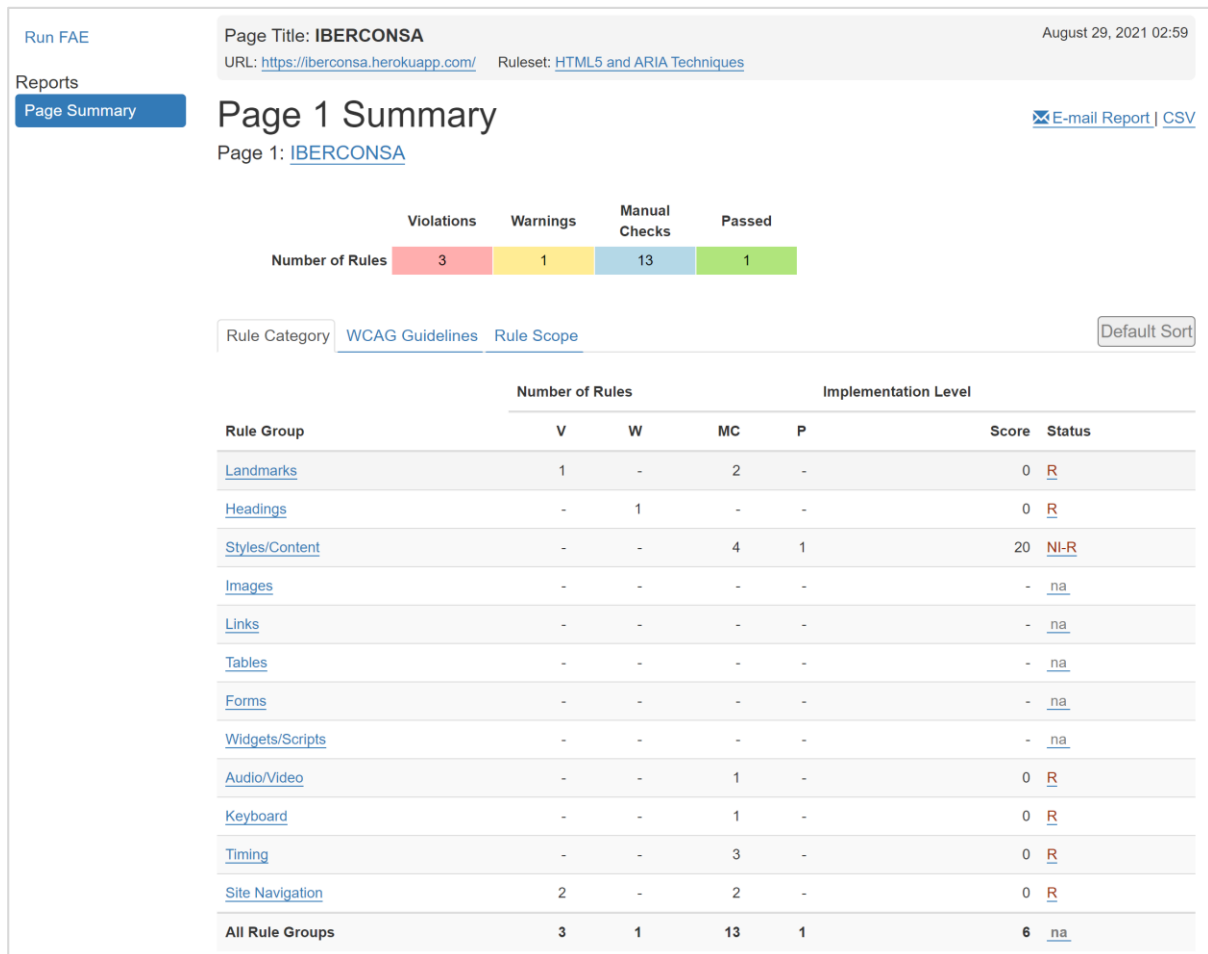


Figura 59 Prueba del sistema: Functional Accessibility Evaluator

Fuente: Elaboración propia

- **Website Grader**

Mediante el uso de esta herramienta se evalúan las características de rendimiento, SEO, adaptabilidad Móvil, y seguridad. Los resultados se muestran en la *Figura 60*, la cual presenta un grado de valoración favorable del 72%, obteniendo una perfecta calificación en cuanto a adaptabilidad móvil con 30/30 y una baja calificación en rendimiento con 17/30.

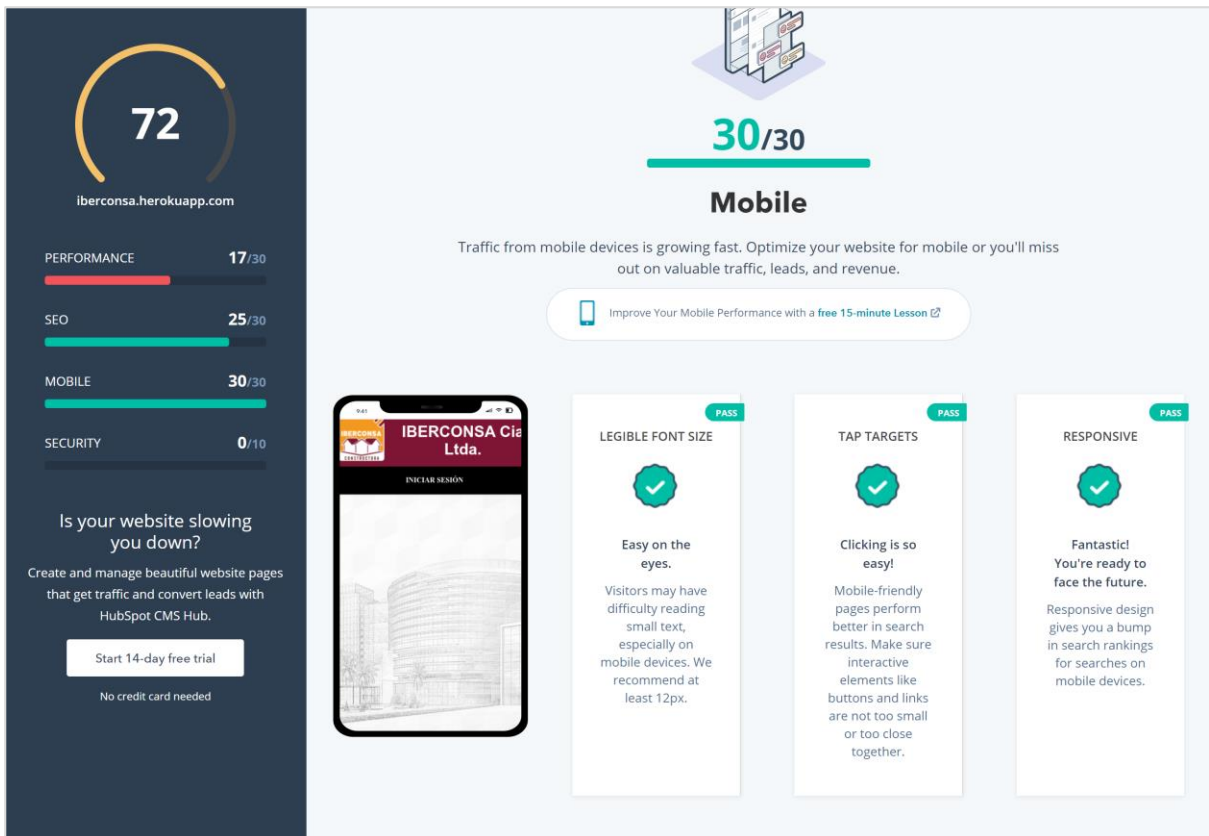


Figura 60 Prueba del sistema: Website Grader

Fuente: Elaboración propia

- **SEO Site Checkup**

Esta herramienta permite la evaluación de la estructura semántica de la aplicación web y brinda las recomendaciones para mejorar el SEO. Los resultados obtenidos en la *Figura 61* muestran una valoración de 72/100 en la cual se cumplen 41 aspectos de los 52 evaluados.

