



**UNIVERSIDAD NACIONAL DE INGENIERÍA
DIRECCION DE AREA DE CONOCIMIENTO DE
TECNOLOGIA DE LA INFORMACION Y COMUNICACIÓN**

Trabajo Monográfico para optar al título de

INGENIERO EN COMPUTACION

Tema:

Sistemas web para la planificación de actividad y monitoreo de los gestores de crédito usando una aplicación móvil de recolección de información para las Microfinancieras de Nicaragua.

Autores:

Br. Javier Enmanuel Matus Widdy
Br. Darwing Antonio Ruiz Herrera

Tutor:

MSc. Gabriel Rafael Lacayo Saballos

febrero de 2024

Managua, Nicaragua

Dedicatorias

Dedico principalmente este trabajo monográfico a Dios, por haberme permitido llegar hasta este punto; por haberme dado salud, y darme lo necesario para seguir adelante día a día para lograr mis objetivos.

A mi madre por haberme apoyado en todo momento; por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.

A mi padre por ser una fuente constante de motivación y aliento pese a las circunstancias, por su perseverancia que ha sido un ejemplo para alcanzar mis metas.

Este proyecto lo dedico a todos los maestros que influenciaron en mi carrera. Especialmente a mi maestro Gabriel Rafael Lacayo Saballos, mi tutor, por su atención, tiempo, comprensión, dedicación durante todo el proyecto y por haberme transmitido sus conocimientos y a mi maestra Lizette Carolina Duarte Mora sin ellos no hubiese sido posible esta tesis.

Javier Enmanuel Matus Widdy

Con el corazón lleno de gratitud, dedico esta monografía a Dios, quien me ha guiado y sostenido durante todo este proceso. Gracias a su infinita sabiduría y amor, he podido superar los obstáculos y alcanzar este logro.

A mi familia, amigos y seres queridos, quienes han sido instrumentos de Dios en mi vida. Su apoyo incondicional y sus oraciones han sido fundamentales para mantener mi motivación y perseverancia en momentos de dificultad.

Además, doy gracias a mi tutor Gabriel Rafael Lacayo Saballos quien me brindó su valioso tiempo y conocimiento. Sus sugerencias y comentarios constructivos fueron una guía en el camino hacia la excelencia.

Darwing Antonio Ruiz Herrera

Agradecimientos

Agradecemos primeramente a Dios, por habernos dado la capacidad y la oportunidad de culminar esta etapa de nuestros estudios superiores con la entrega de este trabajo monográfico; por brindarnos salud, sabiduría, conocimientos y perseverancia.

A nuestras familias que son el pilar de nuestra vida y quienes nos han apoyado en cada paso que hemos dado y a quienes agradecemos por estar a nuestro lado en la finalización de este trabajo monográfico.

A la Universidad Nacional de Ingeniería por acogernos y brindarnos una educación de excelencia, formando nuestras vidas en el ámbito profesional y personal para consolidarnos como mejores personas para la sociedad a través de nuestra profesión.

A nuestro tutor MSc. Gabriel Rafael Lacayo Saballos por tomarse el tiempo de ayudarnos y asesorarnos durante la elaboración de este proyecto.

Y a cada uno de los docentes y amigos que contribuyeron en nuestra formación académica y personal en esta Alma Mater Universidad Nacional de Ingeniería.

Resumen del tema

El presente trabajo monográfico fue elaborado con el objetivo de facilitar y centralizar los procesos internos llevados a cabo por las microfinancieras, de igual manera brindar una herramienta para respaldar de forma segura la información que es manejada y es de beneficio para el personal de la entidad.

Previo al desarrollo del sistema de este proyecto monográfico la microfinanciera **Oportucredit S.A** carecía de un sistema especializado para el control de contribuyentes o la gestión de cobros, por lo cual guardaban parte de la información que utilizan en cuadernos o en Microsoft Excel, como consecuencia les era difícil llevar a cabo ciertos procesos tales como el control de pago de cuotas por cuentas, búsqueda relacionada con información de un contribuyente, entre otros.

Teniendo en cuenta los problemas que presentaba la microfinanciera en cuanto al manejo de la información, se realizó el levantamiento de los requerimientos, de igual manera se llevó a cabo el estudio de factibilidad para saber la viabilidad del proyecto. Haciendo uso de la metodología Secuencial Cascada (Waterfall Model) para llevar a cabo el desarrollo del sistema, se llevaron a cabo tareas de:

- Captura de información: como entrevistas, y observación del flujo de actividades de los involucrados.
- Análisis: para obtener requerimientos funcionales y requisitos no funcionales.
- Codificación: diseñar el sistema en base a los requerimientos obtenidos.

Se llevaron a cabo exhaustivas pruebas funcionales y de seguridad, basadas en los requisitos modelados en los casos de uso, asegurando así el funcionamiento adecuado del sistema y su interacción confiable con el usuario, resultado de un trabajo riguroso y detallado. A través de cada sección en este documento, se sustenta de forma teórica tanto la metodología, como las herramientas de análisis para cada uno de los módulos con los que el sistema de información cuenta.

Índice

1. Introducción	1
2. Antecedentes	2
3. Justificación	4
4. Objetivos	5
4.1 Objetivo General	5
4.2 Objetivos Específicos	5
5. Marco Teórico	6
5.1. Conceptos Generales.....	6
5.1.1 Microfinancieras.....	6
5.1.2 Planificación de Actividades	6
5.1.3 Aplicaciones Web	7
5.1.4 Desarrollo de Aplicaciones Web	8
5.1.5 Aplicaciones Móviles.....	9
5.1.6 Desarrollo de Aplicaciones Móviles	9
5.2 Metodología de desarrollo	11
5.2.1 Metodología Cascada	11
5.2.2 Fases de la Metodología Cascada.....	12
5.3 Herramientas de desarrollo	13
5.3.1 UML (Lenguaje Unificado de Modelado).....	13
5.3.2 Programación Orientada a Objetos (POO)	14
5.3.3 JAVA.....	14
5.3.4 Android Studio	15
5.3.5 PHP	15
5.3.6 Visual Studio Code	16
5.3.7 HTML 5	16
5.3.8 CSS 3	17
5.3.9 JavaScript	18
5.3.10 JQuery	19
5.3.11 MVC.....	19
5.3.12 MySQL.....	20
6. Diseño Metodológico	21

6.1	Análisis de Requerimientos	21
6.2	Diseño de solución	22
6.3	Implementación	25
6.4	Verificación.....	25
6.5	Mantenimiento.....	25
7.	Estudio de Factibilidad	26
7.1	Factibilidad técnica.....	26
7.1.1	Recursos técnicos.....	26
7.2	Factibilidad Operativa.....	27
7.3	Factibilidad Económica.....	28
7.3.1	Costo de Adquisición de Hardware.....	28
7.3.2	Costo de Adquisición de Software	28
7.3.3	Costo Recurrente.....	29
7.3.4	Costo Equipo de Desarrollo (Recursos Humanos)	29
7.4	Factibilidad Legal	30
7.4.1	Requerimientos Legales	30
7.4.2	Licencia para el Software.....	30
7.4.3	Contratos de Servicios.....	30
7.4.4	Beneficios de Automatizar	30
8.	Análisis y presentación de resultados	32
8.1	Fase 1: Análisis de requerimientos	32
8.1.1	Análisis de Requerimientos	33
8.1.2	Definición de Actores.....	35
8.1.3	Diagrama de Casos de Usos	37
8.2	Fase 2: Diseño de la solución	52
8.2.1	Diagrama de Actividades	52
8.2.2	Diagrama de Estados	57
8.2.3	Modelo de Navegación	60
8.2.4	Modelo Relacional	62
8.2.5	Diseño Arquitectónico	65
8.2.6	Diseño de interfaces gráficas de usuario (GUI)	66
8.3	Fase 3: Codificación del Software	73

8.4	Fase 4: Pruebas	75
8.4.1	Pruebas Funcionales del Sistema.....	76
8.4.2	Pruebas de Seguridad del Sistema.....	78
8.4.3	Pruebas Unitarias del Sistema.....	80
9.	Conclusiones	82
10.	Recomendaciones	83
11.	Referencias	84
12.	Anexos	86
12.1	Anexo 1: Contrato de Prestaciones de Servicios	86
12.1.1	Acerca de las partes	86
12.1.2	Clausulas.....	86
12.2	Anexo 2: Capturas del Sistema	89
12.2.1	Sistema web	89
12.2.2	Aplicación Móvil.....	91
12.3	Anexo 4: Entrevista	92

Índice de Figuras

Figura 1. Tomado de la página web: ionos.es - Diagrama Cascada-----	11
Figura 2. Tomado de la página web: Mi Blog Técnico - Diagrama MVC-----	20
Figura 3. Jerarquía de Usuarios.-----	36
Figura 4 Caso de uso administrador-----	40
Figura 5 Caso de uso administrador Continuación-----	41
Figura 6. Casos de uso Supervisor.-----	42
Figura 7. Diagrama de actividades Iniciar Sesión.-----	52
Figura 8. Diagrama de actividades General - Supervisor.-----	53
Figura 9. Diagrama de actividades Monitoreo - Geo Rutas - Supervisor.-----	54
Figura 10. Diagrama de actividades Gestión de Acceso - Administración de Usuario - Administrador.-----	55
Figura 11. Diagrama de actividades Configuración - Creación de Formularios - Administrador.-----	56
Figura 12. Diagrama de estados - Formularios.-----	57
Figura 13. Diagrama de estados - Rutas.-----	58
Figura 14. Diagrama de estados - Sucursales.-----	59
Figura 15. Modelo de navegación – Administrador Sistema Web.-----	61
Figura 16. Modelo de navegación – Supervisor Sistema web.-----	62
Figura 17. Modelo Relacional.-----	63
Figura 18 Modelo Relacional-----	64
Figura 19. Diseño arquitectónico del sistema.-----	66
Figura 20. Estructura de la interfaz de usuario Web.-----	68
Figura 21. Vista Dashboard. Realizada con herramienta: NinjaMock-----	71
Figura 22. Vista Lista de Formularios. Realizada con herramienta: NinjaMock -	72
Figura 23. Vista Menú.-----	73
Figura 24. Vista Agregar Formulario.-----	74
Figura 25. Controlador del Formulario.-----	74
Figura 26. Modelo del Formulario.-----	75
Figura 27. Código de FormControllertest.-----	80
Figura 28. Resultado de FormControllertest.-----	80
Figura 29. Código SucursalControllerTest.-----	81
Figura 30. Resultado de SucursalControllerTest.-----	81

Índice de Tablas

Tabla 1. Especificaciones de Hardware. Fuente: Equipo en posesión -----	26
Tabla 2. Especificaciones de Software. Fuente: Equipo en posesión -----	27
Tabla 3. Costo de Adquisición de Software. -----	28
Tabla 4. Costo Recurrente. -----	29
Tabla 5. Costo del Equipo de Desarrollo. -----	29
Tabla 6. Requerimientos Funcionales del sistema. -----	34
Tabla 7. Requerimientos no Funcionales del sistema. -----	35
Tabla 8. Funciones de los Actores. Fuente: Entrevista -----	37
Tabla 9. Obtenida de la Pagina web ingeniería de Software Plantilla de casos de uso. -----	38
Tabla 10. Niveles de prioridad -----	39
Tabla 11. Frecuencia. -----	39
Tabla 12. Plantilla de casos de uso - Crear Ruta. -----	43
Tabla 13. Plantilla de casos de uso - Visualizar Ruta. -----	44
Tabla 14. Plantilla de casos de uso - Listar Usuarios. -----	45
Tabla 15. Plantilla de casos de uso - Listar Trayectoria de Usuario por fecha. -	46
Tabla 16. Plantilla de casos de uso - Listar Actividades de Usuario. -----	47
Tabla 17. Plantilla de casos de uso - Registrar Cliente. -----	48
Tabla 18. Plantilla de casos de uso - Editar Cliente. -----	49
Tabla 19. Plantilla de casos de uso - Visita de Clientes. -----	50
Tabla 20. Plantilla de casos de uso - Reasignar Actividades de Usuario. -----	51
Tabla 21. Pruebas Funcionales - inserción de un Nuevo Formulario. -----	76
Tabla 22. Pruebas Funcionales - Obtención de todos los Formularios. -----	77
Tabla 23. Pruebas de Seguridad - Iniciar Sesión. -----	78
Tabla 24. Pruebas de Seguridad - Ingreso no Autorizado. -----	79

1. Introducción

En este documento se describe el desarrollo de un sistema web para centralizar el control de las actividades de los colaboradores¹ y la gestión de cobros de la microfinanciera centralizar el control de actividades de los colaboradores y la gestión de cobros , con el fin de brindar un medio que facilite el procesamiento de la información de los registros para un correcto ordenamiento, planificación de jornadas y seguimiento de los ingresos financieros procedentes de los clientes; y de esta manera, apoyar en la agilización de las funciones y la toma de decisiones que se realizan actualmente en el negocio. Así mismo, se desarrolló una aplicación móvil en paralelo que permitiría recibir notificaciones de las tareas asignadas mediante el sistema web. Esta aplicación informa la ubicación en tiempo real del colaborador de crédito y habilitaba la carga de imágenes de manera eficiente. Se busca mejorar la eficiencia del negocio, mediante la automatización de los procesos de control y gestión de cobros, así mediante la implementación de una herramienta que facilite la comunicación y seguimiento del trabajo de los colaboradores de manera remota. (Laudon & Laudon, 2012)

Para el desarrollo del sistema se optó por la por la metodología de desarrollo Cascada (Waterfall Model) la cual se define como una secuencia de pasos. El principal objetivo del enfoque cascada es proporcionar un lenguaje de modelado específico en este caso basado en UML, una metodología dirigida por fases y herramientas de soporte tanto para el diseño sistemático como para la generación semiautomática de aplicaciones. (IONOS, 2019)

Una de las metas de GICAPP² es que este sistema se adapte a los procesos de cualquier microfinanciera del país, para ello se trabajará con una empresa que es socia, que permitirá conocer los procesos que se plantean automatizar.

¹ Personal de la Microfinanciera.

² El nombre de la empresa no es un acrónimo y está conformado por las iniciales de los fundadores.

2. Antecedentes

La empresa GICAPP especializada en desarrollo de software comenzó sus labores hace 7 años aproximadamente orientada a cubrir las necesidades de los clientes, siendo estos principalmente las microfinancieras.

Para la gerencia de las microfinancieras es de vital importancia conocer los lugares y clientes que el equipo de gestores de crédito a visitado, ya que en muchas ocasiones dos gestores han estado en el mismo sitio o reportan un lugar como visitado, haciendo de la calendarización un proceso con muchas deficiencias debido a la mala administración de los datos de visitas y ubicación de los gestores de crédito.

Entre las tareas que se calendarizan están; visitas de campo para elaboración de inventario y la captación de nuevos clientes mediante entrevistas principalmente en los mercados del país.

La microfinanciera **Oportucredit S.A** carecía de un sistema de información orientado a colaboradores, actualmente el manejo se realiza en anotaciones en cuadernos para el control de los colaboradores, luego esta información semanalmente es transcrita digitalmente en Microsoft Excel esto con el fin de llevar un seguimiento de los registros para la gestión de cobro.

A continuación, se presenta un resumen de la consulta documental de proyectos realizados que guardan relación con el tema propuesto.

Mobile Suitcase

Sistema Móvil que registra actividades, y posee geolocalización esta fue creada por Simplified Technologies una empresa de origen colombiano, y administrada en Nicaragua por parte de Claro Nicaragua. (Simplified., n.d.)

Working Day Suite

Sistema Web de Monitoreo y planificación de actividades, esta fue desarrollada por Internalia Group una empresa de desarrollo española, representada en Nicaragua por Tigo Nicaragua. (Group, I., 2016)

En conclusión, la propuesta planteada en este documento se ajusta directamente a suplir las necesidades de empresas microfinancieras que lleven a cabo su proceso de monitoreo y asignación de actividades de manera manual, o por el uso de un sistema web que no se adapta completamente a las necesidades de ellas.

Tener el control total sobre la ubicación de sus gestores y / o clientes, tener en cuenta las acciones que realice el usuario para evitar el monitoreo son algunas de las características que se han integrado, que las aplicaciones antes mencionadas no poseen, por mencionar algunas.

Una de las características que las aplicaciones antes mencionadas no poseen y es de suma importancia en un rubro donde tener la mayor parte del control sobre lo que realiza tu personal es saber, cuando se desactivan las funcionalidades de datos móviles y GPS, nuestra solución es poder enviar un correo electrónico al personal correspondiente.

3. Justificación

En los últimos años el control de las diferentes actividades y monitoreo de los gestores de crédito en las microfinancieras han adquirido sistemas especializados para sus procesos que en ocasiones sufren de pérdida de información valiosa, estos procesos requieren una calendarización por parte del personal administrativo; así como seguimiento de los colaboradores que realizan sus tareas fuera de las microfinancieras.

Este sistema permitirá realizar una planificación inmediata y eficiente de sus actividades a través de un módulo de planificación y mantener un seguimiento en tiempo real del personal que se encuentra en labores fuera de la compañía mediante un módulo de monitoreo y seguimiento, el módulo de reporte brindará la información obtenida.

Además de la seguridad de la información existen otros beneficios de la adquisición de un sistema especializado tales como mejora en la eficiencia de los procesos, simplificación de tareas, accesibilidad, comodidad, control eficaz de los registros, reducción de costos y maximización de resultados.

Con la incorporación de este sistema se contribuyó en un mejor control de los registros implicados en los procesos. Siendo este una solución a los inconvenientes presentados como redundancia, pérdida de datos y problemas respecto al seguimiento del historial de pago de clientes.

Por otra parte, el sistema se desarrolló en plataforma web ya que brindan beneficios enormes en comparación con los sistemas tradicionales de escritorio dado que no necesitan instaladores, no están sujetos a las actualizaciones periódicas y sólo requieren un dispositivo con acceso a internet que abra un navegador para trabajar desde cualquier parte del mundo. Así mismo, se desarrolló una aplicación móvil, la característica principal que presentan los sistemas móviles es su portabilidad buscan que el manejo de la información sea de forma centralizada y ordenada.

4. Objetivos

4.1 Objetivo General

Desarrollar una aplicación web para la planificación de actividades y desarrollo de aplicación móvil para el monitoreo de los gestores de crédito usando una aplicación móvil para las Microfinancieras de Nicaragua

4.2 Objetivos Específicos

- Analizar la estructura de trabajo dentro de la Microfinanciera con el fin de obtener los requerimientos funcionales y no funcionales que deberá cumplir el sistema web y la aplicación móvil.
- Diseñar el modelado del sistema web y aplicación móvil a partir del análisis de los requerimientos utilizando las herramientas basadas en el lenguaje unificado de modelado (UML) haciendo uso de la metodología Cascada.
- Codificar un sistema web basado en el diseño realizado utilizando como herramienta de desarrollo el lenguaje PHP, HTML5, CSS3 y MySQL como gestor de base de datos.
- Codificar la aplicación móvil en Android mediante el IDE de desarrollo Android Studio y el lenguaje de programación JAVA.
- Implantar el sistema en un servidor web para la Microfinanciera garantizando su correcta operatividad y funcionamiento a través de una etapa de pruebas.

5. Marco Teórico

A continuación, se presenta el marco teórico, el cual estará compuesto por la información y conceptos necesarios para el desarrollo e implementación del presente proyecto, incluye los aspectos generales del desarrollo del sistema, conceptos básicos de las tecnologías a utilizar y las herramientas necesarias para el desarrollo del producto descrito en este documento.

5.1. Conceptos Generales

5.1.1 Microfinancieras

De acuerdo con (Accion International, 2021), son una organización no gubernamental que ofrece microcréditos, las microfinanzas son "la provisión de servicios de la banca o financieros dirigidos a hogares o microempresas de bajos o medianos ingresos económicos, incluyendo la opción de microcréditos." El sistema de microfinanzas, que casi siempre funcionan en el ámbito internacional, es una gama de servicios que, según (Oikocredit International, 2021), es una cooperativa financiera mundial, comprenden el microcrédito, la prestación de servicios de ahorros, pensiones, servicios de pago, préstamos para vivienda, seguros, emergencias y otros préstamos privados, etc. Las organizaciones que ofrecen estos servicios se conocen como instituciones microfinancieras (IMF) (en inglés, Microfinance Institutions o MFIs). Estas instituciones microfinancieras pueden ser entidades sin fines de lucro o también bancos comerciales y otras instituciones financieras.

5.1.2 Planificación de Actividades

Las actividades son eventos relacionados con un caso y que se planifican durante un período de tiempo determinado. Hay dos tipos de actividad: la actividad estándar y la actividad recurrente.

Una actividad estándar es una actividad puntual relacionada con un caso y planificada durante un período de tiempo determinado. Por ejemplo, una reunión puntual entre el cliente principal de un caso y un usuario.