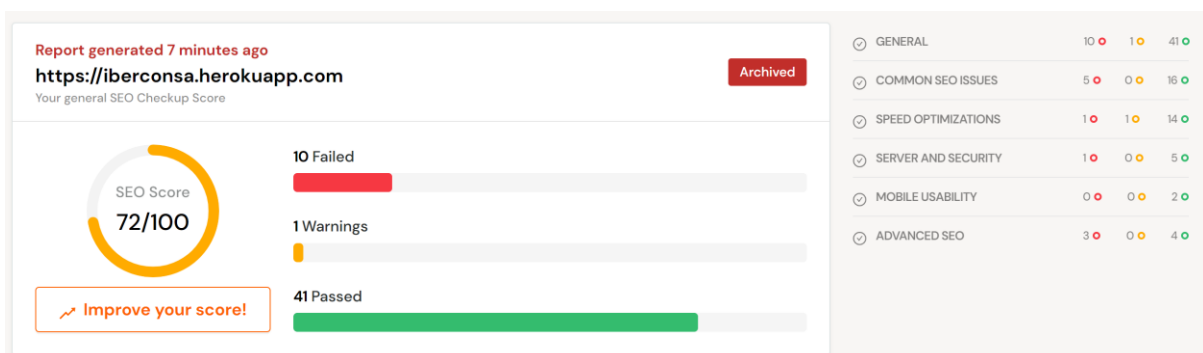


Figura 61 Prueba del sistema: SEO Site Checkup

Fuente: Elaboración propia

3.3 CONCLUSIONES.

Finalizado el proceso de desarrollo de la aplicación para la gestión de cartera mediante la metodología SWIRL se logra concluir lo siguiente:

- El uso de la metodología SWIRL dentro de la propuesta tecnológica benefició al proceso de desarrollo de la aplicación, gracias a la comunicación constante con el cliente, y al proceso iterativo en la ejecución de las diferentes actividades por fase. Las mismas que ayudaron a mantener una estructura estable en las funcionalidades del proyecto durante la inclusión de cambios y nuevos requerimientos.
- Mediante la aplicación de los modelos de recolección de información, y las actividades inmersas dentro de las etapas de análisis y planificación de la metodología SWIRL se logró elicitación de los requerimientos funcionales y no funcionales, así como las funcionalidades básicas que debe cumplir el sistema para satisfacer las necesidades y efectuar el proceso de gestión de cartera que contempla la empresa de manera eficiente.
- La aplicación de las diversas herramientas CASE en la etapa de diseño benefició al desarrollo del proyecto, así como la elaboración de modelados básicos y necesarios para el entendimiento de las funcionalidades y estructura del sistema por parte de los interesados, incluyendo las interfaces de usuario, modelados de bases de datos, y casos de uso, garantizando también el correcto entendimiento del alcance y requerimientos iniciales del sistema.
- Gracias al uso de las tecnologías de Angular.js, Node.js y MongoDB se logró implementar un sistema administrativo y diseñado por módulos que cubre el proceso de gestión de cartera de la empresa y podrá ser empleado para optimizar sus procesos; además de presentar un alto grado de disponibilidad y seguridad en el lado del servidor, y una interfaz amigable por el lado del cliente gracias a las herramientas que brindan las tecnologías seleccionadas.

3.4 RECOMENDACIONES.

Una vez concluida la propuesta tecnológica se plantean las siguientes recomendaciones para trabajos similares o futuros.

- Incluir la comunicación con los interesados al finalizar las iteraciones de cada fase, visualizando los resultados esperados y los obtenidos, de manera que los cambios requeridos sean adecuados a la planificación y estructura del proyecto sin generar riesgos.
- Seleccionar los frameworks y tecnologías para el desarrollo basados en un análisis de sus características, beneficios y desventajas con relación a los requerimientos del proyecto; así como la selección de tecnologías con amplio soporte con comunidades en línea, de manera que el proceso sea culminado de manera rápida, y sin errores.
- Identificar el proceso de negocio a optimizar desde diferentes escenarios, de manera que los requerimientos sean adaptables a las diferentes situaciones que se podrían presentar durante la ejecución del proceso, y no sea necesario una modificación en los requerimientos ya definidos o prototipos ya diseñados.
- Una vez que el sistema se encuentre funcional, realizar las pruebas del sistema respectivo empleando herramientas de validación en línea, corrigiendo los errores que se puedan presentar y considerando las advertencias que se emiten, garantizando así la calidad y correcto funcionamiento del sistema.

BIBLIOGRAFÍA

- [1] R. Alvarado, K. Acosta, y Y. V. Buonaffina, «Necesidad de los sistemas de información gerencial para la toma de decisiones en las organizaciones», *InterSedes*, vol. 19, n.º 39, pp. 17-31, jun. 2018, doi: 10.15517/isucr.v19i39.34067.
- [2] K. Eun-Jung, K. A-Hyun, y K. Sang-Soo, «Toward the understanding of the appropriation of ICT-based Smart-work and its impact on performance in organizations», *Technological Forecasting and Social Change*, vol. 171, p. 120994, oct. 2021, doi: 10.1016/j.techfore.2021.120994.
- [3] E. S. Pane, «Save And Loan Credit Information System Based On Web», *Journal of Applied Engineering and Technological Science*, vol. 2, n.º 1, nov. 2020, doi: 10.37385/jaets.v2i1.166.
- [4] A. L. Leal Fica, M. A. Aranguiz Casanova, y J. Gallegos Mardones, «Análisis de riesgo crediticio, propuesta del modelo Credit Scoring1», *Revista Facultad de Ciencias Económicas: Investigación y Reflexión*, vol. XXVI, n.º 1, pp. 181-207, 2018.
- [5] M. Rashidi, «The Role of Managers' Ability to Modify Credit Conditions and Reduce Share Returns spread», *Journal of Asset Management and Financing*, vol. 8, n.º 3, pp. 123-139, sep. 2020, doi: 10.22108/amf.2020.119807.1480.
- [6] B. Vogel-Heuser, J. Fischer, S. Feldmann, S. Ulewicz, y S. Rösch, «Modularity and architecture of PLC-based software for automated production Systems: An analysis in industrial companies», *Journal of Systems and Software*, vol. 131, pp. 35-62, sep. 2017, doi: 10.1016/j.jss.2017.05.051.
- [7] J. Medina Cruz, E. Pineda Ballesteros, y F. R. Téllez Acuña, «Requerimientos de software: prototipado, software heredado y análisis de documentos», *Ingeniería y Desarrollo*, vol. 37, n.º 2, pp. 327-345, 2019.
- [8] J. de D. Murillo Morera y S. Caamaño Polini, «Implementación De Un Servidor Ftp Utilizando El Modelo Cliente/Servidor Mediante El Uso De Sockets En Lenguaje C Unix Con El Fin De Mejorar Los Tiempos De Respuesta En La Red», *Uniciencia*, n.º 24, pp. 83-89, 2010.
- [9] J. Calvo, J. Gracia, y E. Bayo, «Robust design to optimize client-server bi-directional communication for structural analysis web applications or services», *Advances in Engineering Software*, vol. 112, pp. 136-146, oct. 2017, doi: 10.1016/j.advengsoft.2017.04.010.
- [10] L. F. Maldonado-Granados, I. E. Restrepo-Delgado, O. A. Murcia-Camargo, y J. P. Gómez-Gil, «Software for ontological and collaborative online representation of knowledge», *Ingeniería, investigación y tecnología*, vol. 19, n.º 2, pp. 147-158, 2018, doi: 10.22201/fi.25940732e.2018.19n2.013.
- [11] M. E. Uribe Macías, «Analysis of construction projects stakeholders from corporate social responsibility*», *Cuadernos de Administración (Universidad del Valle)*, vol. 33, n.º 58, pp. 58-76, 2017.
- [12] C. A. Gyöřödi, D. V. Dumșe-Burescu, D. R. Zmaranda, R. Ș. Gyöřödi, G. A. Gabor, y G. D. Pecherle, «Performance Analysis of NoSQL and Relational Databases with CouchDB and MySQL for Application's Data Storage», *Applied Sciences*, vol. 10, n.º 8524, p. 8524, nov. 2020, doi: 10.3390/app10238524.
- [13] K. I. Gómez Sotelo, C. Baron, P. Esteban, C. Y. A. G. Estrada, y L. de J. Laredo Velázquez, «How to find non-functional requirements in system developments», *IFAC-PapersOnLine*, vol. 51, n.º 11, pp. 1573-1578, ene. 2018, doi: 10.1016/j.ifacol.2018.08.272.