Una actividad recurrente es una actividad que se repite a lo largo de un período. Por ejemplo, una visita supervisada semanalmente entre un menor y sus padres.

Las actividades difieren de las reuniones las actividades se utilizan para configurar un suceso básico en el calendario de un usuario para planificar el tiempo de la interacción con los participantes de un caso. (Corporation, IBM, 2023)

5.1.3 Aplicaciones Web

De acuerdo con (Zofío Jiménez, 2013) se denomina aplicación Web al software que reside en un ordenador, denominado servidor Web, que los usuarios pueden utilizar a través de internet o de una intranet, con un navegador web, para obtener los servicios que ofrezca.

Las aplicaciones Web se presentan en una amplia variedad de formas y tamaños, están escritas en todo tipo de lenguajes y se ejecutan en cualquier sistema operativo. En el núcleo de todas estas aplicaciones está la base de que todas sus funcionalidades son ejecutadas usando el protocolo HTTP y los resultados son formateados por lo general usando HTML.

La Ingeniería del software, la cual es la encargada de estudiar los principios y metodologías para el desarrollo y mantenimiento de sistemas de software, define aplicación Web como el conjunto de herramientas que los usuarios pueden usar para acceder a un servidor web a través de internet o intranet mediante el uso de navegadores Web. Dicho de otra forma, una aplicación Web es una aplicación de software que se codifica bajo un determinado lenguaje de programación, que es soportado por los navegadores Web y que sirve para que el usuario pueda interactuar con el servidor Web. Para desarrollar aplicaciones Web, se van a utilizar los lenguajes de programación o bien los entornos de desarrollo integrado,

conocidos como IDE, en los cuales se diseñará mediante código la aplicación Web y luego se podrá ir testeando con las herramientas que ofrece que dicho código es correcto y cumple con sus objetivos (Cardador Cabello, 2014).

5.1.4 Desarrollo de Aplicaciones Web

El desarrollo de aplicaciones informáticas evoluciona continuamente para adaptarse a las tecnologías de la información y las comunicaciones (TIC). El auge de internet y de la web ha influido notablemente en el desarrollo de software durante los últimos años. Actualmente la interfaz de los sistemas de información se implementa utilizando tecnologías web que ofrecen numerosas ventajas tales como el uso de una interfaz uniforme y la mejora del mantenimiento del sistema. Sin embargo, la existencia de numerosos estándares y los intereses de los fabricantes de tecnologías web dificulta el desarrollo de este tipo de aplicaciones (UMU, 2016).

“La web” es un estado de la World Wide Web en el cual las páginas web son diseñadas con un estilo diferente a las que componían el estado anterior, la denominada Web 1.0 o web de “Solo lectura”, donde la mayoría de las páginas eran estáticas, había apenas interacción con los usuarios, y solo el administrador (Webmaster) podía realizar las pocas actualizaciones que se realizaban de contenido. La denominada Web 2.0 o Web “Social” representa la evolución de las aplicaciones tradicionales hacia las aplicaciones web enfocadas al usuario final.

La Web 2.0 es más una actitud que una tecnología. Actualmente se va hacia la Web 3.0, se está desarrollando la web “Semántica”, donde las actividades de los usuarios se analizan en detalle (las conductas de comportamiento, el historial de compras, frecuencia y tamaño de pedidos, gustos, preferencias, etc.). Son datos que quedan registrados al navegar a través de la Web. La Web 3.0 será más “inteligente”, los usuarios podrán hacer búsquedas más cercanas al lenguaje natural. Por ejemplo, si una persona tiene que buscar vuelos mañana por la mañana, el servidor tendrá que entender si “Mañana” se refiere al día siguiente o

al periodo previo a las 12:00. Es decir, el servidor tiene que entender nuestro lenguaje (Zofío Jiménez, 2013).

5.1.5 Aplicaciones Móviles

Según (CINK.ES, 2019) una aplicación móvil o app es un tipo de software especial y de tipo específico, el cual se instala en dispositivos móviles como tabletas y teléfonos inteligentes.

Entre sus características, las apps son programas que se instalan en dispositivos móviles para llevar a cabo distintas tareas como de software regular se tratase. Sin embargo, su mayor diferencia radica en dos cualidades; la manera en que se obtiene la app en el dispositivo y su modo de uso directo y enfocado a una función menos general que el software tradicional de computadora. (CINK.ES, 2019)

5.1.6 Desarrollo de Aplicaciones Móviles

Una aplicación móvil, es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles. Este tipo de aplicaciones permiten al usuario efectuar un variado conjunto de tareas (Profesional, Ocio, Educativas, De acceso a servicios, etc.) facilitando las gestiones o actividades a desarrollar.

El desarrollo de aplicaciones móviles son los procedimientos y procesos establecidos que intervienen cuando se crea software para pequeños dispositivos informáticos. Al igual que el desarrollo de aplicaciones web, los procesos de desarrollo de aplicaciones móviles tienen sus raíces en el desarrollo de software tradicional. (Santiago & Trinaldo, 2015)

Por lo general se encuentran disponibles a través de ciertas plataformas de distribución, o por intermedio de las compañías propietarias de los sistemas operativos móviles tales como Android, IOS, Windows Phone, entre otros. Existen aplicaciones móviles gratuitas y otras de pago, donde en promedio el 20 a 30%

del coste de la aplicación se destina al distribuidor y el resto es para el desarrollador. (Zhang & Adipat, 2005)

Al ser las aplicaciones residentes en los dispositivos, estas están escritas en algún lenguaje de programación compilado; y su funcionamiento y recursos se encamina a aportar una serie de ventajas tales como: Un acceso más rápido y sencillo a la información necesaria, un almacenamiento de datos personales que a prioridad es seguro, una gran versatilidad en cuanto a su utilización, la atribución de funcionalidades específicas, mejorar la capacidad de conectividad y disponibilidad de servicios.

5.2 Metodología de desarrollo

5.2.1 Metodología Cascada

El desarrollo en cascada (en inglés, Waterfall Model) es un procedimiento lineal que se caracteriza por dividir los procesos de desarrollo en fases sucesivas del proyecto. Al contrario que en los modelos iterativos, cada una de estas fases se ejecuta tan solo una vez. Los resultados de cada una de las fases sirven como hipótesis de partida para la siguiente. El Waterfall Model se utiliza, especialmente, en el desarrollo de software. (IONOS, 2019)



Figura 1. Tomado de la página web: ionos.es - Diagrama Cascada

5.2.2 Fases de la Metodología Cascada

El modelo de desarrollo en cascada sigue una serie de etapas de forma consecutiva, la etapa siguiente empieza cuando termina la etapa anterior. Las fases que componen el modelo son las siguientes:

- Análisis/Requisitos del software.

En esta fase se hace un análisis de las necesidades del cliente para determinar las características del software a desarrollar, y se especifican todas las funciones del sistema sin entrar en detalles técnicos. Hay que ser especialmente cuidadoso en esta primera fase, ya que en este modelo no se pueden añadir nuevos requisitos en mitad del proceso de desarrollo.

- Diseño.

En esta etapa se describe la estructura interna del software, así como su apariencia, y las relaciones entre las entidades que lo componen.

- Implementación.

En esta fase se programan los requisitos especificados haciendo uso de las estructuras de datos diseñadas en la fase anterior. La programación es el proceso que nace de la formulación de un problema llevado a la computación, de ahí a un programa que se ejecute produciendo los pasos necesarios para resolver dicho problema.

- Verificación.

Como su propio nombre indica, una vez se termina la fase de implementación se verifica que todos los componentes del sistema funcionen correctamente y cumplen con los requisitos.

- Instalación y mantenimiento.

Una vez se han desarrollado todas las funcionalidades del software y se ha comprobado que funcionan correctamente, se inicia la fase de instalación y mantenimiento. Se instala la aplicación en el sistema y se comprueba que funcione correctamente en el entorno en que se va a utilizar.

5.3 Herramientas de desarrollo

5.3.1 UML (Lenguaje Unificado de Modelado)

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común, semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación.

UML es popular entre programadores, pero no suele ser usado por desarrolladores de bases de datos. Una razón es sencillamente que los creadores de UML no se enfocaron en las bases de datos. A pesar de ello, el UML es efectivo para el modelado de alto nivel de datos conceptuales y se puede usar en diferentes tipos de diagramas UML. (Lucid Software Inc., 2022)

Algunos de los diagramas más comunes del UML son:

- Diagrama de Clases
- Diagrama de Objetos
- Diagrama de Casos de Uso
- Diagrama de Estados
- Diagrama de Secuencias
- Diagrama de Actividades
- Diagrama de Colaboraciones
- Diagrama de Componentes
- Diagrama de Distribución

5.3.2 Programación Orientada a Objetos (POO)

La programación orientada a objetos (POO, u OOP según sus siglas en inglés) es un paradigma de programación que usa objetos en sus interacciones, para diseñar aplicaciones y programas informáticos. Está basada en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento. Su uso se popularizó a principios de la década de 1990. En la actualidad, existe una gran variedad de lenguajes de programación que soportan la orientación a objetos.

La programación orientada a objetos es una de las formas más populares de programar y viene teniendo gran acogida en el desarrollo de proyectos de software desde los últimos años. Esta acogida se debe a sus grandes capacidades y ventajas frente a las antiguas formas de programar.

5.3.3 JAVA

Java es la base para prácticamente todos los tipos de aplicaciones de red, además del estándar global para desarrollar y distribuir aplicaciones móviles y embebidas, juegos, contenido basado en web y software de empresa. Con más de 9 millones de desarrolladores en todo el mundo, Java le permite desarrollar, implementar y utilizar de forma eficaz interesantes aplicaciones y servicios. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.

Java está diseñado para permitir el desarrollo de aplicaciones portátiles de elevado rendimiento para el más amplio rango de plataformas informáticas posible al poner a disposición de todo el mundo aplicaciones en entornos heterogéneos, las empresas pueden proporcionar más servicios y mejorar la productividad, las comunicaciones y colaboración del usuario final y reducir drásticamente el costo de propiedad tanto para aplicaciones de usuario como de empresa. (Oracle, 2023)

5.3.4 Android Studio

Android Studio fue anunciado en el año 2013 en la conferencia de Google I/O fue creado para reemplazar a Eclipse, la plataforma que se usaba para la creación de aplicaciones y que todavía hoy en día es utilizada por muchos programadores. De esta manera, con Android Studio, Google consigue su propio IDE para el desarrollo de aplicaciones, pudiendo instalar todo el SDK para desarrollar apps específicas adaptadas a la mayor parte de versiones.

En la actualidad Android Studio es la plataforma que se postula como el más completo IDE para desarrollar aplicaciones Android con muchas características que destacan de los otros programas usados para este trabajo, está basado en IntelliJ y puede ser descargado de forma gratuita a través de la licencia de Apache 2.0.

Cuenta con una estructura simple que permite organizar los proyectos de manera que facilite su ubicación y su publicación, como también un entorno para desarrollar más potente, fácil e intuitivo. Permite ver el desarrollo a tiempo real de las aplicaciones y las pantallas en las que será usada la aplicación, y a su vez nos ofrece plantillas para diferentes elementos para programar como el uso de mapas. (AS, 2016)

5.3.5 PHP

PHP es un lenguaje de scripting popular multipropósito que es especialmente útil para el desarrollo web rápido, flexible y pragmático, PHP potencia todo desde blogs hasta la web más popular alrededor del mundo con seguridad y orientado a objetos, que permite a los desarrolladores crear una gran variedad de aplicaciones seguras y sólidas. Puede usar PHP para crear aplicaciones cliente de Windows, servicios web XML, componentes distribuidos, aplicaciones cliente-servidor, aplicaciones de base de datos y muchas, muchas más cosas.

La sintaxis de PHP es muy expresiva, pero también sencilla y fácil de aprender. Cualquier persona familiarizada con C++ o Java, reconocerá al instante la sintaxis de PHP. (The PHP Group, 2001-2023)

5.3.6 Visual Studio Code

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Permite tener varios documentos abiertos mediante pestañas, e incluso emplear varios paneles para aquellos que utilicen más de un monitor. Dispone de modo de pantalla completa, para aprovechar al máximo el espacio visual disponible de la pantalla. Para navegar por el código cuenta con un Minimap, un panel que permite moverse por el código de forma rápida. Dispone de autoguardado, muchas opciones de personalización, cuenta con un buen número de herramientas para la edición del código y automatización de tareas. Soporta macros, Snippets y autocompletar, entre otras funcionalidades. Algunas de sus características son ampliables mediante extensiones.

Es una herramienta concebida para programar sin distracciones. Su interfaz de color oscuro y claro provee una riqueza de coloreado de sintaxis, que centra nuestra atención completamente. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C++, C#, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity). (Microsoft, 2023)

5.3.7 HTML 5

HTML 5 (Hypertext Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML 5 especifica dos variantes de sintaxis para HTML: una «clásica», HTML (text/html), conocida como HTML 5, y una variante XHTML conocida como sintaxis XHTML 5 que deberá servirse con sintaxis XML (application/xhtml+xml). Esta es la primera vez

que HTML y XHTML se han desarrollado en paralelo. La versión definitiva de la quinta revisión del estándar se publicó en octubre de 2014.

Al no ser reconocido en viejas versiones de navegadores por sus nuevas etiquetas, se recomienda al usuario común actualizar su navegador a la versión más nueva, para poder disfrutar de todo el potencial que provee HTML 5.

El desarrollo de este lenguaje de marcado es regulado por el Consorcio W3C. (Standard, H.L., 2023)

5.3.8 CSS 3

CSS (siglas en inglés de Cascading Style Sheets), en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles (como Firefox OS).

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características presentacionales, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento. (W3C, 2023)

5.3.9 JavaScript

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas y JavaScript del lado del servidor (Server-Side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo, en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo. Desde 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1, una versión de JavaScript. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se publicó en julio de 2015.

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes. Su relación es puramente comercial, tras la compra del creador de Java (Sun Microsystems) de Netscape Navigator (creador de LiveScript) y el cambio de nombre del lenguaje de programación.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. JavaScript es el único lenguaje de programación que entienden de forma nativa los navegadores. Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias se van descargando junto con el código HTML. (mozilla.org contributors, 1998-2023)

5.3.10 JQuery

jQuery es una biblioteca de JavaScript rápida, pequeña y rica en funciones. Hace que cosas como el recorrido y la manipulación de documentos HTML, manejo de eventos, la animación y Ajax sean mucho más simples con una API fácil de usar que funciona en una multitud de navegadores. Con una combinación de versatilidad y extensibilidad, jQuery ha cambiado la forma en que millones de personas escriben JavaScript. (OpenJS Foundation, 2022)

5.3.11 MVC

MVC son las siglas de Model View Controller, o su traducción Modelo Vista Controlador.

Una aplicación web basada en este patrón separa su código en tres partes diferenciadas:

- El controlador: Es el punto de entrada de la aplicación, se mantiene a la escucha de todas las peticiones, ejecuta la lógica de la aplicación, y muestra la vista apropiada para cada caso.
- El modelo: Contiene todo el código relacionado con el acceso a datos. Es importante que sea un código lo más genérico posible y se pueda reutilizar en otras situaciones y proyectos. El modelo incluye consultas a la base de datos y validaciones de entrada de datos.
- La vista: la vista contiene el código que representará lo que se visualizará por pantalla.

El objetivo de usar este patrón de diseño es desacoplar lo más posible el código de la aplicación, y así potenciar la flexibilidad y la reutilización de sus partes. (Conceptos de Programación)

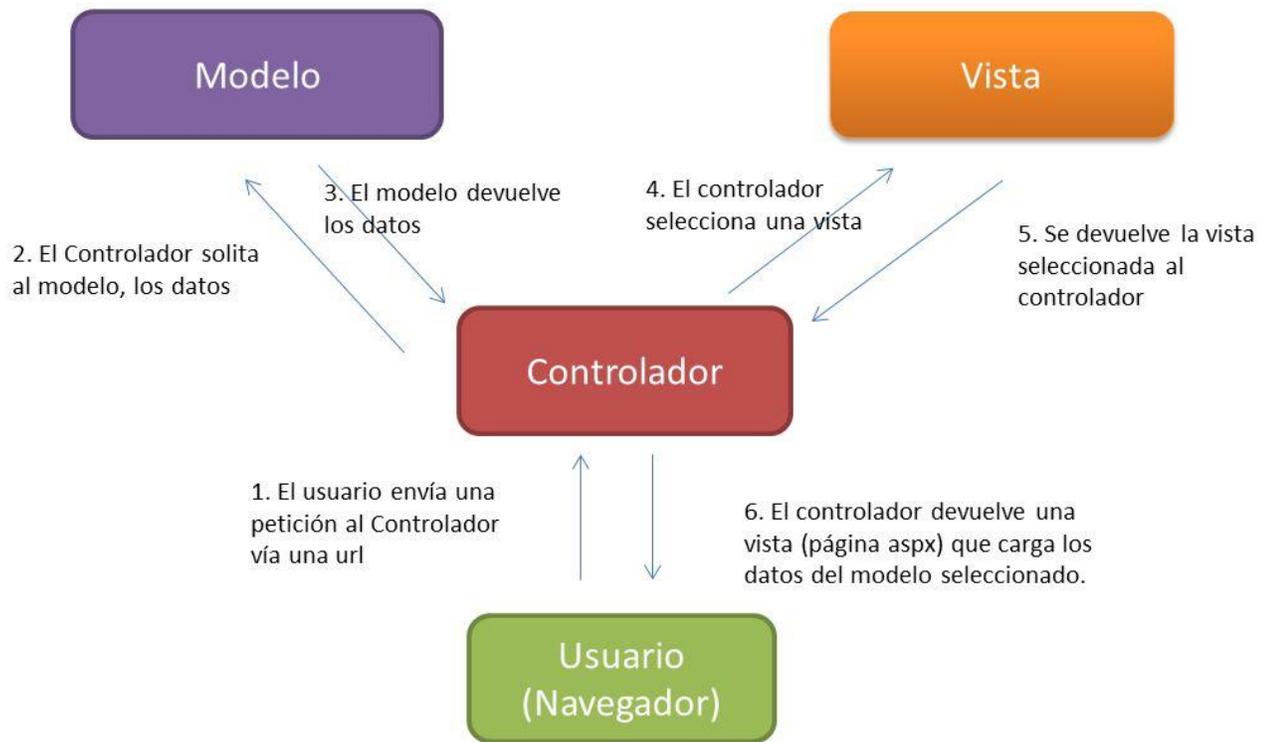


Figura 2. Tomado de la página web: [Mi Blog Técnico](#) - Diagrama MVC

5.3.12 MySQL

Es un sistema de gestión de bases de datos relacionales (RDBMS) de Oracle que está diseñado para el entorno personal y empresarial. MySQL se ejecuta en SQL (Structure Query Language). (Oracle Corporation, 2023)

6. Diseño Metodológico

En este apartado, se hablo acerca de la metodología que se estuvo utilizando durante el desarrollo del sistema mencionado.

La metodología que se selecciono es Cascada, (en inglés, Waterfall Model) nos proporcionó un desarrollo lineal de modelado y pudo efectivamente ofrecer un resultado final más predecible en términos de presupuesto, línea del tiempo y alcance.

Así mismo, esta metodología cuenta con una secuencia de fases que determinan las actividades a desarrollar durante el ciclo de vida del proyecto.

Estas son:

1. Análisis de requerimientos
2. Diseño de la solución
3. Implementación
4. Verificación
5. Mantenimiento

A continuación, detallaremos cada una de las fases.

6.1 Análisis de Requerimientos

Durante esta fase se estuvo definiendo la problemática y necesidades que se presentan en la microfinanciera **Oportucredit S.A** para así determinar la solución a estas mismas. Para la recolección de la información se estuvieron utilizando métodos como entrevistas, observación a los interesados principales y de cómo se llevan a cabo las actividades diarias.

Entre las herramientas que se estuvieron utilizando en esta fase son:

- **Especificación de los requerimientos del sistema:** Los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos

requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema. (Sommerville, 2005)

- **Modelos de casos de uso:** esta herramienta ayudó en el análisis respecto al comportamiento que tendrá el usuario final dentro del sistema.
- **Diagrama de actividades:** Utilizando un diagrama de actividades puede mostrarse los procesos que se siguen, en el mismo los modelos se representan con nodos de objetos y las transformaciones como actividades estereotipadas (círculos icónicos especiales), una cadena de transformaciones define el flujo de control.
- **Diagrama de estado:** Mediante un diagrama de estados, es posible visualizar los diversos estados por los que atraviesa un sistema o entidad, representando cada estado como un nodo. Los eventos que desencadenan transiciones entre estados se expresan como actividades estereotipadas, a menudo simbolizadas por flechas direccionales. Dichas transiciones definen la evolución del sistema a lo largo del tiempo, esbozando claramente el flujo de control. Similar al diagrama de actividades, el diagrama de estados ofrece una representación gráfica intuitiva que facilita la comprensión de los procesos dinámicos y las interacciones complejas en sistemas en constante cambio.