- [14] S. Islam y N. Evans, «Key Success Factors of PRINCE2 Project Management Method in Software Development Project», *International Journal of Engineering Materials and Manufacture*, vol. 5, n.º 3, jul. 2020, doi: 10.26776/ijemm.05.03.2020.02.
- [15] E. A. Kalinga, «Learning Software Development through Modeling Using Object Oriented Approach with Unified Modeling Language: A Case of an Online Interview System», *Journal of Learning for Development*, vol. 8, n.º 1, pp. 74-92, 2021.
- [16] C. J. Torrecilla-Salinas, J. Sedeño, M. J. Escalona, y M. Mejías, «Agile, Web Engineering and Capability Maturity Model Integration: A systematic literature review.», *Information and Software Technology*, vol. 71, pp. 92-107, mar. 2016, doi: 10.1016/j.infsof.2015.11.002.
- [17] L. de la C. Delgado Olivera y L. M. Díaz Alonso, «Modelos de Desarrollo de Software», *Revista Cubana de Ciencias Informáticas*, vol. 15, n.º 1, pp. 37-51, mar. 2021.
- [18] J. L. Cantú-Mata, F. Torres-Castillo, S. Alcaraz-Corona, y F. Banda-Muñoz, «Calidad, tiempo y costo en proyectos de desarrollo de software», *Interciencia*, vol. 43, n.º 10, pp. 707-710, 2018.
- [19] E. Martínez-Vidal y S. V. Hurtado-Gil, «Proceso de desarrollo de software con técnicas de creatividad, para las Pymes de la ciudad de Neiva», *Revista UIS Ingenierías*, vol. 18, n.º 1, pp. 99-109, 2018.
- [20] O. E. C. Cárdenas Villavicencio, M. P. Zea Ordóñez, M. R. Valarezo Pardo, y R. A. Ramón Ramón, «Comparativa de tendencias de desarrollo de software móvil», *3C TIC. Cuadernos de desarrollo aplicados a las TIC*, pp. 123-147, mar. 2021, doi: 10.17993/3c.
- [21] J. R. Molina Ríos y N. Pedreira Souto, «Swirl», *metodología para el diseño y desarrollo de aplicaciones web*. 3ciencias, 2019. Accedido: jul. 31, 2021. [En línea]. Disponible en: <https://dialnet.unirioja.es/servlet/libro?codigo=741383>
- [22] C. Asare y P. Lawer Angmor, «The Effect of Debt Financing on the Profitability of SMEs in Accra Metropolis.», *Journal of Arts and Social Sciences*, vol. 13, n.º 2, sep. 2020, Accedido: ago. 04, 2021. [En línea]. Disponible en: <https://doaj.org/article/065fc6dfba9a4846a2abe86f9f15387b>
- [23] L. Lorente, «Crecimiento, crédito e inflación*», *Revista de Economía Institucional*, vol. 21, n.º 40, pp. 9-68, 2019.
- [24] E. Acosta Gonzaga, J. A. Álvarez Cedillo, y A. Gordillo Mejía, «Arquitecturas en n-Capas: Un Sistema Adaptivo», *Polibits*, n.º 34, pp. 34-37, 2006.
- [25] A. D. Rodríguez Peña y L. G. Silva Rojas, «Arquitectura de software para el sistema de visualización médica Vismedic», *Revista Cubana de Informática Médica*, vol. 8, n.º 1, pp. 75-86, jun. 2016.
- [26] J. M. García, Á. E. Gil, y E. A. Sánchez, «Desarrollo de una arquitectura de software para el robot móvil Lázaro», *Ingeniare. Revista chilena de ingeniería*, vol. 26, n.º 3, pp. 376-390, 2018, doi: 10.4067/S0718-33052018000300376.
- [27] R. Padmanaban, M. Thirumaran, P. Anitha, y A. Moshika, «Computability evaluation of RESTful API using Primitive Recursive Function», *Journal of King Saud University - Computer and Information Sciences*, dic. 2018, doi: 10.1016/j.jksuci.2018.11.014.
- [28] Z. Mamani Rodríguez, L. Del Pino Rodríguez, y J. C. Gonzales Suarez, «Arquitectura basada en Microservicios y DevOps para una ingeniería de software continua», *Industrial Data*, vol. 23, n.º 2, 2020, Accedido: ago. 04, 2021. [En línea]. Disponible en: <https://www.redalyc.org/journal/816/81665362014/>

- [29] N. Comas Arias, B. Catalá González, y O. Oro Dosouto, «Prueba de bondad de ajuste para la distribución de distancias en secuencias de datos categóricos», *Revista Cubana de Ciencias Informáticas*, vol. 15, n.º 2, pp. 62-76, jun. 2021.
- [30] H. S. Krohn, «Programación de buscadores en JavaScript para diccionarios digitales**», *Cuadernos de Lingüística Hispánica*, n.º 34, pp. 109-130, 2019, doi: 10.19053/0121053X.n34.2019.9410.
- [31] L. A. Llerena Ocaña, G. E. Fernández Villacres, F. A. Viscaino Naranjo, y F. P. Baño Naranjo, «Frameworks basados en typescript para el desarrollo de aplicaciones web interactivas», *Dilemas contemporáneos: educación, política y valores*, vol. 8, n.º 3, ago. 2021, doi: 10.46377/dilemas.v8i3.2644.
- [32] S. M. Velásquez Restrepo, J. A. Londoño Gallego, J. D. Vahos, y C. López Romero, «Desarrollo de una plataforma web multimedial para la elaboración de proyectos bajo la metodología de marco lógico», *Lámpsakos*, n.º 18, pp. 12-20, 2017, doi: 10.21501/21454086.2601.
- [33] G. Contreras Mota, R. Lemuz López, y C. Bautista Ramos, «Pulse oximeter with Internet data visualization», *Sistemas & Telemática*, vol. 16, n.º 45, pp. 9-18, 2018, doi: <https://doi.org/10.18046/syt.v16i45.2746>.
- [34] K. Tserpes, M. Pateraki, y I. Varlamis, «Strand: scalable trilateration with Node.js», *J Cloud Comp*, vol. 8, n.º 1, p. 16, nov. 2019, doi: 10.1186/s13677-019-0142-y.
- [35] A. Mardan, «Using Express.js to Create Node.js Web Apps», *Practical Node.js: Building Real-World Scalable Web Apps*, vol. Practical Node.js, pp. 51-87, 2018, doi: https://doi.org/10.1007/978-1-4842-3039-8_2.
- [36] S. Bjeladinovic, Z. Marjanovic, y S. Babarogic, «A proposal of architecture for integration and uniform use of hybrid SQL/NoSQL database components», *Journal of Systems and Software*, vol. 168, p. 110633, oct. 2020, doi: 10.1016/j.jss.2020.110633.
- [37] F. J. Moreno Arboleda, J. E. Quintero Rendón, y R. Rueda Vásquez, «Una comparación de rendimiento entre Oracle y MongoDB», *Ciencia e Ingeniería Neogranadina*, vol. 26, n.º 1, abr. 2016, doi: 10.18359/rcin.1669.
- [38] J. R. M. Ríos *et al.*, «MMS», *Metodología para el Diseño y Desarrollo de Aplicaciones Móviles*. 3Ciencias, 2021.
- [39] J. K. Céleri Pacheco, C. X. Vega Oyola, y J. Maza Córdova, «Validación de una metodología de evaluación de calidad de sitios web: Caso de estudio UTMACH», *Cumbres*, vol. 3, n.º 1, pp. 167-174, 2017.
- [40] A. Alsaedi, «Comparing Web Accessibility Evaluation Tools and Evaluating the Accessibility of Webpages: Proposed Frameworks», *Information*, vol. 11, n.º 1, Art. n.º 1, ene. 2020, doi: 10.3390/info11010040.

ANEXOS

ANEXO A: ANÁLISIS DE RIESGOS

Tabla 27 Análisis de riesgos: RG-1

Riesgo No:		RG-1		Fecha:	
Descripción:		Poca disponibilidad del cliente para la elicitación de requerimientos			
Etapa	Categoría	Stakeholders afectados	Actividad	Probabilidad	Impacto
Análisis	Alcance	Cliente	El cliente no posee disponibilidad de tiempo para definir las funcionalidades del sistema..	40%	0.75
Análisis	Requerimientos	Cliente	Los requerimientos no son detallados en las reuniones respectivas	30%	0.80

Fuente: Elaboración propia

Tabla 28 Análisis de riesgos: RG-2

Riesgo No:		RG-2		Fecha:	
Descripción:		Complejidad de entendimiento del proceso de negocio			
Etapa	Categoría	Stakeholders afectados	Actividad	Probabilidad	Impacto
Análisis	Requerimientos	Cliente	Al identificar los requerimientos según las actividades de la empresa.	45%	0.5
Diseño	Funcionalidad	Cliente, Equipo de desarrollo	Al diseñar los botones, restricciones y funcionalidades de cada operación.	60%	0.75

Fuente: Elaboración propia

Tabla 29 Análisis de riesgos: RG-3

Riesgo No:		RG-3	Fecha:		
Descripción:		Diseño de interfaz de usuario difícil de entender			
Etapa	Categoría	Stakeholders afectados	Actividad	Probabilidad	Impacto
Diseño	Prototipado	Usuario final	El diseño del prototipo de la interfaz no contiene una estructura consecutiva.	50%	0.30
Diseño	Diseño de interfaces	Usuario final	Los botones o contenido no reflejan el proceso a ejecutar.	50%	0.50

Fuente: Elaboración propia

Tabla 30 Análisis de riesgos: RG-4

Riesgo No:		RG-4	Fecha:		
Descripción:		Diseño de base de datos compleja			
Etapa	Categoría	Stakeholders afectados	Actividad	Probabilidad	Impacto
Diseño	Modelado de base de datos	Equipo de desarrollo	Los campos y tablas de la base de datos no están diseñados adecuadamente	80%	0.8
Implementación	Desarrollo	Equipo de desarrollo	Las funciones del api-rest para la comunicación y almacenamiento de datos es compleja.	75%	0.8

Fuente: Elaboración propia

Tabla 31 Análisis de riesgos: RG-5

Riesgo No:		RG-5	Fecha:		
Descripción:		Poco conocimiento sobre la herramienta, metodología o tecnologías de desarrollo.			
Etapa	Categoría	Stakeholders afectados	Actividad	Probabilidad	Impacto
Análisis	Diseño	Equipo de desarrollo	Falta de conocimiento sobre elaboración de procesos o diagramas según la metodología de desarrollo.	30%	0.55
Implementación	Desarrollo	Equipo de desarrollo	Falta de conocimiento de las tecnologías de desarrollo, lenguaje o gestor de base de datos.	60%	0.8

Fuente: Elaboración propia

ANEXO B: DESGLOSE DE REQUERIMIENTOS

A continuación, se detallan cada uno de los requerimientos funcionales incluyendo sub-actividades, debido a que cada módulo contempla diversos procesos

Tabla 32: Especificación de requerimiento funcional: RF-01

Especificación de Requerimiento funcional.		
RF-01	Autenticación del usuario mediante el inicio de sesión (validación de usuario y contraseña).	
Categoría	Completo, prioritario	
Fecha	Responsable	Bárbara Romero
Observación	Inicio de sesión mediante el uso de usuario y contraseña. Es necesaria la validación.	
Actores	Usu-Administrador, Usu-Final	
Post-condición	Al ingresar un usuario o contraseña erróneo se visualiza un mensaje de error. Al ser validado se presenta la página principal con las opciones a las que puede acceder según sus permisos.	

Fuente: Elaboración propia.

Tabla 33: Especificación de requerimiento funcional: RF-02

Especificación de Requerimiento funcional.		
RF-02	Gestión de los usuarios, ingreso, asignación de permisos, modificación, y visualización.	
Categoría	Completo, prioritario	
Fecha	Responsable	Bárbara Romero
Observación	RF-02.1: Ingresar un nuevo usuario de tipo administrador o empleado). RF-02.2: Asignar permisos a los usuarios de tipo administrador y empleados. RF-02.3: Modificar los datos personales y permisos asignados al usuario. RF-02.4: Visualizar los datos personales, rol, y permisos de cada usuario solo como usuario tipo administrador. RF-02.5: Visualizar el perfil del usuario (solo información personal) como usuario tipo empleado. RF-02.6: Modificar la contraseña. RF-02.7: Inhabilitar y habilitar a un usuario (dar de baja) solo como administrador.	
Actores	Usu-Administrador	
Post-condición	Las opciones del módulo se visualizarán según los permisos otorgados.	

Fuente: Elaboración propia.

Tabla 34: Especificación de requerimiento funcional: RF-03

Especificación de Requerimiento funcional.		
RF-03	Gestión de clientes (ingresar, modificar, visualizar).	
Categoría	Completo, prioritario	
Fecha		Responsable Bárbara Romero
Observación	RF-03.1: Registrar un nuevo cliente. RF-03.2: Modificar la información de clientes. RF-03.3: Eliminar un cliente que no posea contratos o créditos vinculados.	
Actores	Usu-Administrador, Usu-Final	
Post-condición	Las opciones del módulo se visualizarán según los permisos otorgados.	