6.2 Diseño de solución

Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web y móvil.

El sistema está compuesto de las siguientes secciones y módulos:

Módulos del sistema web

- Supervisores
 - Añadir un supervisor
 - Eliminar supervisor
- Formularios
 - Crear formularios
 - Actualizar formularios
 - Inactivar formularios
 - Ver formulario
- Monitoreo
 - Listar todos los usuarios con su ubicación y porcentaje de batería
 - Ver la trayectoria del usuario por fecha
 - Observar el kilometraje del usuario
- Clientes
 - Registrar Clientes
 - Editar Clientes
 - Eliminar Clientes
 - Importar clientes
- Usuarios
 - Crear usuario web
 - Brindar token móvil
 - Editar usuario
 - Eliminar un usuario
 - Asignar rol
- Planificación
 - Crear una actividad
 - Editar una actividad
 - Ver una actividad
- Reportes
 - Generar reportes por rango de fecha y atributo
- Rutas
 - Crear rutas
 - Cambiar de estado
- Sucursales
 - Crear una sucursal
 - Editar una sucursal
 - Eliminar una sucursal
 - Mostrar una sucursal

Módulos del apartado Móvil

- Actividades
 - Listar todas las actividades asignadas
 - Completar actividades
- Clientes
 - Registrar clientes
 - Registrar visita de cliente
- Rutas
 - Mostrar todos los clientes de la ruta asignada
- Monitoreo
 - Guardar trayectoria del usuario
 - Guardar ubicación GPS de cualquier evento completado
- Alertas
 - Generar alertas por eventos de desactivación de GPS o datos móviles
- Formularios
 - Completar formularios previamente creados

En la fase de diseño se utilizó la información recolectada anteriormente para resolver la problemática y cumplir con los requisitos anteriormente propuestos.

Las herramientas para utilizar en esta fase son:

- **Modelo de navegación:** En un sistema para la web es útil saber cómo están enlazadas las páginas. Ello significa que necesitamos un diagrama conteniendo nodos y enlaces.
- **Modelo relacional:** Un modelo relacional consiste en representar datos por medio de tablas relacionadas cuyas filas se llaman tuplas y las columnas variables, conformando así una base de datos. (Arias, 2020)
- **Diseño Interfaces Graficas de Usuarios**

6.3 Implementación

Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

Es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final.

6.4 Verificación

Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.

En esta fase se asegurará el cumplimiento de los requerimientos del sistema haciendo pruebas de caja negra, de igual manera pruebas de validación con la interacción del usuario final demostrando que todos los casos de usos se encuentran funcionales.

6.5 Mantenimiento

Una vez implementado el sistema, con esta fase se pretende mantener el sitio siempre actualizado. Además, es posible encontrar errores que no se detectaron durante el proceso de desarrollo por medio de la retroalimentación de parte de los usuarios durante el tiempo que han utilizado el sistema.

7. Estudio de Factibilidad

Para determinar la viabilidad del desarrollo e implantación de un equipo ya sea software o hardware, es necesario la ejecución de un estudio previo que permita obtener las valoraciones a través de la recolecta de información importante para la toma de decisiones dentro de un negocio donde se quieran realizar los cambios, estudio conocido como Estudio de Factibilidad. El estudio de factibilidad está conformado de 4 aspectos básicos: Técnico, Operativo, Económico y Legal.

7.1 Factibilidad técnica

7.1.1 Recursos técnicos

7.1.1.1 Especificaciones de Hardware

Respecto al hardware la empresa cuenta con un equipo de cómputo laptop el cual cumple con los requerimientos técnicos para la ejecución del sistema, cabe mencionar que el sistema estará alojado en un servidor web, debido a esto no se requiere la compra de hardware con especificaciones superiores al ya existente. Sin embargo, se recomienda que conforme al crecimiento de la empresa se adquieran más equipos.

La siguiente tabla muestra las especificaciones del hardware con que cuenta actualmente la empresa.

Equipo	Cantidad	Marca	Especificación
Laptop	1	Dell	Procesador: Intel® Core™ i7-7600U 2.80 GHz Arquitectura: 64 bits Pantalla: 14.0" de alta definición completa (FHD) (1920 x 1080) Memoria RAM: 8GB DDR3 Gráfica: Tarjeta Gráfica Intel HD Graphics. Disco Duro: Hard Disk Drive (HDD) 1 TB

Tabla 1. Especificaciones de Hardware. Fuente: Equipo en posesión

7.1.1.2 Especificaciones de Software

Sistema Operativo	Programas
Windows 10 Ultimate 64 Bits	Office 2016
	Adobe Reader
	Navegadores: Mozilla Firefox, Google Chrome, Microsoft Edge
	Herramientas de Windows
	Windows Defender

Tabla 2. Especificaciones de Software. Fuente: Equipo en posesión

7.2 Factibilidad Operativa

El desarrollo e implantación del proyecto, será operativamente viable para la ejecución de las tareas planteadas por el representante de la empresa GICAPP. Dado que facilitará la administración y el control de las actividades, brindando una herramienta necesaria para realizar un trabajo más ágil y dinámico.

Para un mejor alcance como organización la plataforma se desarrollará lo más amigable posible, en base a los requerimientos del cliente, de tal forma que sin mucha dificultad los usuarios, puedan adaptarse y aprovechar al máximo las facilidades que está brinde, ahorrando gran parte de su tiempo y permitiendo la realización de otras actividades.

Los usuarios que harán uso del sistema no necesitan un alto nivel de conocimiento en el área de informática para poder utilizarlo, solamente se requiere que el usuario tenga una mínima experiencia en el uso de internet y que tengan acceso a este servicio. Sin embargo, se realizarán capacitaciones durante un lapso establecido en consenso a la directiva; debido a esto no se espera un mayor obstáculo durante la incorporación del sistema y posterior puesta en marcha de este. El éxito de este proyecto a nivel operacional permitirá mejorar los métodos de trabajo dentro del entorno de la organización, ahorrando principalmente tiempo y esfuerzo. De esta manera, se cuenta con la aprobación del representante de la empresa para la implantación del sistema con los cambios y mejoras que este ofrezca.

7.3 Factibilidad Económica

Esta parte del trabajo se basa en cuantificar el costo monetario tanto del diseño y desarrollo del sistema como de los recursos tipo hardware y software necesario para su implementación.

7.3.1 Costo de Adquisición de Hardware

Tal como se mencionó en la sección de factibilidad técnica no es necesario realizar una inversión en equipo de hardware ya que la empresa cuenta con el equipo técnico necesario para una correcta ejecución del sistema.

7.3.2 Costo de Adquisición de Software

Para la ejecución del desarrollo e implantación de este proyecto no se incurrirá en gastos respecto a software para la configuración de servidores, y en licencias para las herramientas de desarrollo, puesto que en su mayoría gozan de licencia de software libre GNU GPL (General Public License) distribuyendo de forma gratuita.

En la tabla siguiente se presentará el costo en cuanto a software.

Descripción	Costo
Visual Studio Code ³	Licencia Libre GPL
Lucid Chart ⁴	Libre/Pago
Gestor de Base de datos Mariadb ⁵	Licencia Libre GPL
Ninjamock ⁶	Libre/Pago
Costo Total	USD \$0.00

Tabla 3. Costo de Adquisición de Software.

³ <https://code.visualstudio.com>

⁴ <https://lucid.app/pricing/lucidchart>

⁵

https://mariadb.org/download/?t=mariadb&p=mariadb&r=11.4.0&os=windows&cpu=x86_64&pkg=msi&m=acorn

⁶ <https://ninjamock.com/account/pricing>

7.3.3 Costo Recurrente

Para la implantación del sistema se considera la contratación de un servicio de alojamiento web con la empresa Hostinger a un costo promocional de USD \$39,16, con los beneficios de dominio gratuito por un año y un certificado SSL permanente gratuito.

En la tabla siguiente se presentan los costos estimados por la renovación de los servicios:

Meses	Descripción	Costo
12	Hosting	USD \$59,93
12	Dominio	USD \$13,33
	Certificado SSL ⁷	USD \$12,24

Tabla 4. Costo Recurrente.

7.3.4 Costo Equipo de Desarrollo (Recursos Humanos)

Rol	Cantidad	Salario Mensual	Tiempo Estimado de Desarrollo	Costo
Desarrollador Jr	2	USD \$700,00	12 meses	USD \$8400,00
Costo Total				USD \$16800,00

Tabla 5. Costo del Equipo de Desarrollo.

Los valores antes mencionados representan el costo de elaboración de una aplicación, no obstante, para la empresa GICAPP no tendrá costo alguno ya que se considerará como un aporte de la Universidad en beneficio de la sociedad.

⁷ <https://www.hostinger.com/> (Aplica para Hosting, Dominio y Certificado).

7.4 Factibilidad Legal

En el presente proyecto se respeta y se hace cumplir la ley de los derechos de autor cumpliendo con todas las prerrogativas que dicha ley establece, con el objetivo de evitar multas o demandas a la hora de implementar el sistema. Además, se tuvo en cuenta la ley número **787 “LEY DE PROTECCIÓN DE DATOS PERSONALES”**, especialmente en su artículo número 7 en la Obligación de informar al obtener los datos personales teniendo en cuenta a GICAPP, como Cliente Final, y donde se refiere a datos personales como la información de sus procesos productivos.

7.4.1 Requerimientos Legales

- Aseguramiento de que el proyecto no infringe ninguna norma o ley establecida ya sea Municipal o Nacionalmente.
- Se debe garantizar el respeto a los acuerdos, convenios y reglamentos internos de tipo empresarial, industrial, sindical, religioso, partidista, cultural, deportivo u algún otro relacionado con el ámbito del proyecto.

7.4.2 Licencia para el Software

Nos permite determinar los derechos que tienen los autores sobre la documentación realizada por estos en este proyecto, la cual es exclusividad de los desarrolladores del sistema.

7.4.3 Contratos de Servicios

Se podrá encontrar en el apartado de **Anexos 1**.

7.4.4 Beneficios de Automatizar

Los beneficios de este proyecto se clasificaron en dos: Beneficios tangibles y Beneficios intangibles.

7.4.4.1 Beneficios Tangibles

- Agilización en los tiempos de respuesta ante una solicitud de datos.
- Se gestionan los datos con mayor rapidez lo que permite al personal realizar las tareas en menor tiempo.
- Reducción de gastos en artículos de oficina.
- Persistencia de la información al estar almacenada en un gestor de base de datos en la nube y de esta manera no ocurre un deterioro de esta con el tiempo.

7.4.4.2 Beneficios Intangibles

- Mayor disponibilidad de la información almacenada al realizar consultas vía internet.
- Fácil manejo del sistema por parte del usuario ya que contará con una interfaz de fácil comprensión, manejo y uso de esta.
- Reducción en tiempos de respuesta del sistema.
- Mayor control y seguimiento de los registros contables referentes a gestión de cobro.
- Mejor imagen de la empresa hacia sus clientes al proporcionar confianza y seguridad en la administración de sus datos.

8. Análisis y presentación de resultados

En esta fase se presenta el análisis del Sistema web para la planificación de actividades y monitoreo de los gestores de crédito usando una aplicación móvil de recolección de información para las Microfinancieras de Nicaragua y los resultados de este. Para tal efecto abordamos elementos como: requerimientos funcionales y no funcionales, definición de actores y su respectiva jerarquía, descripción de los escenarios presentes por medio de diagramas de casos de uso y diagramas de actividades.

Estaremos describiendo las fases que determinan las actividades a desarrollar durante el ciclo de vida del proyecto.

Siendo estas:

- Fase 1: Análisis de requerimientos
- Fase 2: Diseño de la solución
- Fase 3: Implementación
- Fase 4: Verificación
- Fase 5: Mantenimiento

A continuación, se desarrolló cada una de las fases.

8.1 Fase 1: Análisis de requerimientos

En esta fase se presenta el análisis del Sistema web para la planificación de actividades y monitoreo de los gestores de crédito usando una aplicación móvil de recolección de información para las Microfinancieras de Nicaragua., para tal efecto se abordan elementos como: requerimientos funcionales y no funcionales, definición de actores y su respectiva jerarquía, descripción de los escenarios presentes por medio de diagramas de casos de uso y diagramas de actividades.

8.1.1 Análisis de Requerimientos

El sistema debe cumplir una serie de objetivos, para lo cual se han establecido requerimientos funcionales los cuales representaron el comportamiento interno del sistema. Así mismo, un conjunto de requerimientos adicionales o requerimientos no funcionales que se pueden considerar restricciones que inciden en el comportamiento del sistema, pero no refieren a funciones que ejecutará; estos corresponden a aspectos tales como la disponibilidad, mantenibilidad, flexibilidad, seguridad, facilidad de uso, etc.

Se presentarán los requisitos recopilados de entrevistas y reuniones.

8.1.1.1 Requerimientos Funcionales

Código	Descripción
Requerimientos generales del sistema	
RF.01	El sistema permite tener acceso al inicio de sesión a cuatro tipos de usuarios: Administrador, Supervisor, Gestor y Promotor.
RF.02	El sistema debe facilitar el perfil de usuario para la edición de campos necesarios (Nombre de usuario, contraseña, etc.).
Requerimientos funcionales del Administrador	
RF.03	El sistema debe de permitir al Administrador: Listar, registrar, asignar rol, editar, filtrar y dar de baja a usuarios.
RF.04	El sistema debe de permitir al Administrador: Listar, registrar, actualizar, filtrar y dar de baja roles.
RF.05	El sistema debe permitir al Administrador: Listar, registrar, filtrar y eliminar planificaciones.
RF.06	El sistema debe permitir al Administrador: Listar, Registrar, Filtrar y Eliminar rutas.

RF.07	El sistema debe permitir al Administrador: Listar, Registrar, Filtrar y Eliminar sucursales.
RF.08	El sistema debe permitir al Administrador: Registrar, Editar, Eliminar e Importar clientes.
RF.09	El sistema debe permitir al Administrador: Generar, Filtrar y Listar reportes.
RF.10	El sistema debe permitir al Administrador: Crear, Actualizar, Inactivar y Listar Formularios
RF.11	El sistema debe permitir al Administrador Listar todos los usuarios con su ubicación y porcentaje de batería, Ver la trayectoria del usuario por fecha y observar el kilometraje del usuario
Requerimientos generales del Supervisor	
RF.12	El sistema debe permitir al supervisor: Listar, crear, filtrar y actualizar los móviles.
RF.13	El sistema debe permitir al supervisor: Listar, crear, filtrar y actualizar las rutas.
Requerimientos generales del Gestor	
RF.14	El sistema debe permitir al gestor: Listar y actualizar las rutas que tiene asignadas.
Requerimientos generales del Promotor	
RF.15	El sistema debe permitir al promotor registrar clientes prospectados para nuevos créditos.

Tabla 6. Requerimientos Funcionales del sistema.

8.1.1.2 Requerimientos no Funcionales

Código	Descripción
RNF.01	El sistema debe mostrar una página principal para los usuarios del sistema.
RNF.02	El sistema debe mostrar interfaces adecuadas para cada tipo de usuario.
RNF.03	El sistema debe brindar mensajes de error informativo a los usuarios.
RNF.04	El ingreso al sistema debe estar restringido por credenciales.
RNF.05	La aplicación web debe poseer un diseño "Responsive" a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tablets y teléfonos inteligentes.
RNF.06	Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.
RNF.07	El sistema se visualizará y funcionará correctamente en las versiones más recientes de navegadores web, especialmente: Mozilla Firefox, Chrome y Microsoft Edge.
RNF.08	Los campos de los formularios deben estar correctamente validados para evitar el envío de datos incorrectos a la base de datos.

Tabla 7. Requerimientos no Funcionales del sistema.

8.1.2 Definición de Actores

Los casos de uso definen las diferentes funciones que serán realizadas por el sistema en consecuencia de la interacción de este con cada uno de los actores definidos por el rol que se le asigne. En el Sistema web para la planificación de actividades y monitoreo de los gestores de crédito usando una aplicación móvil de recolección de información para las Microfinancieras de Nicaragua, interactúan 4 tipos de actores.

A continuación, se definirán los actores, sus funciones y la jerarquía de estos.

8.1.2.1 Jerarquía de Usuarios

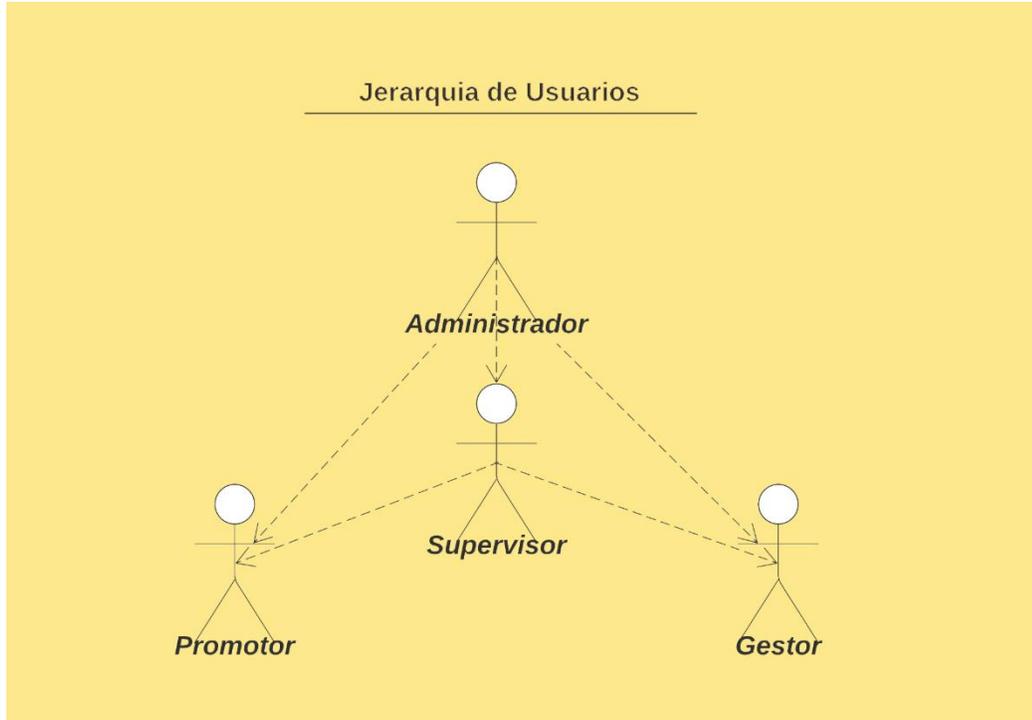


Figura 3. Jerarquía de Usuarios.

8.1.2.2 Definición de Funciones

Tipo	Descripción
Administrador	El Administrador, es el encargado del registro de nuevos usuarios de igual manera asignar los roles que considere respectivos como lo son Administrador, Supervisor, Promotor y Gestor, así mismo se encarga; de igual manera, lleva la gestión, administración (formularios, sucursales), gestionar los servicios, planes de pago y los reportes.
Supervisor	El Personal Supervisor, tiene la funcionalidad de supervisar más de una ruta asignada que cada gestor haya pasado.
Promotor	El Personal Promotor, se encarga de prospectar créditos es decir tiene acceso a crear nuevos clientes, pero no ver clientes existentes.
Gestor	El Personal Gestor, tiene la responsabilidad de llevar el control de los cobros y realizar el registro de los datos generales de los clientes.

Tabla 8. Funciones de los Actores. Fuente: Entrevista

8.1.3 Diagrama de Casos de Usos

Se presentan los diagramas de caso de uso principales y secundarios del sistema los cuales señalan la interacción entre el grupo de usuario con el sistema desarrollado; junto con la descripción de cada uno de los casos haciendo uso de plantillas de caso de uso en las que se describe en qué consiste, qué actores intervienen, el flujo normal, postcondiciones y otras especificaciones de cada caso representado en el diagrama.

Las plantillas de caso de uso seguirán la estructura que se detalla a continuación.

Plantillas de casos de usos

ID	Identificador único de cada caso de uso
Nombre	Nombre completo del caso de uso
Autor	Nombre del autor (o autores) del caso de uso
Fecha	Fecha de creación del caso de uso
Actores	Especificación de los usuarios que interactúan con el caso de uso
Prioridad	Nivel de importancia de la realización del caso de uso
Frecuencia	Regularidad con la que se lleva a cabo el caso de uso
Descripción: Breve explicación del funcionamiento del caso de uso.	
Precondiciones: Establece las condiciones que deben cumplirse para la ejecución del caso de uso.	
Flujo Normal: Listado de las actividades necesarias para el correcto desarrollo del caso de uso.	
Flujo alternativo: Son las posibles salidas del sistema, en caso de que el usuario decida no continuar con el flujo normal del caso de uso.	
Postcondiciones: Establece el estado del sistema cuando el caso de uso se realiza con éxito.	