Fuente: Elaboración propia.

Tabla 35: Especificación de requerimiento funcional: RF-04

Especificación de Requerimiento funcional.		
RF-04	Gestión de roles y permisos del sistema.	
Categoría	Completo, prioritario	
Fecha		Responsable Bárbara Romero
Observación	RF-04.1: Crear, modificar y eliminar un rol. RF-04.2: Asignar permisos por defecto a roles. RF-04.3: Remover permisos por defecto a roles. RF-04.4: Visualizar los permisos por defecto de cada rol.	
Actores	Usu-Administrador	
Post-condición	Las opciones del módulo se visualizarán según los permisos otorgados.	

Fuente: Elaboración propia.

Tabla 36: Especificación de requerimiento funcional: RF-05

Especificación de Requerimiento funcional.		
RF-05	Gestión de crédito.	
Categoría	Completo, prioritario	
Fecha		Responsable Bárbara Romero
Observación	RF-05.1: Crear y visualizar créditos. RF-05.2: Modificar crédito solo como usuario administrador. RF-05.3: Visualizar e Imprimir crédito. RF-05.4: Visualizar el monto de deuda y monto de crédito a pagar en el mes de un cliente.	
Actores	Usu-Administrador, Usu-Final	
Post-condición	Las opciones del módulo se visualizarán según los permisos otorgados.	

Fuente: Elaboración propia.

Tabla 37: Especificación de requerimiento funcional: RF-06

Especificación de Requerimiento funcional.		
RF-06	Gestión de contratos.	
Categoría	Completo, prioritaria	
Fecha		Responsable Bárbara Romero
Observación	RF-06.1: Vincular contrato a crédito. RF-06.2: Guardar o Imprimir contrato. RF-06.3: Modificar contrato solo como usuario administrador. RF-06.4: Reasignación de contrato. RF-06.5: Visualización de contenido de contratos. RF-06.6: Listar contratos.	
Actores	Usu-Administrador, Usu-Final	
Post-condición	Las opciones del módulo se visualizarán según los permisos otorgados.	

Fuente: Elaboración propia.

Tabla 38: Especificación de requerimiento funcional: RF-07

Especificación de Requerimiento funcional.		
RF-07	Gestión de pagos.	
Categoría	Completo, prioritaria	
Fecha		Responsable Bárbara Romero
Observación	RF-07.1: Ingresar nuevo pago de cuota de un cliente. RF-07.2: Pagar totalmente una deuda. RF-07.3: Anular pago de cuota solo como usuario administrador.	
Actores	Usu-Administrador, Usu-Final	
Post-condición	Las opciones del módulo se visualizarán según los permisos otorgados.	

Fuente: Elaboración propia.

Tabla 39: Especificación de requerimiento funcional: RF-08

Especificación de Requerimiento funcional.		
RF-08	Visualización de un panel administrativo para la gestión de cartera de los clientes (visualización de cuentas en mora, cuentas al día, etc).	
Categoría	Completo, prioritaria	
Fecha		Responsable Bárbara Romero
Observación	Visualización de datos para toma de decisiones mediante un panel de control.	
Actores	Usu-Administrador	
Post-condición	Las opciones del módulo se visualizarán según los permisos otorgados.	

Fuente: Elaboración propia.

Tabla 40: Especificación de requerimiento funcional: RF-10

Especificación de Requerimiento funcional.			
RF-09	Configurar los valores por defecto de la empresa.		
Categoría	Completo, prioritaria		
Fecha		Responsable	Bárbara Romero
Observación	Gestionar los campos independientes como estado civil, entidades bancarias, formas de pago, etc.		
Actores	Usu-Administrador		
Post-condición	Las opciones del módulo se visualizarán según los permisos otorgados.		

Fuente: Elaboración propia.

ANEXO C: ESTRUCTURA DE DESGLOSE DE TRABAJO

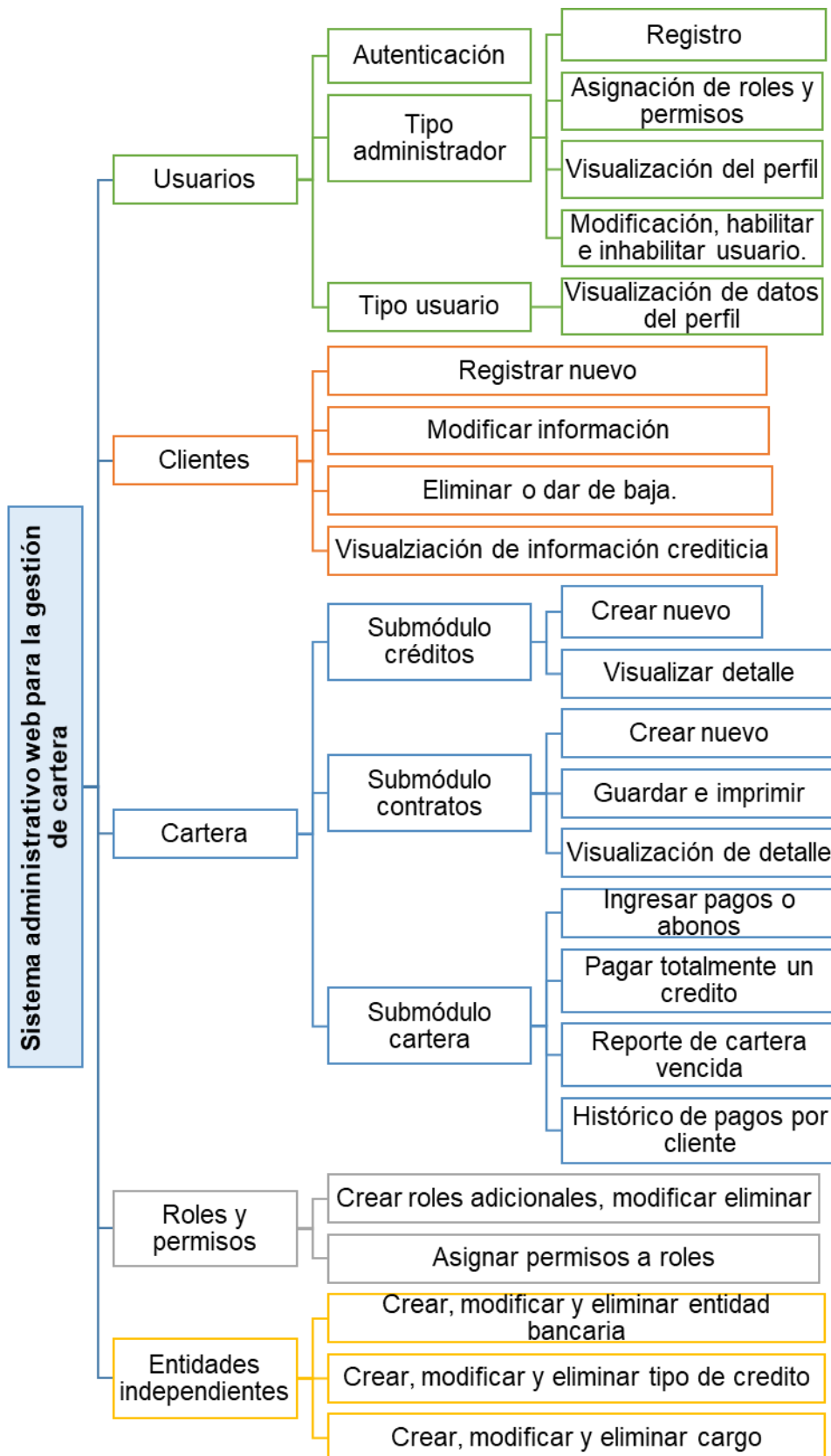


Figura 62 Estructura de desglose de trabajo

Fuente: Elaboración propia

ANEXO D: MODELADO DE BASE DE DATOS

A continuación, se detallan los esquemas principales dentro de la base de datos.

Esquema de Usuarios.

```
{
  id_usuario: {
    type: String, required: true,
    trim: true, unique: true,
    minLength: 10, maxLength: 10
  },
  nick_usuario: {
    type: String, required: true,
  },
  nombres: {
    type: String, required: true,
  },
  apellidos: {
    type: String, required: true,
  },
  email_usuario: {
    type: String, unique: true
  },
  telefono_usuario: String,
  password_usu: {
    type: String, required: true
  },
  cargo: {
    type: String, required: true
  },
  estado: {
    type: String, required: true,
  },
  estricto: {
    type: Boolean, required: true,
  },
  roles: [{
    type: mongoose.Schema.Types.ObjectId,
    ref: 'Roles', required: true
  }],
  permisos: [{
    type: mongoose.Schema.Types.ObjectId,
    ref: 'Permisos', required: true
  }],
}
```

Figura 63 Esquema base de datos: Usuarios

Fuente: Elaboración propia

Esquema de Clientes.

```
{
  _id: {
    type: String,
    required: true, trim: true,
    minLength: 10, maxLength: 10
  },
  nombre_cli: { type: String, required: true, },
  apellido_cli: { type: String, required: true, },
  direccion_cli: { type: String, required: true, },
  ciudad_cli: { type: String, required: true, },
  correo_cli: { type: String, trim: true },
  telefonos: {
    telefono1_cli: {
      type: String,
      required: true, trim: true
    },
    telefono2_cli: {
      type: String, trim: true
    },
  },
  estado_civil: String,
  conyuge: {
    cedula: {
      type: String, minlength: 10,
      maxlength: 10, trim: true
    },
    estado_laboral: String,
    nombre: String,
    apellido: String,
    correo: String,
    telefono: String
  },
  estado_laboral: String,
  empresa: {
    nombre_emp: String,
    cargo_emp: String,
    tipo_emp: String,
    direccion_emp: String,
    telefono_emp: String,
    descripcion_emp: String
  },
  creditos: [{
    type: String,
    ref: 'Contratos',
  }]
}
```

Figura 64 Esquema base de datos: Clientes

Fuente: Elaboración propia

Esquema de Contratos.

```
{
  _id: {
    type: String, required: true,
    trim: true, unique: true
  },
  codigo_lote: String,
  cliente: {
    type: String,
    ref: 'Clientes',
  },
  estado: {
    type: String, required: true,
    default: 'Vigente'
  },
  tipo_credito: {
    type: String,
    required: true
  },
  fecha_creacion: { type: Date, required: true },
  mto_total: Number,
  informacion_adicional: String,
  contrato_pdf: Buffer,
  beneficiarios_fallecimiento: [{
    _id: {
      type: String, minlength: 10, maxlength: 10, trim: true
    },
    nombre: String,
    apellido: String,
    correo: String,
    telefono: String
  }],
  credito: { type: String, ref: 'Creditos', required: true },
  entrada: { type: String, ref: 'Entradas', required: true },
}
```

Figura 65 Esquema base de datos: Contratos

Fuente: Elaboración propia

Esquema de Créditos

```
{
  _id: {
    type: String, required: true, trim: true, unique: true
  },
  cliente: {
    type: String, ref: 'Clientes', required: true
  },
  contrato: {
    type: String, required: true, unique: true
  },
  num_cuotas: {
    type: Number, required: true
  },
  tasa_interes: {
    type: mongoose.Schema.Types.Double, required: true
  },
  mto_credito: {
    type: mongoose.Schema.Types.Double, required: true
  },
  mto_interes: {
    type: mongoose.Schema.Types.Double, required: true
  },
  valor_dividendo: { type: mongoose.Schema.Types.Double },
  saldo: {
    type: mongoose.Schema.Types.Double, required: true, default: 0
  },
  estado: { type: String, required: true, default: 'VIGENTE' },
  fecha_inicio: { type: Date, required: true },
  fecha_fin: { type: Date, required: true },
  pagos: [{
    codigo: { type: Schema.ObjectId, ref: 'Pagos' },
    fechaPago: Date,
    fechaVenc: Date,
    estado: String,
    monto: mongoose.Schema.Types.Double,
    num_periodo: Number,
  }]
}
```

Figura 66 Esquema base de datos: Creditos

Fuente: Elaboración propia

Esquema de Entradas

```
{
  _id: {
    type: String, required: true, trim: true, unique: true
  },
  cliente: { type: String, ref: 'Clientes', required: true },
  contrato: { type: String, required: true, unique: true },
  valor_entregado: { type: Number },
  num_cuotas: { type: Number, required: true },
  tasa_interes: {
    type: mongoose.Schema.Types.Double, required: true
  },
  mto_credito: {
    type: mongoose.Schema.Types.Double, required: true
  },
  mto_interes: {
    type: mongoose.Schema.Types.Double, required: true
  },
  valor_dividendo: { type: mongoose.Schema.Types.Double },
  saldo: {
    type: mongoose.Schema.Types.Double, required: true, default: 0
  },
  estado: { type: String, required: true, default: 'VIGENTE' },
  fecha_inicio: { type: Date, required: true },
  fecha_fin: { type: Date, required: true },
  pagos: [{
    codigo: { type: Schema.ObjectId, ref: 'Pagos' },
    fechaPago: Date,
    fechaVenc: Date,
    estado: String,
    monto: mongoose.Schema.Types.Double,
    num_periodo: Number,
  }]
}
```

Figura 67 Esquema base de datos: Entradas

Fuente: Elaboración propia

Esquema de Pagos

```
{
  codigo: {
    comprobanteIng: { type: String, required: true, trim: true },
    codigo1: String,
    codigo2: String,
    codigo3: String,
  },
  concepto: String,
  contrato: {type: String, ref: 'Contratos'},
  cliente: {type: String, ref: 'Clientes'},
  num_periodo: { type: Number },
  met_pago: { type: String },
  ent_bancaria: { type: String },
  nro_ReciboBanco: String,
  fecha_venc: { type: Date },
  mto_dividendo: { type: mongoose.Schema.Types.Double },
  mto_interes: {
    type: mongoose.Schema.Types.Double
  },
  mto_capital: {
    type: mongoose.Schema.Types.Double
  },
  saldo_capital: {
    type: mongoose.Schema.Types.Double
  },
  interes_mora: { type: mongoose.Schema.Types.Double },
  valor_interes_mora: { type: mongoose.Schema.Types.Double },
  mto_cancelado: { type: mongoose.Schema.Types.Double },
  fecha_pago: { type: Date },
  abonos: [{
    mto_cancelado: { type: mongoose.Schema.Types.Double },
    fecha_pago: { type: Date, default: Date.now },
    met_pago: { type: String },
    ent_bancaria: { type: String },
    nro_ReciboBanco: String
  }],
  estado: { type: String, default: 'VIGENTE' },
  usuario: {
    type: mongoose.Schema.Types.ObjectId,
    ref: 'Usuarios',
  }
}
```