Tabla 9. Obtenida de la Pagina web [ingeniería de Software](#) Plantilla de casos de uso.

Niveles de prioridad del caso de uso

Prioridad	Descripción
Baja	El caso de uso no afecta el funcionamiento de otros casos.
Media	Importante, no es necesaria la ejecución del caso de uso constantemente.
Alta	Obligatorio, el caso de uso afecta el funcionamiento de otros casos.

Tabla 10. Niveles de prioridad

Frecuencia	Descripción
Baja	Raramente
Media	Algunas veces
Alta	Siempre

Tabla 11. Frecuencia.

Nota: A continuación, se presenta el diagrama de caso de uso para el actor de tipo supervisor con sus respectivas plantillas de caso de uso para presentar la interacción de estos actores con el sistema.

8.1.3.1 Casos de Uso – Administrador

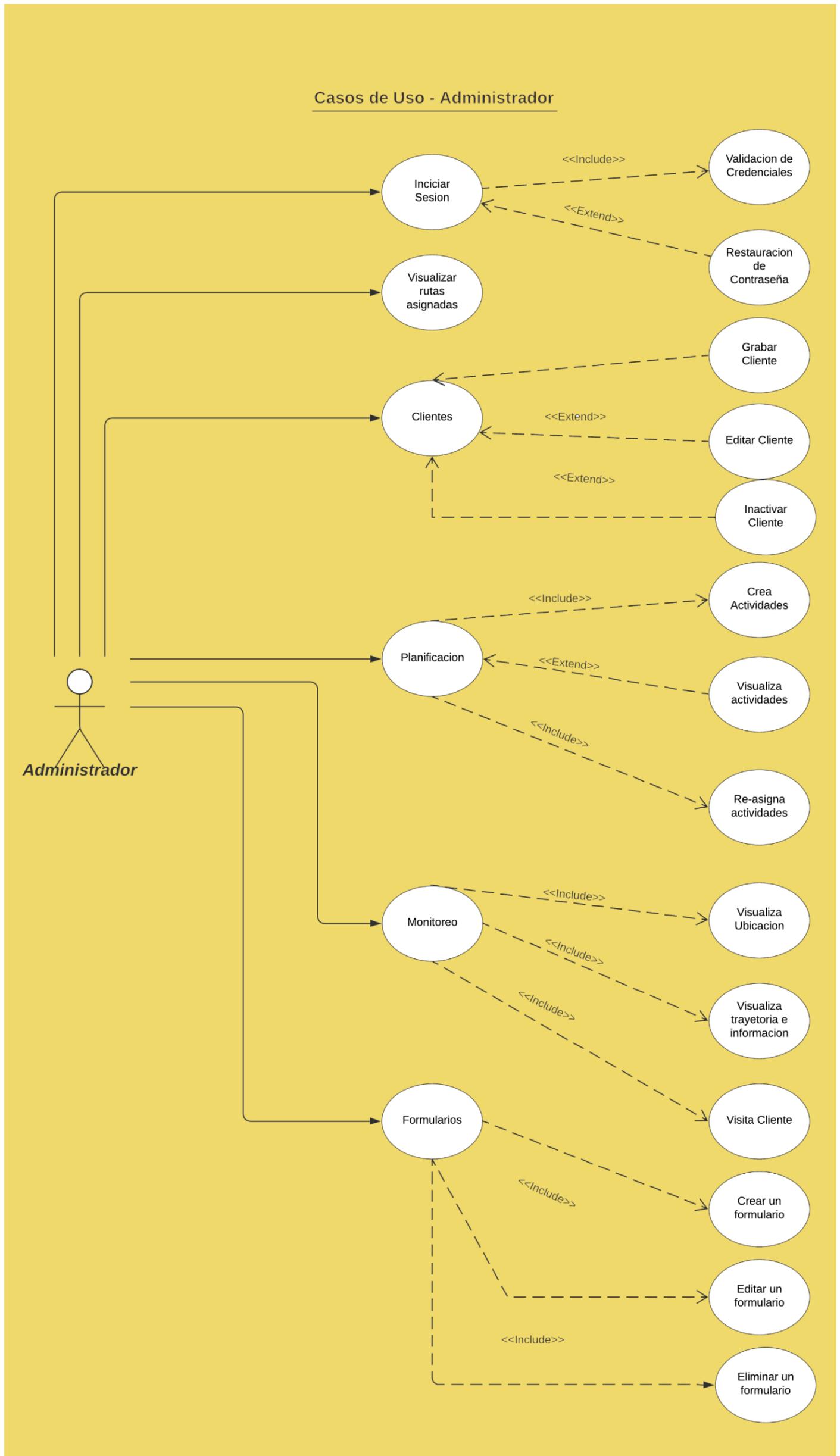


Figura 4 Caso de uso administrador

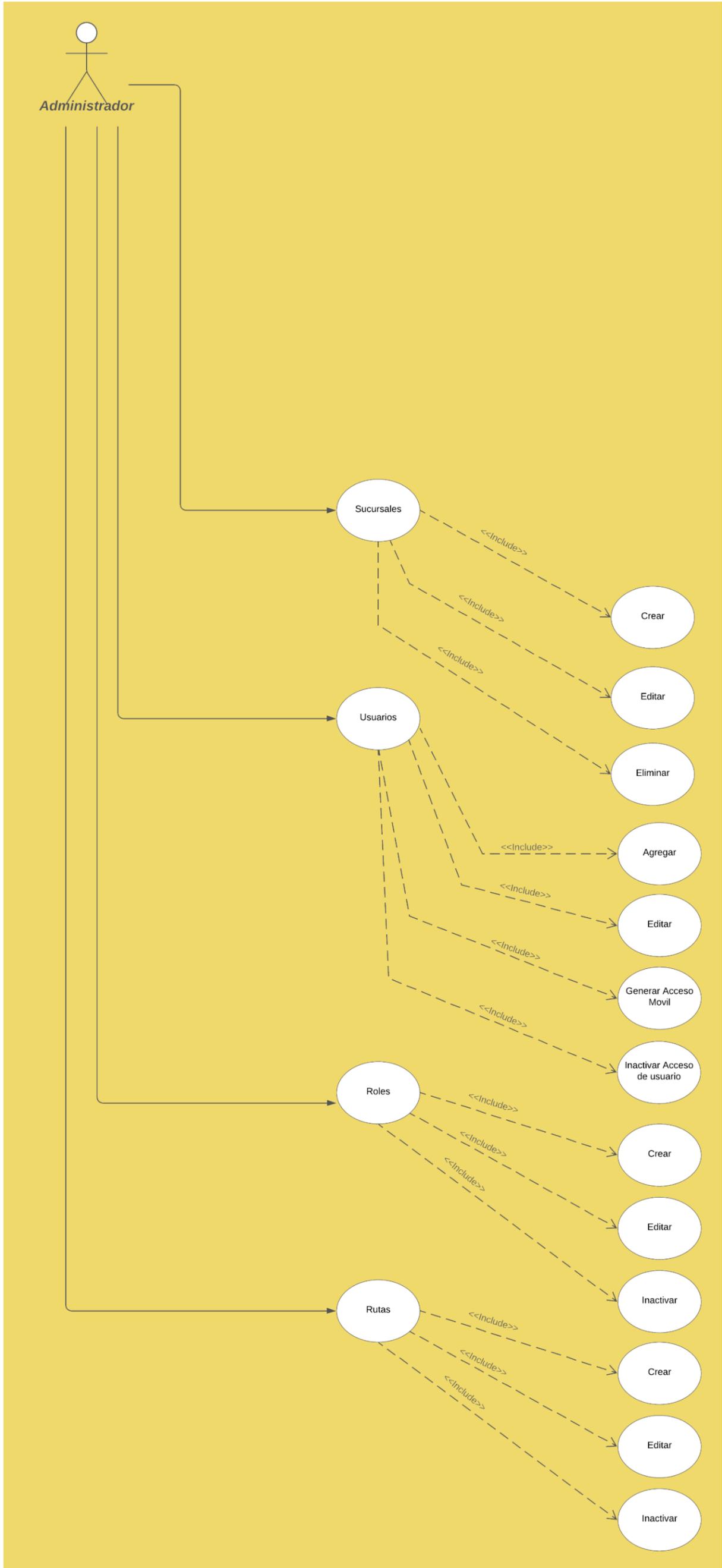


Figura 5 Caso de uso administrador Continuación

8.1.3.2 Casos de Uso – Supervisor

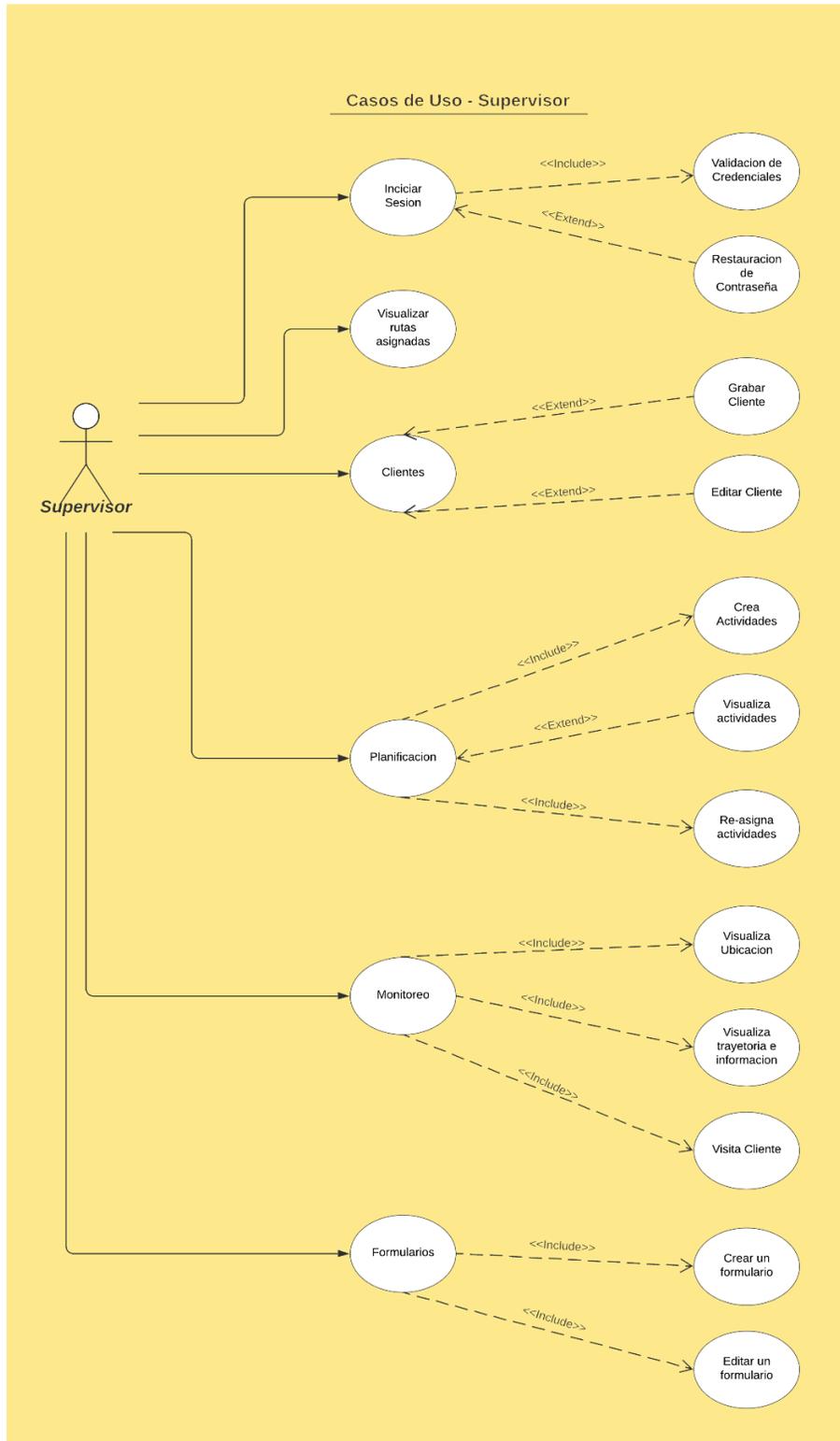


Figura 6. Casos de uso Supervisor.

ID	CU 40
Nombre	Crear ruta
Autor	Javier Enmanuel Matus Widdy
Fecha	26/10/2020
Actores	Administrador, Supervisor
Prioridad	Alta
Frecuencia	Alta
Descripción: Permitir al Actor crear una ruta.	
Precondiciones: Los actores deben de estar autenticados en el sistema y tener permisos de administrador o supervisor.	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor pulsa un botón "Crear ruta". 2. El sistema muestra un formulario con los campos necesarios. 3. El actor ingresa sus nuevos valores y guarda. 4. El sistema valida los datos ingresados, en caso de haber error se le avisa al actor y se le permite corregirlos. 5. El sistema almacena en la base de datos los nuevos valores. 	
Flujo alternativo: <ul style="list-style-type: none"> • En caso de que el actor decida cancelar el registro. <ul style="list-style-type: none"> ○ Una vez el sistema despliega un formulario con los campos necesarios de un usuario, muestra un botón para cancelar el registro. ○ El usuario pulsa el botón "Cancelar". ○ El sistema retorna a la vista con el listado de rutas. 	
Postcondiciones: Los datos han sido agregados satisfactoriamente en la base de datos.	

Tabla 12. Plantilla de casos de uso - Crear Ruta.

ID	CU 41
Nombre	Visualizar rutas
Autor	Javier Enmanuel Matus Widdy
Fecha	26/10/2020
Actores	Administrador, Supervisor
Prioridad	Alta
Frecuencia	Alta
Descripción: Permitir al actor visualizar las rutas.	
Precondiciones: Los actores deben de estar autenticados en el sistema y tener permisos de administrador o supervisor.	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor pulsa el botón de “Visualizar rutas”. 2. El sistema muestra una lista con los datos de las rutas registradas. 3. El actor podrá filtrar y seleccionar un registro para ver detalles. 	
Flujo alternativo: <ul style="list-style-type: none"> • En caso de que no se encuentren registros de rutas en la base de datos el listado mostrará vacío. 	
Postcondiciones: El actor observa la información respecto a las rutas registradas.	

Tabla 13. Plantilla de casos de uso - Visualizar Ruta.

ID	CU 42
Nombre	Listar Usuarios Ubicación y porcentaje.
Autor	Javier Enmanuel Matus Widdy
Fecha	26/10/2020
Actores	Administrador, Supervisor
Prioridad	Alta
Frecuencia	Alta
Descripción: Permitir al actor listar y visualizar todos los usuarios con su ubicación y porcentaje de batería.	
Precondiciones: Los actores deben de estar autenticados en el sistema y tener permisos de administrador o supervisor.	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor pulsa el botón de “Monitoreo” en el menú de opciones. 2. El sistema muestra una lista con los datos de los usuarios registrados. 3. El actor podrá filtrar y seleccionar un registro para ver más detalles. 	
Flujo alternativo: <ul style="list-style-type: none"> • En caso de que no se encuentren registros de usuarios en la base de datos el listado mostrará vacío. 	
Postcondiciones: El actor observa la información respecto a los usuarios registrados.	

Tabla 14. Plantilla de casos de uso - Listar Usuarios.

ID	CU 43
Nombre	Listar Trayectoria de Usuario por fecha.
Autor	Javier Enmanuel Matus Widdy
Fecha	26/10/2020
Actores	Administrador, Supervisor
Prioridad	Alta
Frecuencia	Alta
Descripción: Permitir al actor listar y visualizar el histórico de trayectoria del usuario.	
Precondiciones: Los actores deben de estar autenticados en el sistema y tener permisos de supervisor.	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor pulsa el botón de “Geo Rutas”. 2. El sistema muestra una lista con los datos de los usuarios registrados con acceso móvil. 3. El actor pulsa el botón de “Ver Trayectoria” en el usuario seleccionado. 4. El sistema muestra el mapa con la trayectoria y la distancia recorrida que ha registrado el usuario seleccionado el actor puede filtrar por fecha. 	
Flujo alternativo: <ul style="list-style-type: none"> • En caso de que no se encuentren registros de usuarios en la base de datos el listado se mostrará vacío. 	
Postcondiciones: El actor observa la información respecto al usuario registrado.	

Tabla 15. Plantilla de casos de uso - Listar Trayectoria de Usuario por fecha.

ID	CU 44
Nombre	Listar Actividades del Usuario.
Autor	Darwin Antonio Ruiz Herrera
Fecha	26/10/2020
Actores	Supervisor
Prioridad	Alta
Frecuencia	Alta
Descripción: Permitir al supervisor observar el total de actividades asignadas a los usuarios	
Precondiciones: Los actores deben de estar autenticados en el sistema y tener permisos de supervisor.	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor pulsa el botón de “Planificación” en el menú de opciones. 2. El sistema muestra un calendario con las actividades de los usuarios. 3. El actor podrá ver los detalles de cada una de las actividades asignadas a los usuarios. 	
Flujo alternativo: <ul style="list-style-type: none"> • En caso de que no se encuentren registros de usuarios en la base de datos el listado se mostrará vacío. 	
Postcondiciones: El actor observa la información respecto al usuario registrado.	

Tabla 16. Plantilla de casos de uso - Listar Actividades de Usuario.

ID	CU 45
Nombre	Registrar Clientes Nuevos
Autor	Darwing Antonio Ruiz Herrera
Fecha	26/10/2020
Actores	Supervisor
Prioridad	Alta
Frecuencia	Alta
Descripción: Permitir al supervisor registrar clientes nuevos.	
Precondiciones: Los actores deben de estar autenticados en el sistema y tener permisos de supervisor/promotor y/o gestor.	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor pulsa el botón de “Clientes”. 2. El sistema muestra un formulario con los datos requeridos por la institución. 3. El actor pulsa el botón de “Guardar” el sistema deberá guardar los datos. 	
Flujo alternativo: <ul style="list-style-type: none"> • El sistema mostrará un mensaje de error indicando que no ha sido posible realizar el registro. 	
Postcondiciones: El actor observa la información respecto al cliente registrado.	

Tabla 17. Plantilla de casos de uso - Registrar Cliente.

ID	CU 46
Nombre	Editar Clientes
Autor	Darwing Antonio Ruiz Herrera
Fecha	26/10/2020
Actores	Supervisor
Prioridad	Alta
Frecuencia	Alta
Descripción: Permitir al supervisor Editar clientes.	
Precondiciones: Los actores deben de estar autenticados en el sistema y tener permisos de supervisor.	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor pulsa el botón de “Clientes”. 2. El actor pulsa el botón de “Editar Cliente” 3. El sistema muestra un formulario con los datos anteriormente registrados. 4. El actor pulsa el botón de “Guardar” el sistema deberá actualizar los datos. 	
Flujo alternativo: <ul style="list-style-type: none"> • El sistema mostrará un mensaje de error indicando que no ha sido posible actualizar el registro. 	
Postcondiciones: El actor observa la información respecto al cliente registrado.	

Tabla 18. Plantilla de casos de uso - Editar Cliente.

ID	CU 47
Nombre	Visita de Cliente.
Autor	Darwin Antonio Ruiz Herrera
Fecha	26/10/2020
Actores	Supervisor, Administrador
Prioridad	Alta
Frecuencia	Alta
Descripción: Permitir al supervisor o administrador verificar la visita de clientes realizadas desde la aplicación móvil.	
Precondiciones: Los actores deben de estar autenticados en el sistema	
Flujo Normal: <ol style="list-style-type: none"> 4. El actor pulsa el botón de “Visitas de Clientes” en el menú de opciones. 5. El sistema muestra una tabla con los clientes visitados a la fecha, y poder filtrar por fecha y usuario. 6. El actor podrá ver los detalles de cada una de las visitas realizadas. 	
Flujo alternativo: <ul style="list-style-type: none"> • En caso de que no se encuentren registros de usuarios en la base de datos el listado se mostrará vacío. 	
Postcondiciones: El actor observa la información respecto al usuario registrado.	

Tabla 19. Plantilla de casos de uso - Visita de Clientes.

ID	CU 48
Nombre	Reasignar Actividades
Autor	Darwing Antonio Ruiz Herrera
Fecha	26/10/2020
Actores	Supervisor
Prioridad	Alta
Frecuencia	Alta
Descripción: Permitir al supervisor la opción de reasignar una actividad a un nuevo usuario.	
Precondiciones: Los actores deben de estar autenticados en el sistema y tener permisos de supervisor.	
Flujo Normal: <ol style="list-style-type: none"> 1. El actor pulsa el botón de “Planificación” en el menú de opciones. 2. El sistema muestra un calendario con las actividades de los usuarios. 3. El actor podrá ver los detalles de cada una de las actividades asignadas a los usuarios. 4. El actor selecciona la actividad a reasignar 5. En el apartado de Usuario Asignado, selecciona al nuevo usuario 6. El actor pulsa el botón de “Guardar” el sistema deberá actualizar los datos. 	
Flujo alternativo: <ul style="list-style-type: none"> • En caso de que no se pueda reasignar la actividad el sistema deberá notificar al usuario de que ha ocurrido un error. 	
Postcondiciones: El actor observa la información respecto al usuario registrado.	

Tabla 20. Plantilla de casos de uso - Reasignar Actividades de Usuario.

8.2 Fase 2: Diseño de la solución

8.2.1 Diagrama de Actividades

A continuación, se presentan algunos diagramas de actividades principales llevadas a cabo durante la interacción de un usuario y el sistema con el fin de detallar el flujo de trabajo de los procesos involucrados en dichas actividades; por ejemplo, la actividad de Inicio de sesión, Administrar, monitorear, etc.

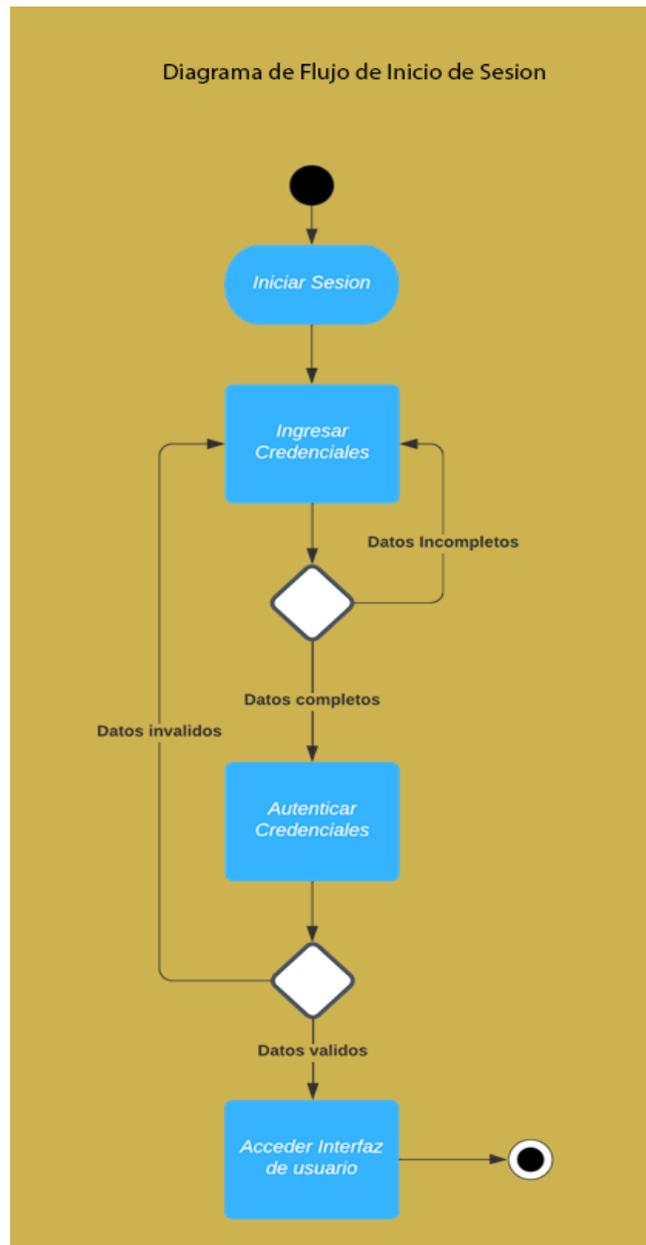


Figura 7. Diagrama de actividades Iniciar Sesión.

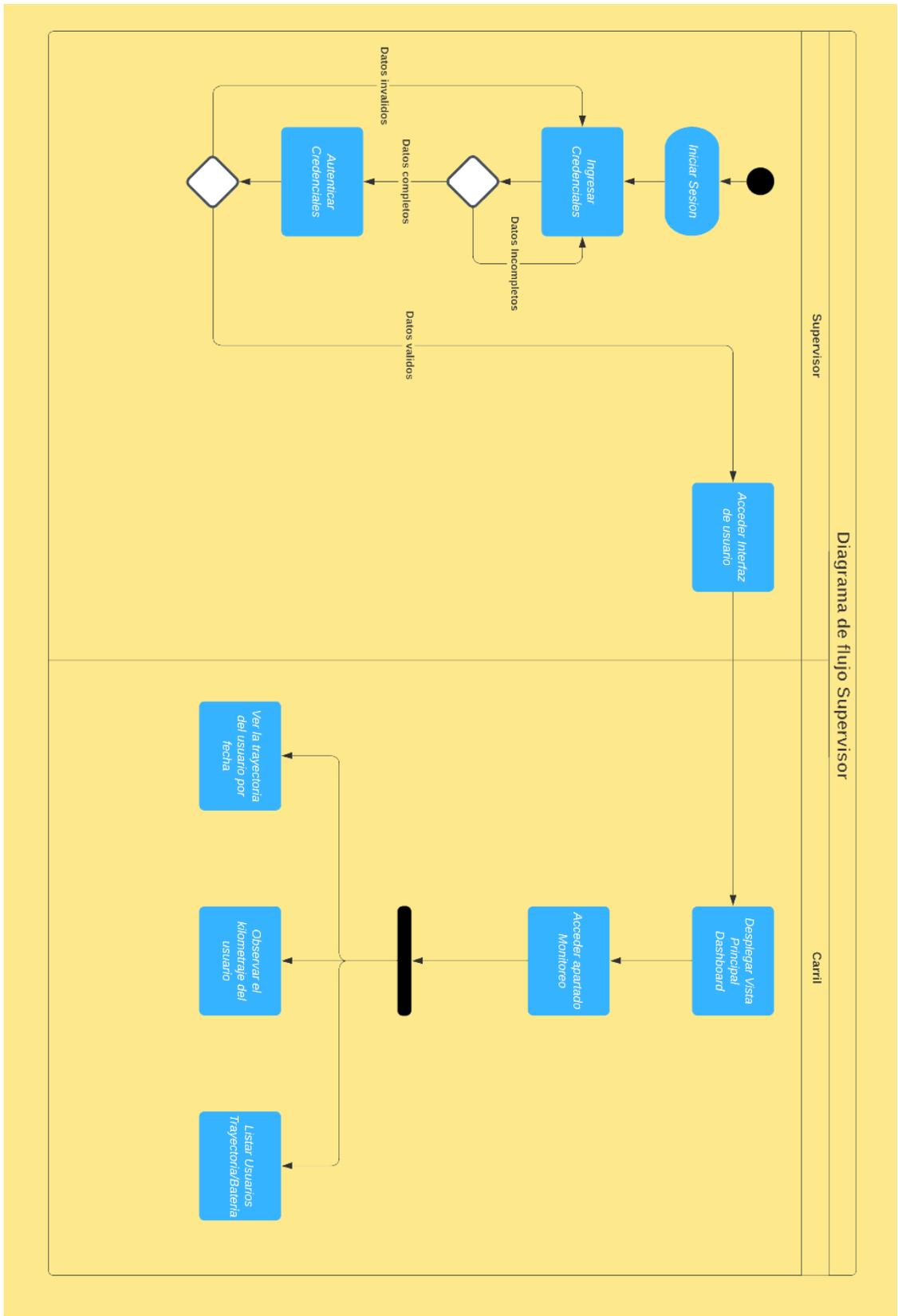


Figura 8. Diagrama de actividades General - Supervisor.

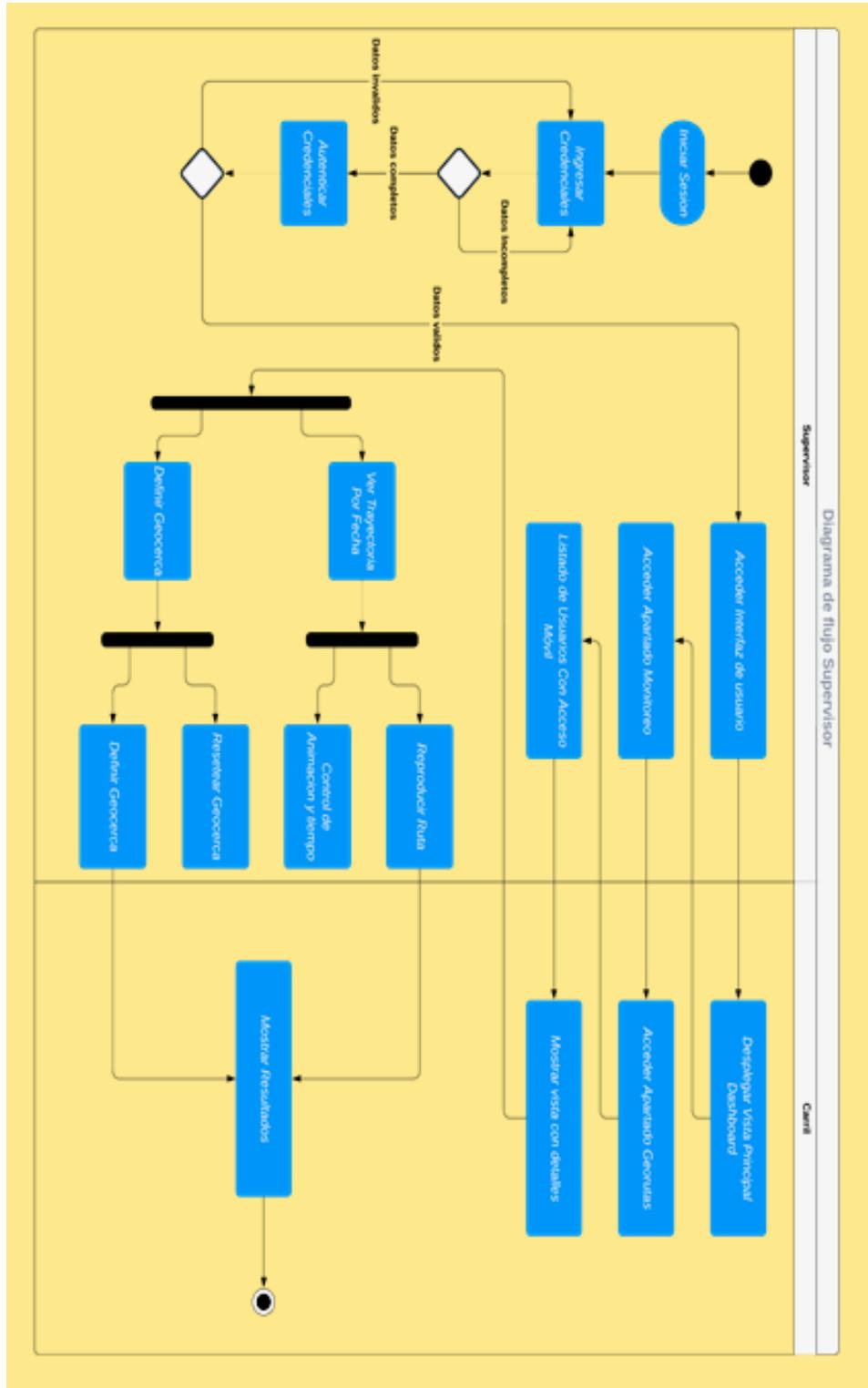


Figura 9. Diagrama de actividades Monitoreo - Geo Rutas - Supervisor.

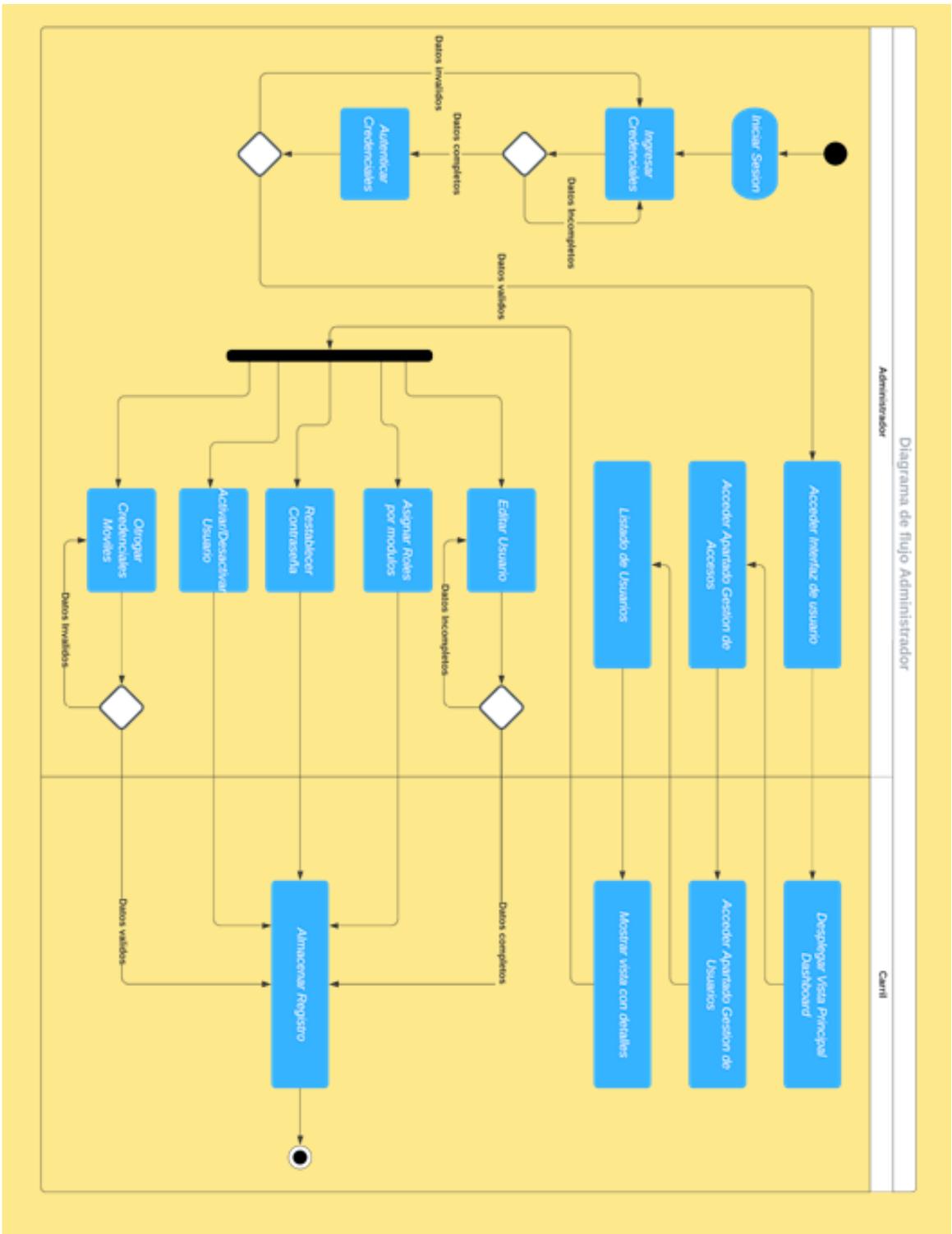


Figura 10. Diagrama de actividades Gestión de Acceso - Administración de Usuario - Administrador.

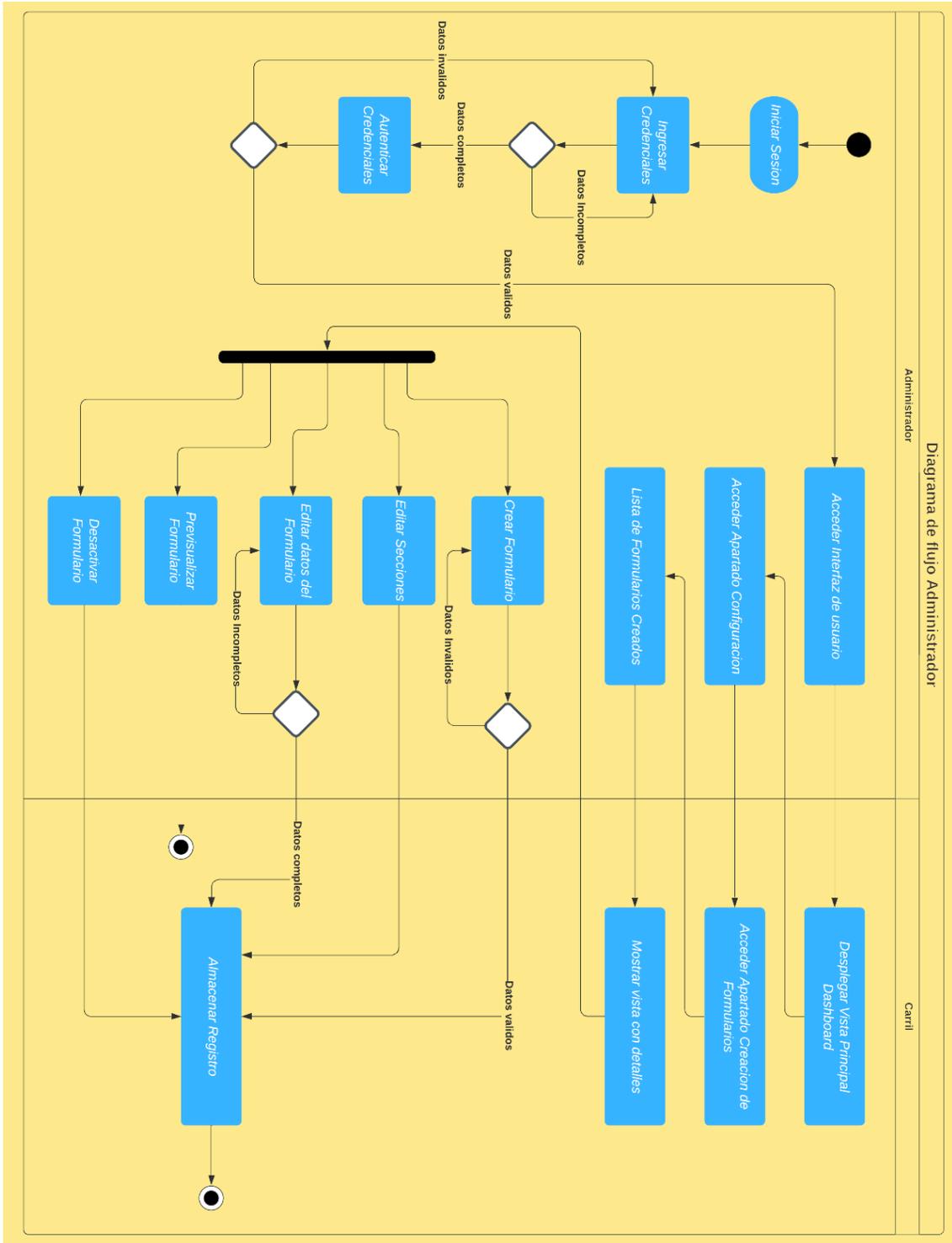


Figura 11. Diagrama de actividades Configuración - Creación de Formularios - Administrador.