Figura 68 Esquema base de datos: Pagos

Fuente: Elaboración propia

Esquema de Notificaciones

```
{
  cliente: {
    type: String, ref: 'Clientes', required: true
  },
  fecha_contacto: {
    type: Date, required: true
  },
  contacto:{
    type: String, required: true
  },
  gestion:{
    type: String, required: true
  },
  fecha_futura:{
    type: Date, required: true
  }
}
```

Figura 69 Esquema base de datos: Notificaciones

Fuente: Elaboración propia

Esquema de Roles

```
{
  nombre: { type: String, required: true, unique: true },
  descripcion: String,
  estricto: { type: Boolean, required: true },
  permisosDefecto: [{ type: Schema.ObjectId, ref: 'Permisos' }]
}
```

Figura 70 Esquema base de datos: Roles

Fuente: Elaboración propia

Esquema de Permisos

```
{
  nombre: { type: String, required: true, unique: true },
  codigo: {type: String, require: true, unique: true}
}
```

Figura 71 Esquema base de datos: Permisos

Fuente: Elaboración propia

ANEXO E: DIAGRAMAS DE CASO DE USO

Cada módulo engloba una serie de funciones las cuales se detallan a continuación. Dentro de los mismos se hará referencia al término CRUD a las acciones de (Crear, Leer, Actualizar, y Eliminar).

Dentro de la gestión de usuarios se emplean actividades CRUD las cuales las realiza el usuario de tipo administrador. Al crear o modificar un usuario es posible asignar roles y permisos al usuario.

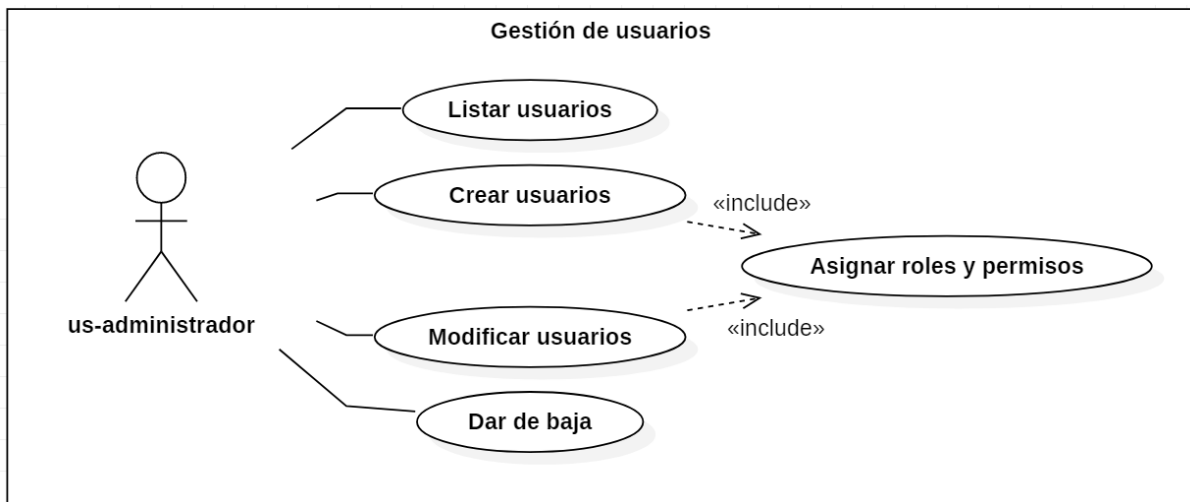


Figura 72 Diagrama de caso de uso: Gestión de usuarios

Fuente: Elaboración propia

Dentro de la gestión de clientes se realizan las actividades inmersas en un CRUD, además de visualizar los detalles de un cliente en específico, incluyendo los contratos que se le ha otorgado.

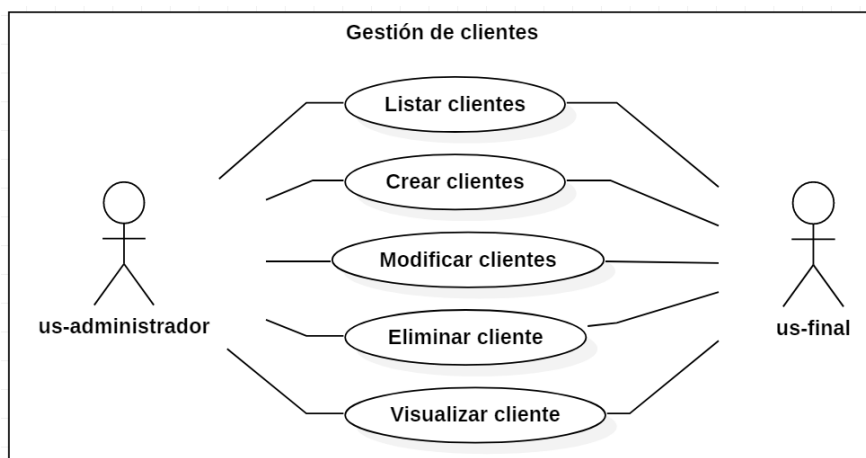


Figura 73 Diagrama de caso de uso: Gestión de clientes

Fuente: Elaboración propia

Dentro de la gestión de roles y permisos intervienen los usuarios de tipo administrador. El administrador tiene permisos para realizar las actividades CRUD en los roles, pero solo listar los permisos.

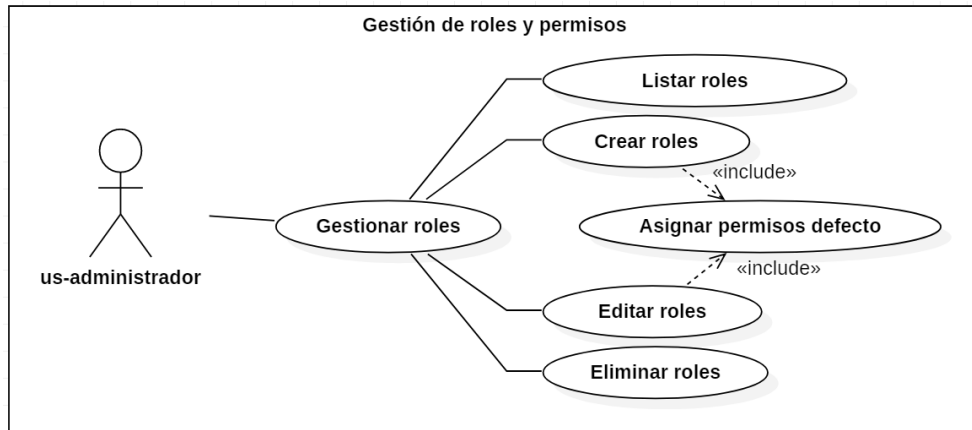


Figura 74 Diagrama de caso de uso: Gestión de roles y permisos
Fuente: Elaboración propia

El usuario de tipo administrador es capaz de realizar las actividades del CRUD del estado civil, el estado laboral, y los cargos. Estos campos son los requeridos dentro de la configuración del sistema.