8.2.2 Diagrama de Estados

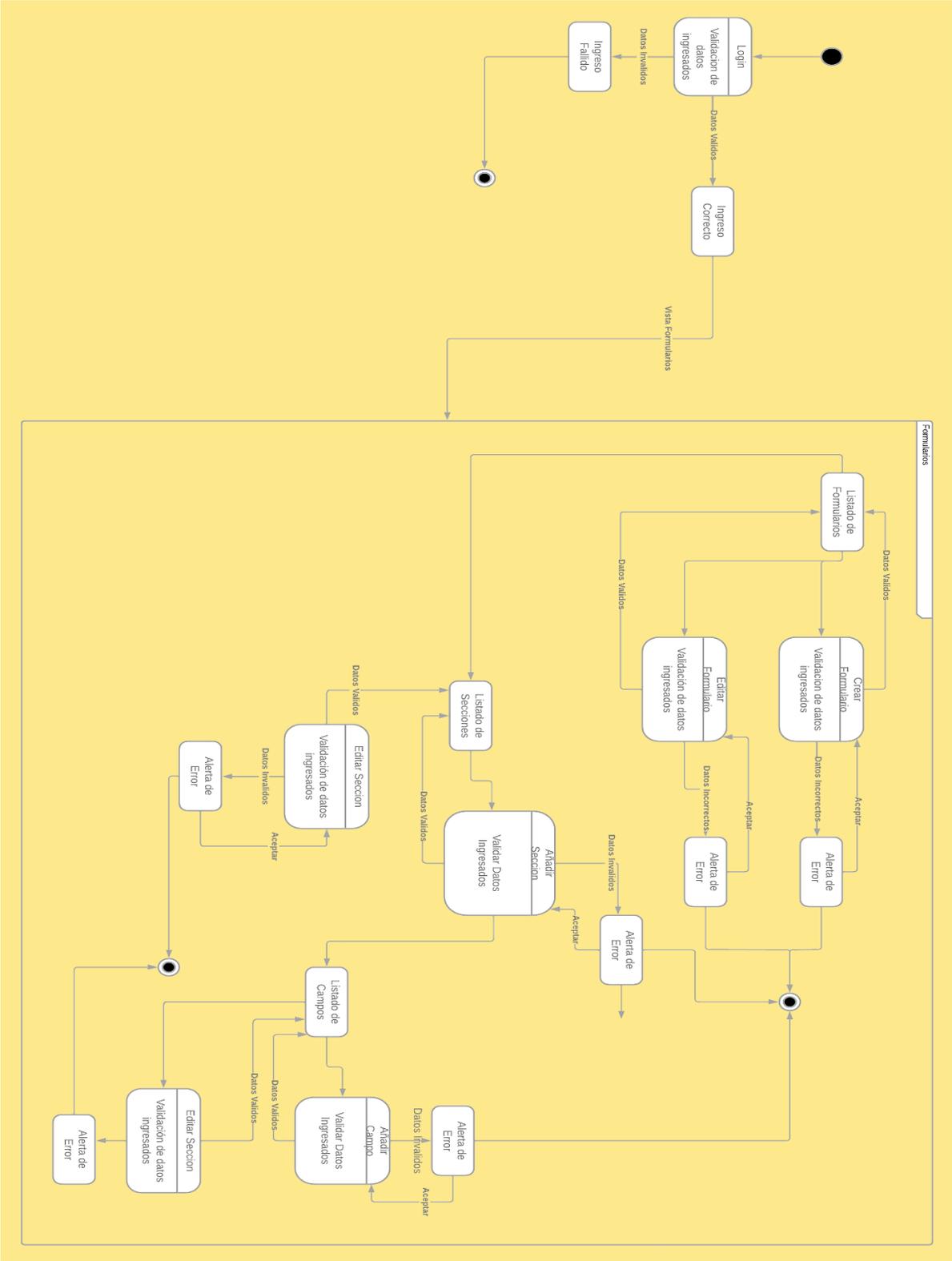


Figura 12. Diagrama de estados - Formularios.

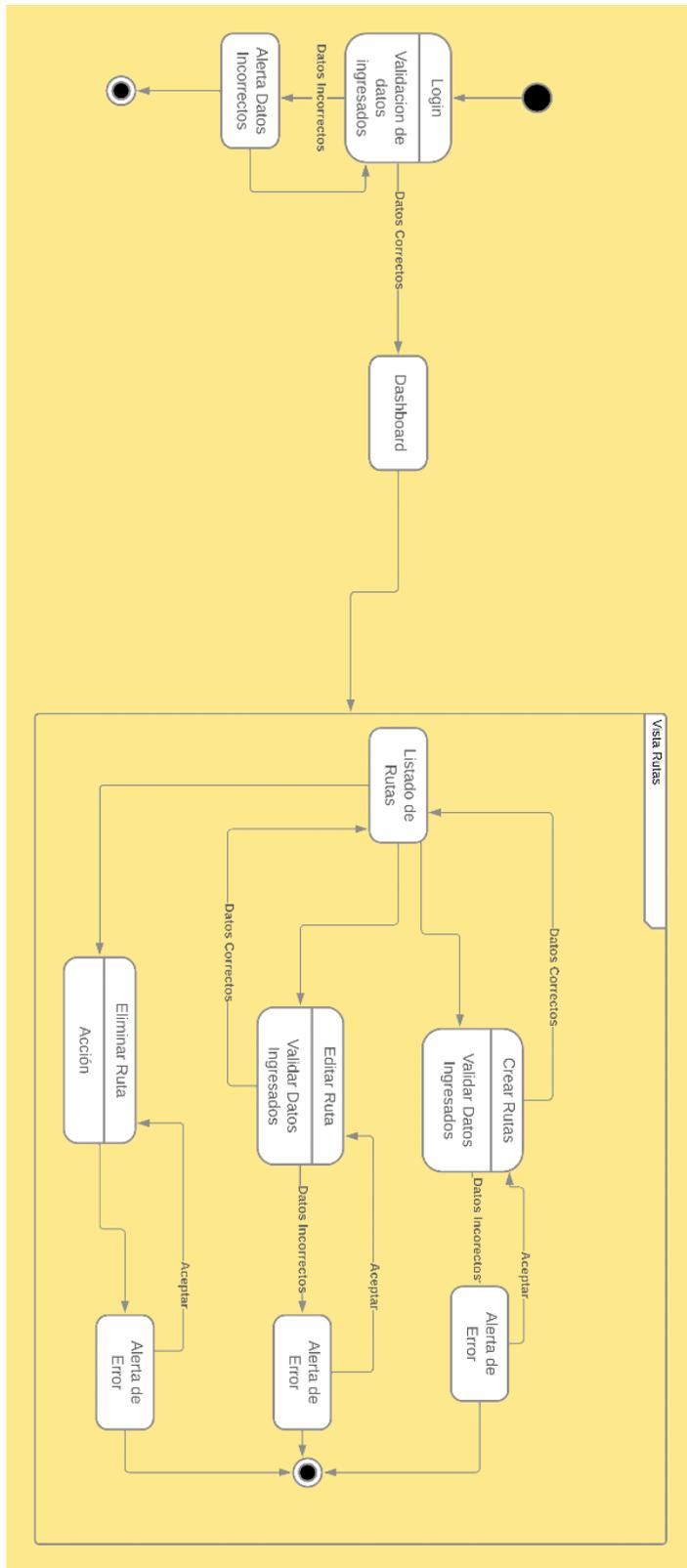


Figura 13. Diagrama de estados - Rutas.

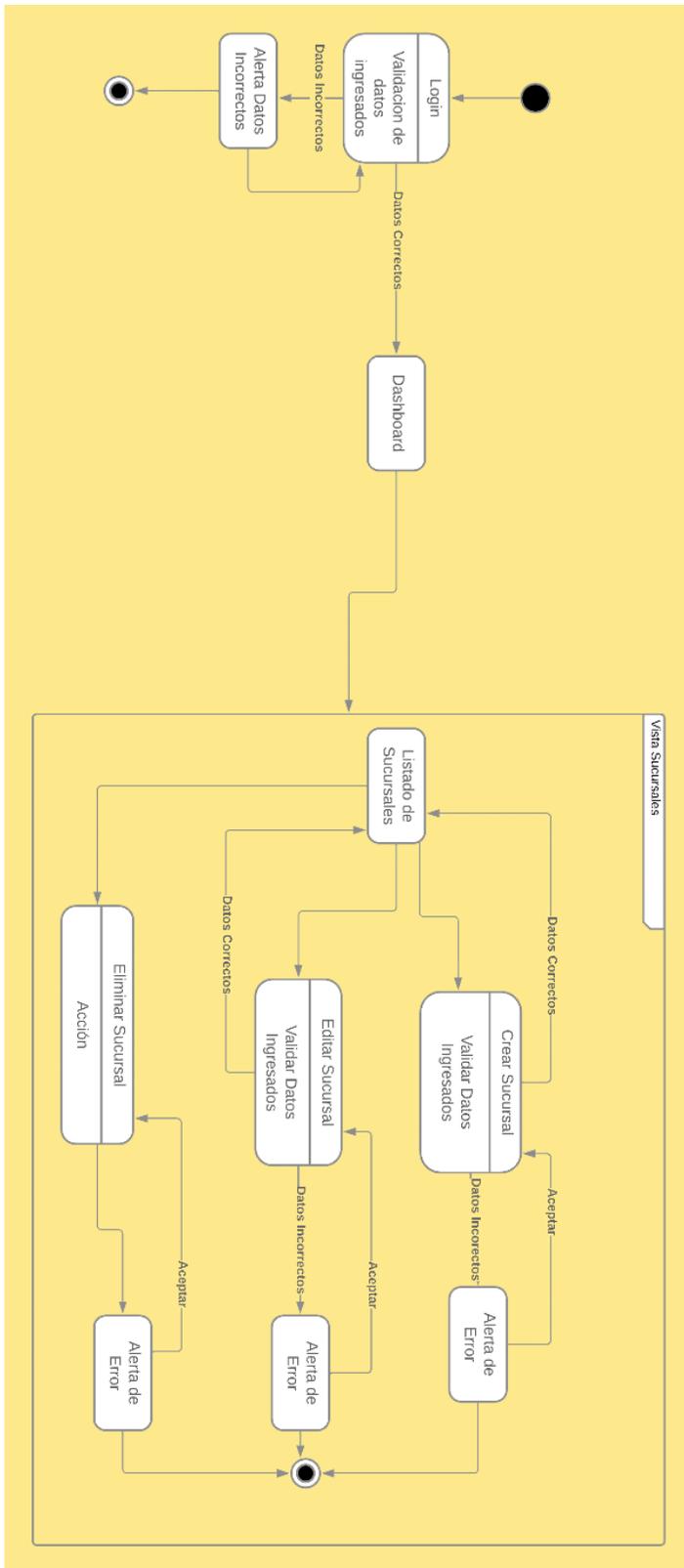


Figura 14. Diagrama de estados - Sucursales.

8.2.3 Modelo de Navegación

Este modelo tiene como objetivo dar una idea de la organización con respecto a los módulos jerárquicamente y su interacción con cada uno de los diferentes tipos de usuarios. A continuación, mostraremos los modelos de navegación de los usuarios con roles de administrador, supervisor, promotor y gestor.

Modelo de Navegacion - Administrador

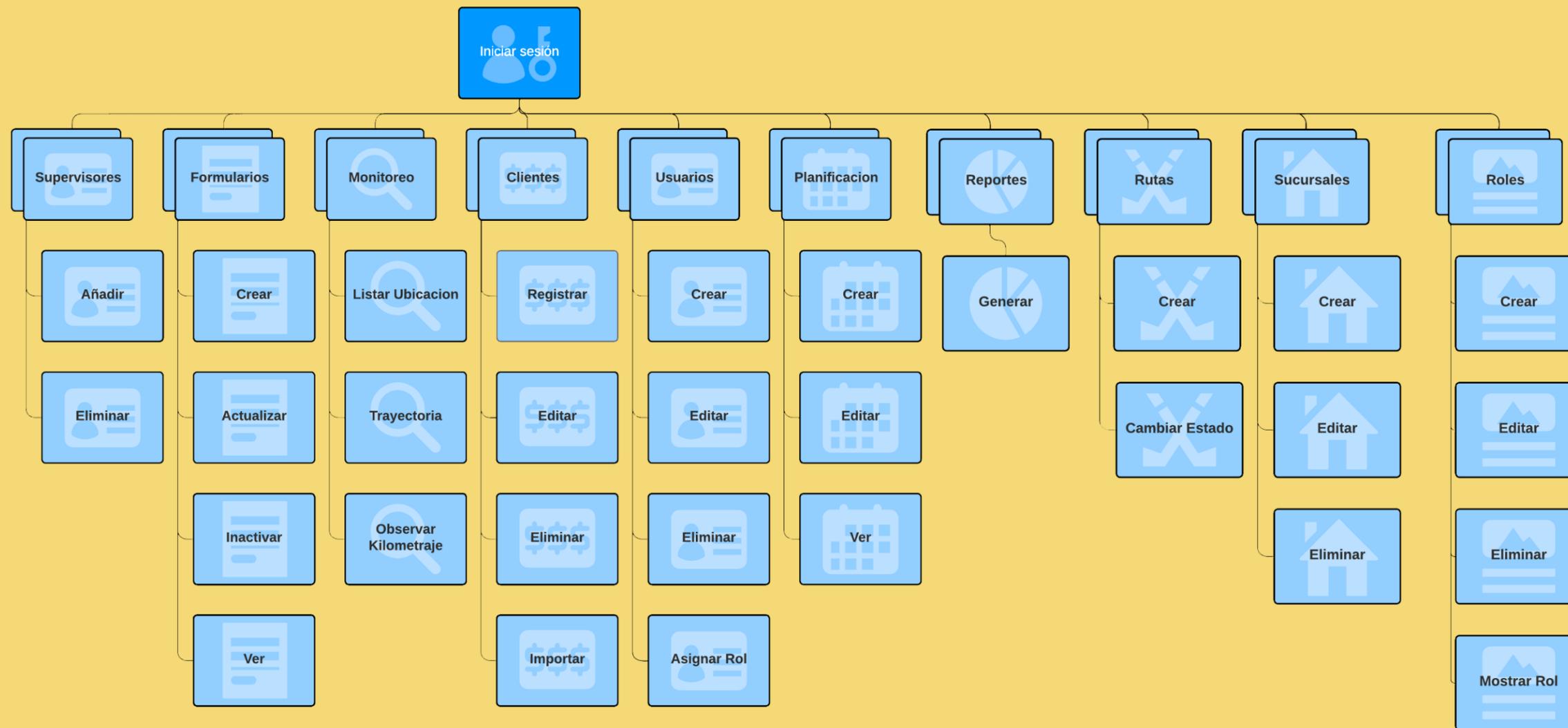


Figura 15. Modelo de navegación – Administrador Sistema Web.



Figura 16. Modelo de navegación – Supervisor Sistema web.

8.2.4 Modelo Relacional

En este apartado se presenta la estructura de la base de datos del sistema elaborada haciendo uso de la función Diseñador proporcionada por el sistema gestor de base de datos PHPMyAdmin, la cual entre sus capacidades está el de poder diseñar y visualizar los diagramas de una base de datos de preferencia de una manera más amigable e intuitiva.

Cabe mencionar, por motivos de legibilidad y espacio en la documentación se ha optado por dividir este diagrama en 3 partes que representan los módulos esenciales del sistema

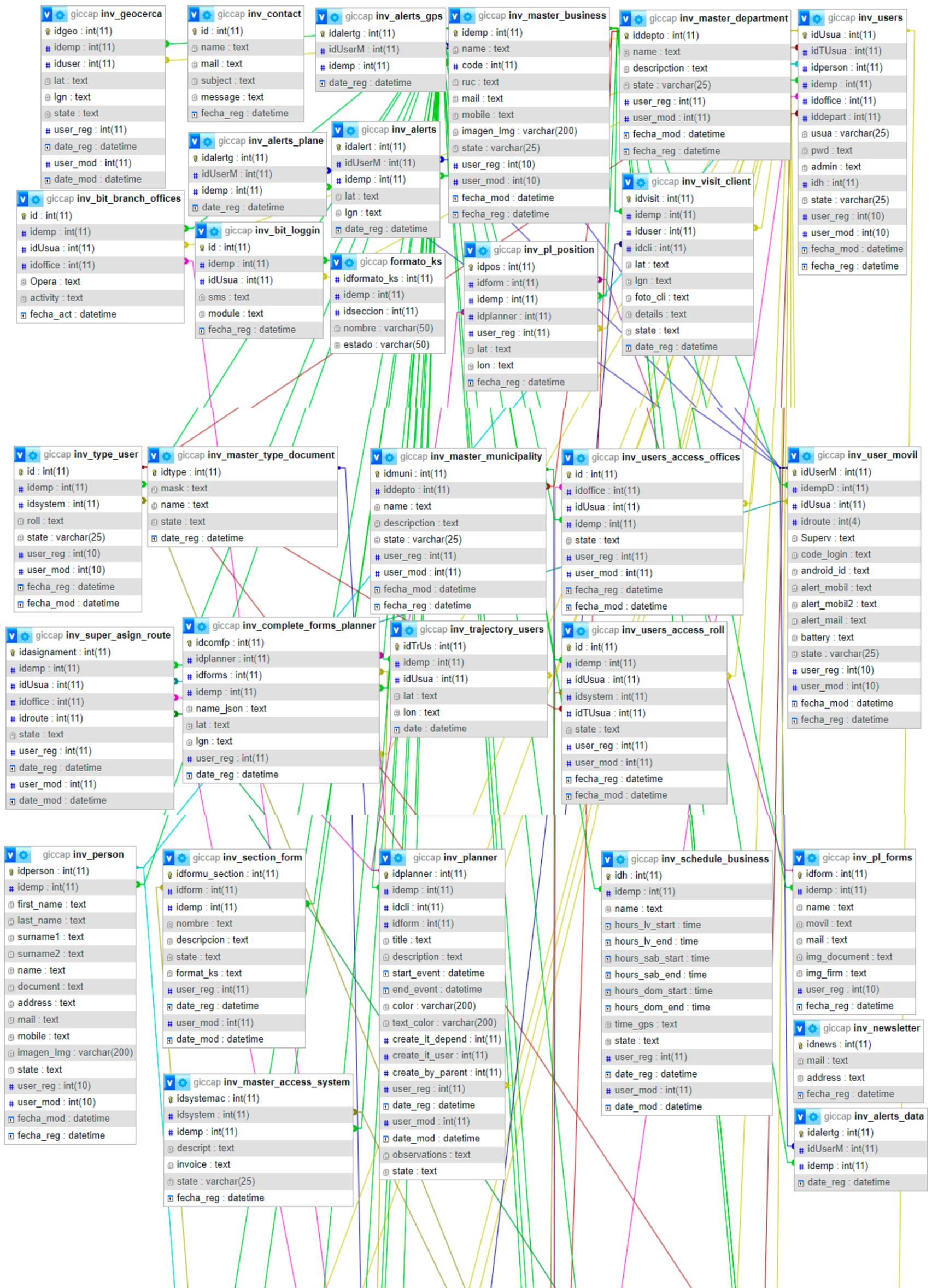


Figura 17. Modelo Relacional.

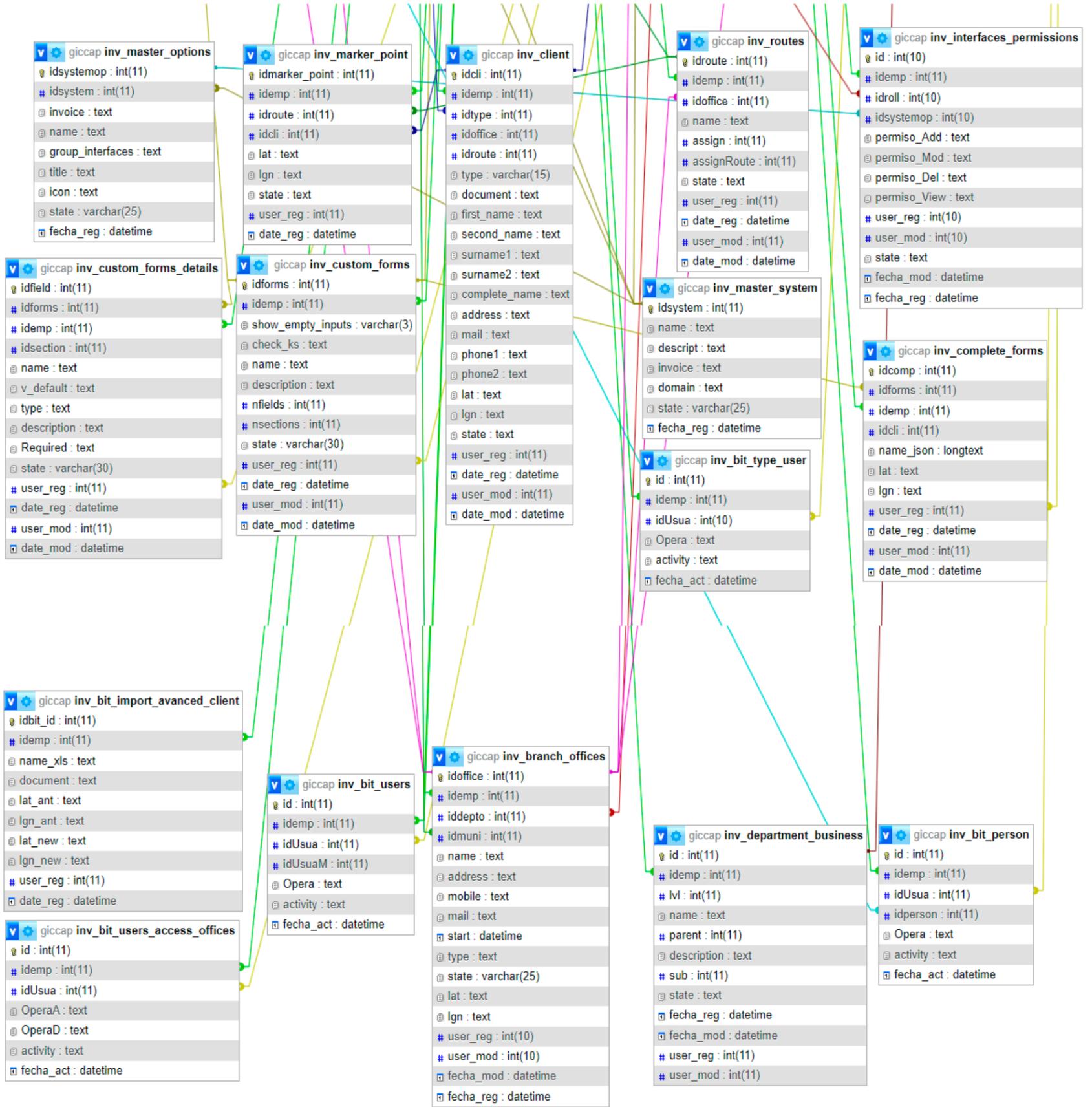


Figura 18 Modelo Relacional

8.2.5 Diseño Arquitectónico

El sistema cuenta con una arquitectura MVC, la cual es uno de varios modelos de infraestructura para una aplicación web sugeridos para lograr separar la interfaz de usuario de la funcionalidad y el contenido de información. No obstante, se hace uso de un modelo MVC distinto al ser este proveído por el modo de programación en PHP.

Primeramente, el usuario se encarga de realizar una petición por ruta a través del sistema con un navegador accesible, a continuación, esta petición por ruta pasa al controlador; después, el controlador se encarga de gestionar el acceso al modelo y a la vista coordinando el flujo de información entre estos, en caso que el controlador solicite información al modelo, este que tiene la lógica de negocio de la base de datos realiza la consulta y devuelve al controlador la información solicitada, de manera que el controlador pueda mandar la información a una vista y esta vista pueda mostrarle al usuario lo solicitado.

El siguiente diagrama presenta el flujo y otros componentes que intervienen en la arquitectura MVC.

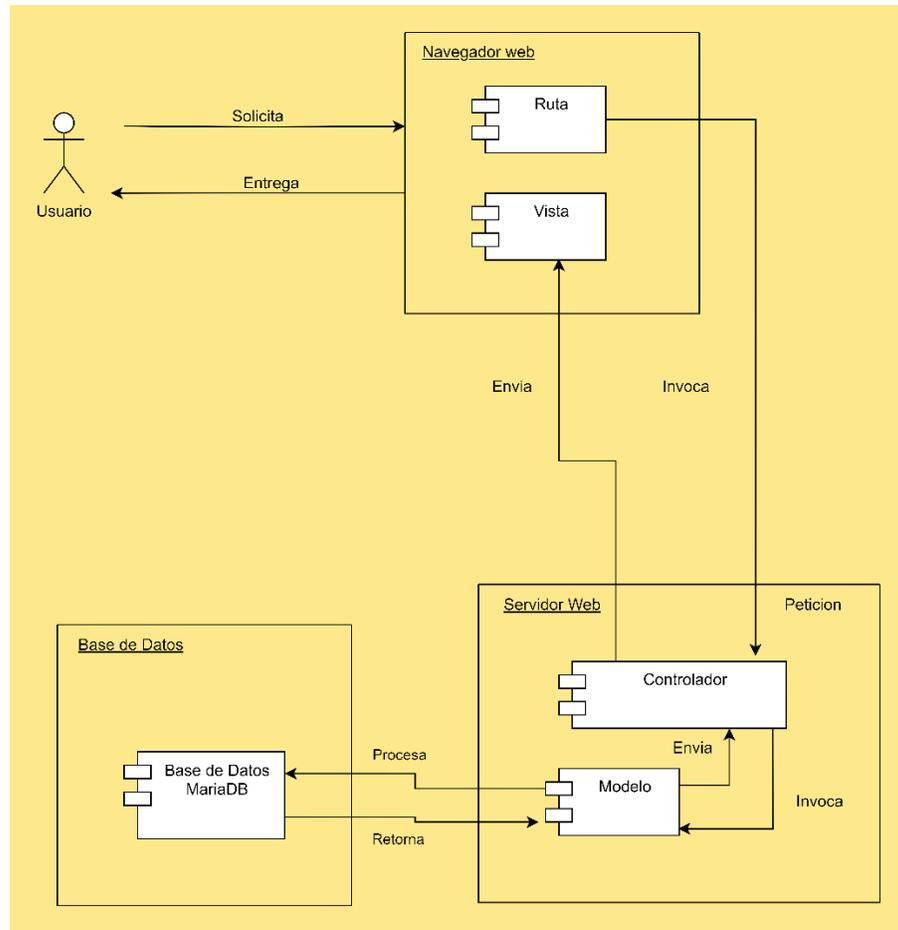


Figura 19. Diseño arquitectónico del sistema.

8.2.6 Diseño de interfaces gráficas de usuario (GUI)

8.2.6.1 Diseño de la interfaz

Para el desarrollo de las distintas interfaces de usuarios se estableció el siguiente esquema o esqueleto, el mismo se puede apreciar en la figura 17. Para más detalle se definirán los elementos.

- **Encabezado:** En esta área se encuentra una barra horizontal que contiene el login del sistema, además de poder desplegar iconos característicos de sitios adaptativos a diferentes tipos de dispositivos.
- **Menú de navegación:** Se ubica de manera vertical en la sección lateral izquierda de todas las interfaces. En esta se encuentran todas las opciones disponibles a los usuarios, además de la información del usuario que haya iniciado sesión. Cabe mencionar, este menú variará respecto al tipo de usuario.
- **Contenido:** En esta sección se ubicará todo el contenido respectivo a la opción seleccionada. Aquí se podrá encontrar listados de rutas, sucursales, usuarios, etc.; de igual manera los distintos formularios para el registro o edición de información, las vistas previas de reportes y otros. También, en la parte superior derecha se hace uso de indicadores de posición de tipo BreadCrumbs, o migas de pan, además de un título característico de la ubicación en que se encuentra.
- **Pie de Página:** En este punto se hace referencia al nivel de patentes y derechos reservados.



Figura 20. Estructura de la interfaz de usuario Web.

En el diseño de las interfaces de usuario se hizo uso de las recomendaciones (estándares) por parte de la W3C, por mencionar algunas como accesibilidad web, html, css, DOM, https. Igualmente se hizo uso del lenguaje de programación javascript y otras librerías.

Para cumplir con los estándares se utilizó el framework bootstrap para agregar adaptabilidad al sitio, lo que hoy en día se conoce como responsive-design y permite que el sitio pueda ser accedido por cualquier tipo de dispositivo móvil (Celular, Tablet o Laptop) o Computadora de escritorio.

8.2.6.2 Diseño Estético

También llamado diseño gráfico, consiste en el esfuerzo artístico que complementa los aspectos técnicos de la ingeniería web, dando el toque atractivo

a la WebApp. Para realizar un diseño estético efectivo primeramente se establece el tipo de usuario que hará uso del sistema.

Teniendo como usuarios meta los colaboradores, sé ha tomado la decisión de buscar una interfaz clara, combinando colores como  (#007BFF) azul,  (#FFFFFF) blanco,  (#F4F6F9) gris, etc.

Para una mejor valoración en cuanto al diseño estético se han tomado como referencias dos cuestiones fundamentales en el diseño, Cuestiones de la plantilla y Cuestiones de diseño gráfico.

En cuanto a las cuestiones de la plantilla, indican recomendaciones a seguir en cuanto a la organización del contenido en la plantilla, dividiéndose en:

- **No temerle al espacio vacío.**

En este punto se ha dejado espacio disponible en todas las interfaces para no tener que sobrecargar las páginas atentando en un criterio de usabilidad.

- **Resaltar el contenido.**

El diseño del sistema o WebApp está orientado a cumplir con las funcionalidades establecidas por parte de los usuarios que dispondrán de este.

- **Organizar los elementos de plantilla.**

En efecto los elementos se encuentran organizados en dirección de esquina superior izquierda a esquina inferior derecha de todo dispositivo.

- **Agrupar navegación, contenido y función geográficamente en la página.**

En cuanto a este punto las interfaces de usuario cuentan con similitudes por lo que el usuario podrá aprender con rapidez el uso del sitio.

- **No extender el sitio con la barra de desplazamiento.**

El sistema busca la reducción del uso de la barra de desplazamiento lateral al disminuir el espacio ocupado por todos los formularios.

➤ **Considerar la resolución y el tamaño de la ventana.**

Durante el desarrollo del diseño del sitio se ha tomado en cuenta la resolución estándar de pantallas 1366x768.

Asimismo, tomando en cuenta las cuestiones de diseño gráfico se ha tomado en consideración los aspectos de presentación, siendo estos los siguientes:

➤ **Tipografía**

Se establecieron la fuente Source Sans Pro

El tamaño de la fuente varía en cuanto a referencia de títulos, subtítulos, etiquetas, botones. Siendo la variación entre 14px y 40px.

➤ **Color**

Colores asociados a la medicina siendo estos: blanco, azul, gris claro, negro, etc.

Se ha tomado como referencia la paleta de colores de material design.

➤ **Layouts**

Se hace uso del sistema enmallado o de rejilla (Grid) que provee el framework Bootstrap para permitir el diseño adaptativo a dispositivos.

➤ **Imágenes**

Según el estándar de la W3C se han hecho uso de imágenes en formato PNG (Portable Network Graphics) al ser uno de los formatos más utilizados en la web.

➤ Navegabilidad

El contenido al ser adaptativo cuenta con suficiente accesibilidad.

Se han implementado APIs externas como extensiones y proveer a los usuarios una mejor satisfacción en la navegabilidad.

8.2.6.3 Diseño de Contenido

El contenido del sistema se encuentra dividido en cuanto a los tipos de usuarios, administradores, supervisor, promotor y gestor. Cada uno cuenta con interfaces de usuarios distintos partiendo por el menú de navegación.

Aplicación Web



Formularios

The screenshot shows a web browser window with a yellow-themed interface. The browser's address bar is empty. The page title is "Formularios". On the left, there is a vertical navigation menu with the following items: Home (highlighted in blue), Configuración, Gestion de acceso, Clientes, Planificación, Monitoreo, Reportes, Cambios de Credenciales, and Cerrar Sesión. The main content area is titled "Listado de Formularios" and contains a table with the following data:

Fecha de Creacion	Nombre	Descripcion	Nº Secciones	Nº Campos	Acciones
25/09/2023	Lorem	Lorem	11	50	   
25/09/2023	Lorem	Lorem	12	23	   
25/09/2023	Lorem	Lorem	12	28	   
25/09/2023	Lorem	Lorem	5	60	   

Figura 22. Vista Lista de Formularios. Realizada con herramienta: NinjaMock

Aplicación Móvil



Figura 23. Vista Menú.

8.3 Fase 3: Codificación del Software

El propósito de esta fase es mostrar el proceso de llevar a código fuente los modelos representados en la fase de diseño, haciendo uso del patrón de arquitectura MVC implementado en el lenguaje PHP, en el cual se separaron los datos de la aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos: modelos, vistas y controladores, respectivamente.

```

Add_Forms.php X
<div class="container-fluid">
  <!-- left column -->
  <div class="col-md-12">
    <!-- general form elements -->
    <div class="card card-primary">
      <div class="card-header">
        <h3 class="card-title">Formularios</h3>
      </div>
      <!-- /.card-header -->
      <!-- form start -->
      <form role="form" id="newForms" autocomplete="off">
        <input type="hidden" name="idemp" id="idemp" value="$emp_id">
        <div class="card-body">
          <div class="col-md-12">
            <label for="nameForm">Nombre del formulario</label>
            <div class="input-group">
              <div class="input-group-prepend">
                <span class="input-group-text"><i class="fas fa-f
              </div>
              <input type="text" class="form-control" name="nameFor
                placeholder="Ingrese Un Nombre">
            </div>
          </div>
          <div class="col-md-12">
            <label for="descriptionForm">Descripcion</label>
            <div class="input-group">
              <div class="input-group-prepend">
                <span class="input-group-text"><i class="fas fa-b
              </div>
              <!-- <input type="text" class="form-control" id="desc
            <textarea class="form-control" name="descriptionForm"
              cols="30" rows="10" placeholder="Ingrese una Desc
            </div>
          </div>
        </div>
      </form>
    </div>
  </div>
</div>

```

Figura 24. Vista Agregar Formulario.

```

formController.php X
<?php
require_once('../model/Form.model.php');

if (isset($_POST['opt'])) {
    $model = new Forms();
    $opt = $_POST['opt'];
    switch ($opt) {
        case 'GetForms':
            if (isset($_POST['idemp'])) {
                $idemp = $_POST['idemp'];
                $dataBD = $model->getForms($idemp);
                if ($dataBD == "") {
                    $data = array(
                        "sEcho" => 1,
                        "iTotalRecords" => "0",
                        "iTotalDisplayRecords" => "0",
                        "Data"=>[]);
                }
            } else {
                $data = $dataBD;
            }

            echo json_encode($data);
            break;
        case 'insertForms':
            if (isset($_POST['name']) && isset($_POST['description']) && isset($_POST['idemp'])) {
                date_default_timezone_set('America/Managua');
                $name = $_POST['name'];
                $description = $_POST['description'];
                $idemp = $_POST['idemp'];
                $state = "Activo";
                $show = $_POST['show'];
                // cambiar por el valor de la sesion del usuario logueado
                $user = 1;
                $fecha = date('Y-m-d H:i:s');
                echo json_encode($model->insertForms($name,$description,$idemp,$user,$fecha,$state,$show));
            }
    }
}

```

Figura 25. Controlador del Formulario.

```
Form.model.php X
// get details forms data
public function get_Data_Details_Form($idsection,$idemp,$idform)
{
    $data = [];
    $con = conexion::getConexion();
    $sql = "SELECT * FROM `inv_custom_forms_details` WHERE idsection = :SECTION AND idforms = :IDFORM";
    $resultado = $con->prepare($sql);
    $resultado->execute( array(":IDEMP" => $idemp,":SECTION" => $idsection,":IDFORM" => $idform));
    $count = $resultado->rowCount();
    if ($count > 0) {
        $data = $resultado->fetchAll(PDO::FETCH_ASSOC);
    }
    else{
        $data = "";
    }
    return $data;
}
// insertar form
public function insertForms($name,$description,$idemp,$user_reg,$dateReg,$state,$show)
{
    $result = [];
    $con = conexion::getConexion();
    $sql = "INSERT INTO `inv_custom_forms`(`idemp`, `name`, `description`, `state`, `user_reg`, `dateReg`, `show`) VALUES (:IDEMP, :NAME, :DESCRIPTION, :STATE, :USER_REG, :DATE_REG, :SHOW)";
    $resultado = $con->prepare($sql);
    $resultado->execute( array(":IDEMP" => $idemp,":NAME" => $name,":DESCRIPTION" => $description,":STATE" => $state,":USER_REG" => $user_reg,":DATE_REG" => $dateReg,":SHOW" => $show));
    $count = $resultado->rowCount();
    if ($count > 0) {
        $result["mensaje"]="Success";
    }
    else{
        $result["mensaje"]="Error";
    }
    return $result;
}
```

Figura 26. Modelo del Formulario.

8.4 Fase 4: Pruebas

A continuación, en esta sección se mostrarán una serie de pruebas funcionales basadas en los requerimientos modelados en los casos de uso, como también pruebas de seguridad a casos como el acceso de un usuario con datos incorrectos y el acceso por rutas sin tener las credenciales necesarias. Dichas pruebas se realizaron al proyecto con el objetivo de determinar posibles errores en el código, corregirlos y entregar un producto final que cumpla con la calidad de software esperado.

Además, se llevó a cabo una evaluación cuantitativa de otros atributos clave relacionados con la calidad del software, centrándose en la usabilidad y la experiencia del usuario. Esto se logró mediante la aplicación de técnicas de evaluación conocidas como "pruebas de usabilidad". Durante este proceso, se compararon el escenario de trabajo tradicional previo a la implantación del sistema con el uso del sistema desarrollado.

8.4.1 Pruebas Funcionales del Sistema

Prueba de Inserción de un nuevo formulario

ID	PF1
Nombre	Agregar un nuevo Formulario
Perfil Usuario	del Usuario autenticado
Función Probar	Por FormsController@InsertForm()
Objetivo	Detectar errores en el método del controlador al registrar los datos de un nuevo Formulario.
Descripción	Se hizo el registro de un Formulario y se verificó que el método funcionara correctamente.
Criterio de Éxito	Se muestra un mensaje diciendo que se ha registrado satisfactoriamente.
Criterio de Falla	Se muestra un mensaje de error en los campos del formulario o al momento de presionar el botón guardar.
Precondiciones	<ul style="list-style-type: none"> • El usuario debe de estar autenticado en el sistema. • Ingresar en la vista de Formularios. • Presionar el botón de “Agregar Formulario” • Llenar el formulario de registro con la información requerida.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario inicia sesión en el sistema. 2. Se dirige al módulo de configuración y presiona el botón creación de formularios. 3. El usuario es redireccionado a la vista de Formularios y a continuación presiona el botón Nuevo Formulario. 4. Es redireccionado a la vista con el formulario para el nuevo registro. 5. Se ingresa la información solicitada. 6. Presiona el botón guardar.
Post Condiciones	Los datos son almacenados exitosamente en la base de datos.

Tabla 21. Pruebas Funcionales - inserción de un Nuevo Formulario.

Prueba de obtención de todos los formularios creados

ID	PF2
Nombre	Obtener Todos los Formularios creados
Perfil	<i>del</i>
Usuario	Usuario autenticado
Función	<i>Por</i>
Probar	FormsController@getForms()
Objetivo	Detectar errores en el método del controlador al obtener los datos de los formularios.
Descripción	Se hizo ingreso a la vista y se verificó que el método funcionara correctamente.
Criterio de Éxito	Se muestran todos los formularios registrados.
Criterio de Falla	No se muestra ninguno de los formularios registrados.
Precondiciones	<ul style="list-style-type: none"> • El usuario debe de estar autenticado en el sistema. • Ingresar en la vista de Formularios.
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario inicia sesión en el sistema. 2. Se dirige al módulo de configuración y presiona el botón creación de formularios. 3. El usuario es redireccionado a la vista de Formularios. 4. Se muestran todos los formularios registrados.
Post Condiciones	Los datos son almacenados exitosamente en la base de datos.

Tabla 22. Pruebas Funcionales - Obtención de todos los Formularios.

8.4.2 Pruebas de Seguridad del Sistema

ID	PS1
Nombre	Iniciar Sesión
Perfil del Usuario	Sin autenticar
Función a Probar	Por LoginController@ctrLogginUser()
Objetivo	Detectar errores en el método del controlador al intentar iniciar sesión con un usuario que no esté autenticado.
Descripción	Se puso a prueba el formulario de inicio de sesión al ingresar credenciales no autenticadas.
Criterio de Éxito	Se muestra un mensaje diciendo "Las credenciales están incorrectas".
Criterio de Falla	El usuario no autenticado logra ingresar con datos incorrectos.
Precondiciones	<ol style="list-style-type: none"> 1. Acceder a la ruta (URL) establecida para la vista de inicio de sesión
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario accede a la ruta de inicio de sesión. 2. Se muestra la vista con el formulario de inicio de sesión. 3. Se ingresan datos no válidos en el formulario. 4. Presiona el botón Iniciar Sesión.
Post Condiciones	<ul style="list-style-type: none"> • El sistema no permite el acceso del usuario con los datos ingresados previamente.

Tabla 23. Pruebas de Seguridad - Iniciar Sesión.

ID	PS2
Nombre	Ingreso no autorizado
Perfil del Usuario	Usuario autenticado
Función a Probar	LoginController@ctrLogginUser()
Objetivo	Detectar errores en el método del controlador al intentar iniciar sesión con un usuario que no esté autenticado.
Descripción	Se puso a prueba el formulario de inicio de sesión al ingresar credenciales no autenticadas.
Criterio de Éxito	Se muestra un mensaje diciendo "Las credenciales están incorrectas".
Criterio de Falla	El usuario no autenticado logra ingresar con datos incorrectos.
Precondiciones	<ol style="list-style-type: none"> 1. Acceder a la ruta (URL) establecida para la vista de inicio de sesión
Flujo Normal	<ol style="list-style-type: none"> 1. El usuario accede a la ruta de inicio de sesión. 2. Se muestra la vista con el formulario de inicio de sesión. 3. Se ingresan datos no válidos en el formulario. 4. Presiona el botón Iniciar Sesión.
Post Condiciones	<ul style="list-style-type: none"> • El sistema no permite el acceso del usuario con los datos ingresados previamente.

Tabla 24. Pruebas de Seguridad - Ingreso no Autorizado.

8.4.3 Pruebas Unitarias del Sistema

Hemos llevado a cabo pruebas exhaustivas de obtención e inserción de formularios en nuestro proyecto utilizando PHPUnit. Estas pruebas han sido esenciales para garantizar que nuestros formularios funcionen de manera precisa y sin errores, contribuyendo así a la robustez y confiabilidad de nuestro sistema.

```
<?php
require_once __DIR__ . '/../vendor/autoload.php';
require_once '../controller/FormsController/FormController2.php';

use PHPUnit\Framework\TestCase;

class FormControllerTest extends TestCase
{
    /** @test */
    public function GetFormsTest()
    {
        $controller = new FormsController();
        $idemp = 1; // Reemplaza con el valor adecuado

        $result = $controller->getForms($idemp);
        $this->assertJson($result);
    }
    /**
     * @test
     */
    public function testInsertForms()
    {
        $controller = new FormsController();
        $name = 'Formulario de prueba';
        $description = 'Descripción de prueba';
        $idemp = 1;
        $show = 'Si';
        $user = 1;

        $result = $controller->insertForms($name, $description, $idemp, $user, date('Y-m-d H:i:s'));
        $this->assertJson($result); // Verificar que la salida sea un JSON válido
    }
}
```

Figura 27. Código de FormControllerTest.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
PS C:\xampp\htdocs\Giccap\test> ..\vendor\bin\phpunit FormControllerTest.php
PHPUnit 10.3.5 by Sebastian Bergmann and contributors.