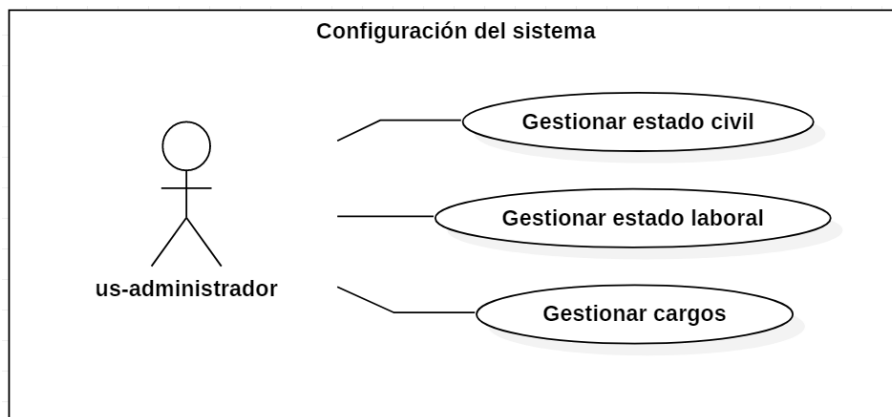


Figura 75 Diagrama de caso de uso: Configuración del sistema
Fuente: Elaboración propia

El usuario de tipo administrador es capaz de realizar las actividades del CRUD de las entidades bancarias, métodos de pago, y tipos de créditos. Estos campos son los requeridos dentro de la configuración de cartera.

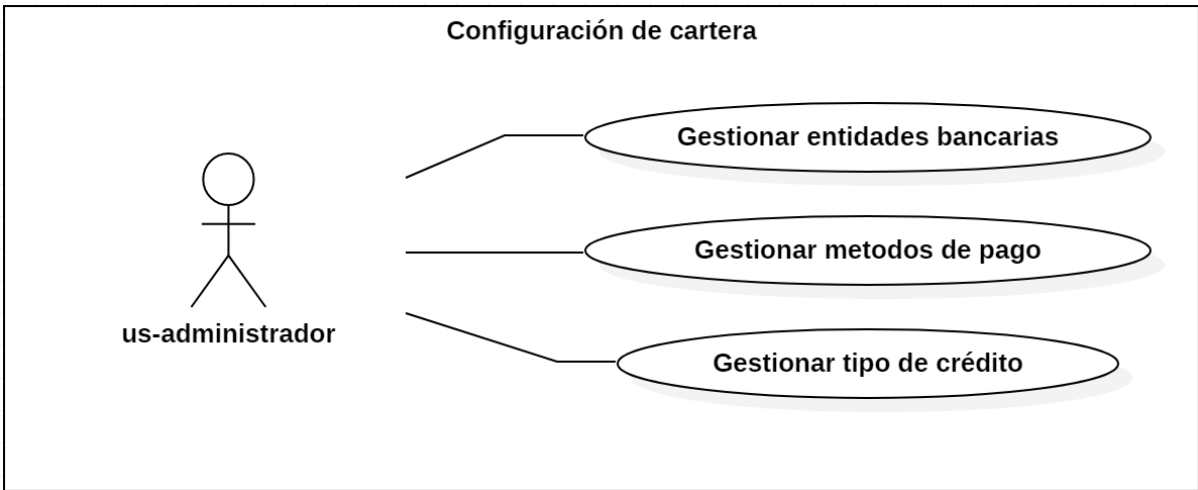


Figura 76 Diagrama de caso de uso: Configuración de cartera
Fuente: Elaboración propia

Dentro de la gestión de pagos se realizan las actividades de ingreso de pago, impresión del comprobante, listado de pagos diarios y los reportes históricos de pago por clientes. Dichas actividades pueden ser accedidas por el usuario tipo administrador y por el usuario final siempre y cuando posea los permisos necesarios.

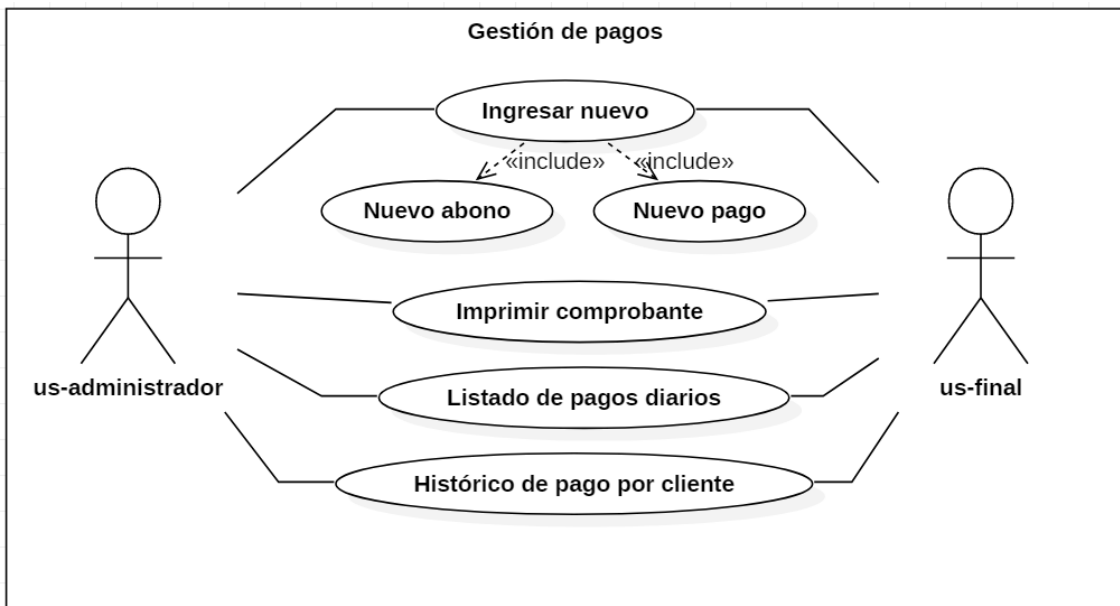


Figura 77 Diagrama de caso de uso: Gestión de pagos
Fuente: Elaboración propia

Dentro de la gestión de cartera solo se involucran los listados de carteras vencidas, de forma general y por cliente, incluyendo en la ultima las notificaciones realizadas a cada uno en caso de existir cuentas vencidas.

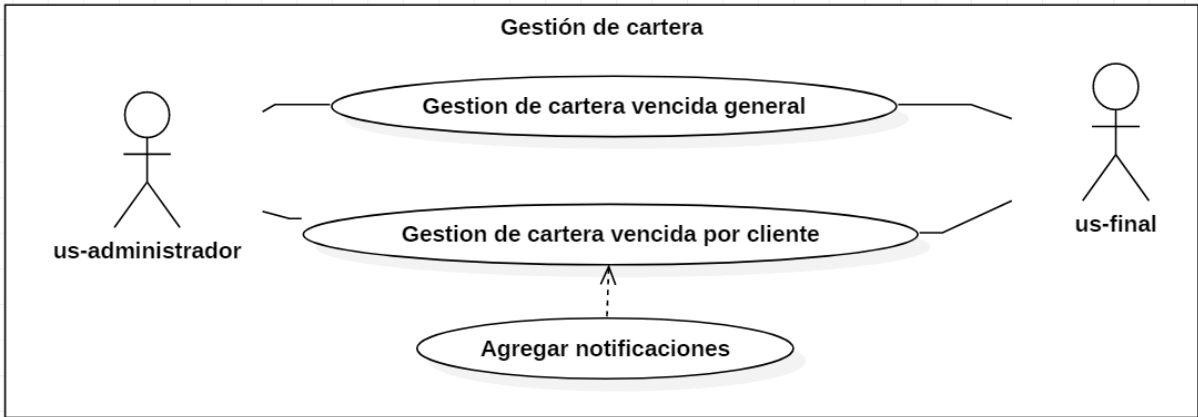


Figura 78 Diagrama de caso de uso: Gestión de cartera
Fuente: Elaboración propia