Runtime:    PHP 8.2.4

..                                                2 / 2 (100%)

Time: 00:00.029, Memory: 6.00 MB

OK (2 tests, 2 assertions)
```

Figura 28. Resultado de FormControllerTest.

```

class SucursalControllerTest extends TestCase
{
    /** @test */
    public function getSucursales()
    {
        $controller = new SucursalController();
        $idemp = 1;

        $result = $controller→getSucursales($idemp);
        $this→assertJson($result);
    }
    /**
     * @test
     */
    public function testInsertSucursal()
    {
        $controller = new SucursalController();
        $name = 'Sucursal de prueba';
        $address = 'direccion de prueba';
        $idemp = 1;
        $mobile = '12345678';
        $mail = 'admin@admin.com';
        $idmuni = 1;
        $consecutive = 1;
        $user = 1;

        $result = $controller→insertSucursal($name, $address,$mobile,$mail,$idmuni,$consecutive,
        $this→assertJson($result); // Verificar que la salida sea un JSON válido
    }
}

```

Figura 29. Código SucursalControllerTest.

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS COMENTARIOS

PS C:\xampp\htdocs\6iccap\test> ..\vendor\bin\phpunit SucursalControllerTest.php
 PHPUnit 10.3.5 by Sebastian Bergmann and contributors.

Runtime: PHP 8.2.4

.. 2 / 2 (100%)

Time: 00:00.017, Memory: 6.00 MB

OK (2 tests, 2 assertions)

Figura 30. Resultado de SucursalControllerTest.

9. Conclusiones

Con el desarrollo del tema monográfico mediante el análisis de la estructura de trabajo se obtuvo un sistema web capaz de llevar un registro de la planificación de actividades y monitoreo de los gestores de crédito usando una aplicación móvil de recolección de información para las Microfinancieras de Nicaragua.

Esto fue llevado a cabo mediante la metodología de desarrollo cascada la cual provee de un marco flexible que permite representar de manera gráfica a través de diagramas los procesos o componentes del sistema haciendo uso del lenguaje unificado de modelado (UML) y sus herramientas.

Tomando en consideración que el optimizar ciertas actividades ejerce un mayor control sobre los colaboradores que desempeñan funciones fuera de las instalaciones, se logro una mayor eficiencia laboral y resultados prometedores en beneficio de la institución.

El éxito del tema monográfico fue llevado a cabo gracias al cumplimiento de los objetivos propuestos para el desarrollo del sistema en el cual se analizó la estructura de trabajo, obteniendo los requerimiento funcionales y no funcionales de este mismo, posteriormente se diseñó el modelo del sistema web utilizando las herramientas basadas en UML, una vez concluido el diseño se realizó la codificación del sistema en base el diseño previo bajo el lenguaje de programación PHP y JAVA siguiendo el patrón de arquitectura del software en capas MVC. Por último, se llevó a cabo exitosamente la implantación del sistema en un servidor web contratado por la empresa.

10. Recomendaciones

En base a las conclusiones se establecen las siguientes recomendaciones:

- Revisar y estudiar el manual de usuario proporcionado para una mejor comprensión y facilidad de uso del sistema.
- Almacenar de forma local los respaldos de seguridad de la base de datos que provee el servicio de hosting contratado.
- Se recomienda tener en cuenta la renovación anual de los servicios de hosting y dominio para asegurar la disponibilidad del sistema.
- Proveer de un software antivirus que al mismo tiempo pueda ser utilizado como firewall en las estaciones de trabajo de los responsables.
- Se recomienda realizar análisis de seguridad periódicos sobre el sistema en búsqueda de vulnerabilidades o cambios de las herramientas utilizadas (CSS y JQuery).

11. Referencias

- Accion International*. (20 de Junio de 2021). Obtenido de Powering Financial Inclusion Through Digital Technology - Accion: <https://www.accion.org>
- Arias, E. R. (1 de Agosto de 2020). *Modelo relacional - Qué es, definición y concepto*. Obtenido de <https://economipedia.com/definiciones/modelo-relacional.html>
- AS. (31 de Mayo de 2016). *Android Studio: Historia Resumida*. Obtenido de Android Studio FAQs: <https://androidstudiofaqs.com/conceptos/android-studio-historia>
- Cardador Cabello, A. L. (2014). *Implantación de aplicaciones web en entornos internet, intranet y extranet*. IC Editorial.
- CINK.ES. (19 de Julio de 2019). *Qué es una app*. Obtenido de Qué es una app: <https://cink.es/blog/que-es-una-app/#definicion-de-aplicaciones-moviles-o-apps>
- Corporation, IBM. (9 de Febrero de 2023). *IBM Corporation*. Obtenido de Planificación de actividades y tareas - Documentación de IBM: <https://www.ibm.com/docs/es/mas-cd/maximo-manage/continuous-delivery?topic=activities-planning-tasks>
- Group, I. (2016). *Working Day Suite for iPhone / Android App / Microsoft / BlackBerry*. Obtenido de Working Day Suite: <http://workingdaysuite.com/>
- IONOS. (11 de Marzo de 2019). *El modelo en cascada: desarrollo secuencial de software*. Obtenido de <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>
- IONOS. (11 de Marzo de 2019). *IONOS*. Obtenido de El modelo en cascada en el desarrollo de software: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>
- Laudon, K., & Laudon, J. (2012). *SISTEMAS DE INFORMACIÓN GERENCIAL*. Mexico: PEARSON EDUCACIÓN.
- Lucid Software Inc. (2022). Obtenido de ¿Qué es el lenguaje unificado de modelado (UML)?: https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml#section_0
- Microsoft. (2023). *Visual Studio Code - Code Editing. Redefined*. Obtenido de <https://code.visualstudio.com>
- mozilla.org contributors. (1998-2023). *JavaScript | MDN*. Obtenido de <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

- Oikocredit International. (20 de Junio de 2021). Obtenido de Oikocredit International: <https://www.oikocredit.coop/es/>
- OpenJS Foundation. (2022). *jQuery*. Obtenido de <https://jquery.com>
- Oracle. (2023). *Dev.java: The Destination for Java Developers*. Obtenido de <https://www.oracle.com/java/>
- Oracle Corporation. (2023). *MySQL*. Obtenido de <https://www.mysql.com>
- Santiago, R., & Trinaldo, S. (2015). *Mobile learning: Nuevas realidades en el aula*. Digital Text.
- Simplified. (s.f.). *Mobile Suitcase*. Obtenido de Mobile SuitCase: Automatiza y Optimiza los Procesos de tu Negocio: <https://www.mobilesuitcase.com/>
- Sommerville, I. (2005). *INGENIERIA DEL SOFTWARE*. ADDISON-WESLEY.
- Standard, H.L. (24 de Diciembre de 2023). *HTML Standard*. Obtenido de <https://html.spec.whatwg.org/multipage/>
- The PHP Group. (2001-2023). *PHP: Hypertext Preprocessor*. Obtenido de <https://www.php.net>
- UMU. (3 de Mayo de 2016). *¿Qué significa World Wide Web, web o www?* Obtenido de Definición de World Wide Web, web o www: <https://www.masadelante.com/faqs/www>
- W3C. (22 de Diciembre de 2023). *Cascading Style Sheets*. Obtenido de <https://www.w3.org/Style/CSS/>
- Zhang, D., & Adipat, B. (2005). Challenges, Methodologies, and Issues in the Usability Testing of Mobile Applications. *International Journal of Human-Computer Interaction*, 293-308.
- Zofío Jiménez, J. (2013). *Aplicaciones Web 2013*. Macmillan Profesional.

12. Anexos

12.1 Anexo 1: Contrato de Prestaciones de Servicios

12.1.1 Acerca de las partes

La empresa GICAPP, representada por **Jean Carlos Rocha**, mayor de edad, Casado, con domicilio en Managua, Carretera vieja a León km 12, número de identificación **001-200294-0035 B**, cumple con las funciones de Representante de la empresa a quien en lo sucesivo se le denominara **El Cliente**.

Los señores **Javier Enmanuel Matus Widdy**, mayor de edad, soltero, Egresado de la Carrera de Ingeniería en Computación, con número de identificación **001-030197-0003G**, con domicilio en Managua, Nicaragua. Y **Darwing Antonio Ruiz Herrera**, mayor de edad, soltero, Egresado de la Carrera de Ingeniería en Computación, con número de identificación **044-110895-0002C**, con domicilio en Managua, Nicaragua. A quienes en lo sucesivo se les denominará **El Equipo**.

Conjuntamente se le denominará **Las Partes**.

12.1.2 Clausulas

PRIMERA. Este contrato rige los servicios prestados única y exclusivamente durante la realización del proyecto denominado “Sistema web para la planificación de actividades y monitoreo de los gestores de crédito usando una aplicación móvil de recolección de información para las Microfinancieras de Nicaragua.”, concepto al cual se le denominará en lo sucesivo **Proyecto**.

SEGUNDA. Este contrato es un mutuo acuerdo de las partes, mismas que las firman al calce de la última hoja de este documento. Las partes reconocen que no existe mala fe o coacción, en ningún momento de celebrarlo.

TERCERA. El Equipo a través de la celebración de este contrato, proporcionará a **El Cliente** los servicios descritos como “Desarrollo de un Sistema

web para la planificación de actividades y monitoreo de los gestores de crédito usando una aplicación móvil de recolección de información para las Microfinancieras de Nicaragua.”, que consiste en presentar a **El Cliente** el material gráfico que constituya al diseño del sistema, con todos los elementos e indicaciones necesarios para su fin.

CUARTA. Se entregará: CD con todos los archivos del proyecto, bocetos aprobados, libro de estilo, manual de usuario, manual técnico, en un plazo que no excederá de **1 mes** a la finalización del proyecto.

QUINTA. El Equipo maneja un cronograma de proyecto que marcará fechas de revisión, presentación, tiempos de corrección, contacto y entrega final.

SEXTA El Cliente asumirá los gastos relacionados a las licencias de software necesarios para la realización del sistema. Los detalles de estas estarán plasmados en los documentos que los diseñadores le entreguen al cliente.

SÉPTIMA. Todas las piezas, técnicas, formatos, ilustraciones, sistemas, estilos y diseños son propiedad de **El Equipo**, pudiendo utilizar dichos elementos en el futuro en proyectos similares o diversos.

OCTAVA. El Equipo no puede, sin embargo, entregar un proyecto futuro con las mismas características; utilizando los elementos, piezas y diseños exactamente iguales al proyecto que nos ocupa.

NOVENA. El Cliente debe incluir los créditos de **El Equipo** en el proyecto, mediante los elementos que el mismo Equipo le proporcione, pudiendo llegar a un acuerdo –por necesidades de uso-, de colocar los créditos en una parte no visible.

DÉCIMA. El presente contrato tendrá una vigencia igual al tiempo en que dure la prestación de los servicios por parte del **Equipo** a **El Cliente**. Todo lo no expuesto o tratado en este contrato será resuelto por las partes mediante documento firmado que se anexa a este. Durante el desarrollo del sistema, se poseerá un plazo estimado para la realización de cambios:

- Caducado el tiempo de desarrollo no se realizará mantenimiento al sistema desarrollado.
- El producto final será entregado sin código fuente solo su equivalente ejecutable en caso contrario tendrá un coste extra a criterio del desarrollador/es.
- Ningún daño provocado al sistema y/o ambiente de desarrollo provocado por agentes externos será adeudado a los desarrolladores.
- En ninguna circunstancia, los desarrolladores del sistema se harán responsables de fallos post entrega.

Una vez aprobado el proyecto de **GICAPP**. tendrán los derechos de anexar sus cláusulas de contratación de los desarrolladores del sistema.

12.2 Anexo 2: Capturas del Sistema

12.2.1 Sistema web

Show 10 entries Search:

#	Fecha	Nombre	Descripcion	Acciones
1	2021-08-02 15:36:07	I. Datos generales del Cliente	I. Datos generales del Cliente	   
2	2021-08-02 16:06:50	II. Datos del Conyugue	II. Datos del Conyugue	   
3	2021-08-02 16:09:38	III. Presupuesto Familiar	III. Presupuesto Familiar	   
4	2021-08-03 11:06:40	IV. Referencias Personales	IV. Referencias Personales	   
5	2021-08-03 11:12:41	V. Datos Generales del Negocio	V. Datos Generales del Negocio	   
6	2021-08-03 15:19:10	VI. Declaración de deudas con Proveedores o Financieras	VI. Declaración de deudas con Proveedores o Financieras	   
7	2021-08-03 15:27:43	VII. Gastos del Negocio.	VII. Gastos del Negocio.	   
8	2021-08-03 15:36:20	VIII. Intención del Crédito	VIII. Intención del Crédito	   
9	2021-08-03 15:42:55	IX. Análisis de Evaluación al Cliente	IX. Análisis de Evaluación al Cliente	   

cciones del Formulario: SOL

Lista de Secciones

Show 10 entries

Nueva Sección

Nombre de la Sección

A

Descripcion de la Sección

B

Cerrar Guardar

Fecha	Nombre	Descripcion
2021-08-02 15:36:07	I. Datos ge	del Cliente
2021-08-02 16:06:50	II. Datos d	ugue
2021-08-02 16:09:38	III. Presup	miliar
2021-08-03 11:06:40	IV. Referencias Personales	IV. Referencias Personales
2021-08-03 11:12:41	V. Datos Generales del Negocio	V. Datos Generales del Negocio

Administración de Usuarios

Inicio / Usuarios [+ Agregar Usuario](#)

Lista de Usuarios

Copy CSV Excel PDF Print Search:

Persona:	Sucursal:	Usuario:	Registrado El:	Registrado Por:	Estado:	Opciones:
ADMON	CASA MATRIZ	admon	07-06-2021		Activo	    
Alan Ariel Neira Ramirez	CASA MATRIZ	aneira	08-06-2021	ADMON	Activo	    
Alexander Barrios Altamirano	CASA MATRIZ	ABARRIOS	25-03-2022	ADMON	Activo	    
Carlos Portobancos Vargas	CASA MATRIZ	CPORTOBANCO	25-03-2022	ADMON	Activo	    
Ernesto José Lopez Bolaños	CASA MATRIZ	elopez	08-06-2021	ADMON	Activo	    
Esmeralda del Carmen Valle López	CASA MATRIZ	evalle	08-06-2021	ADMON	Activo	    
Ever Antonio Loasiga Uriarte	CASA MATRIZ	eluaisiga	08-06-2021	ADMON	Activo	    
Janiel Andres Zelaya Flores	CASA MATRIZ	jzelaya	08-06-2021	ADMON	Activo	    
Josue Jiron	CASA MATRIZ	JJIRON	25-03-2022	ADMON	Activo	    
Leydi del Socorro Urbina Morales	CASA MATRIZ	lurbina	08-06-2021	ADMON	Inactivo	    

Showing 1 to 10 of 14 entries

Previous **1** 2 Next

Otorgar Acceso a Sucursales

Formularios Personales

Formularios Creados

Excel PDF Print

Preview Form

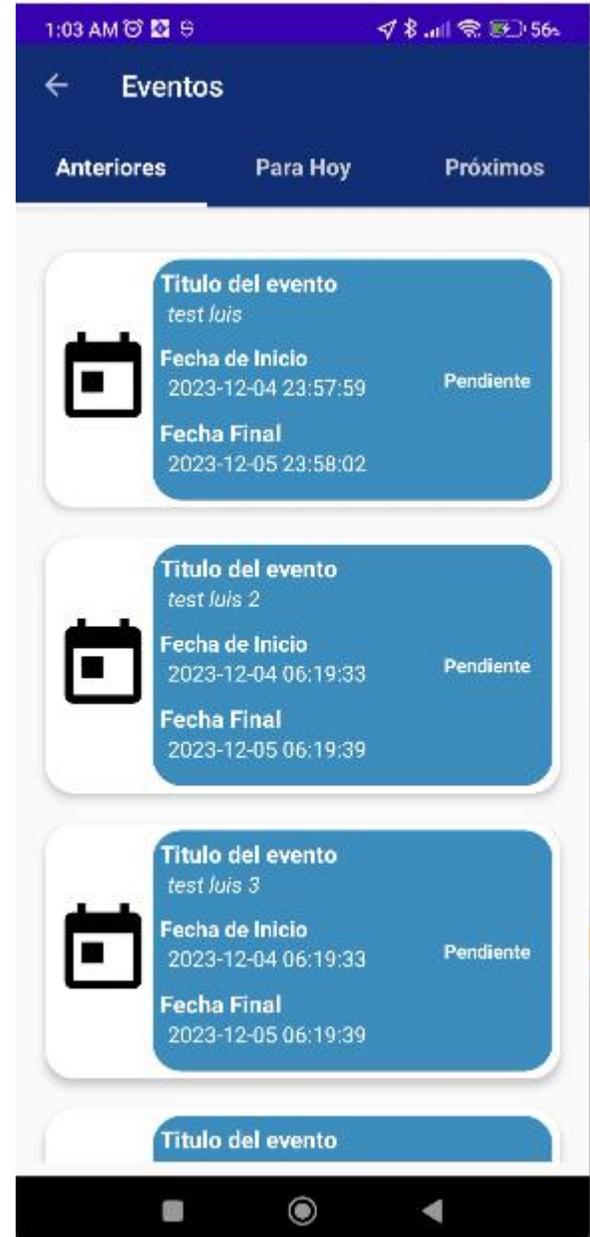
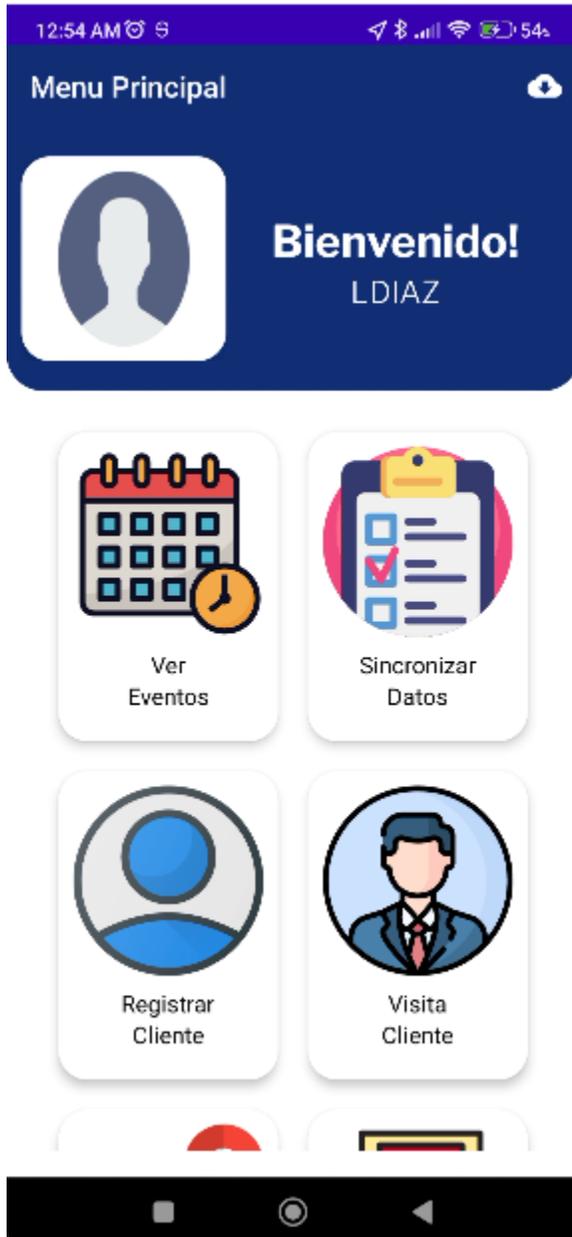
SONDEO DE PRECIO CANAL SUPERMERCADO

- I. Productos KolaShaler
- II. Productos Coca Cola
- III. Productos Pepsi
- IV. Productos Prix Cola
- V. Productos Milca
- VI. Firmas Display

X Cerrar

No. Secciones	No. Campos Tot
6	46
6	65
6	20

12.2.2 Aplicación Móvil



12.3 Anexo 4: Entrevista

La entrevista, que se llevó a cabo con éxito, proporcionó información crucial para el desarrollo del sistema web y la aplicación móvil diseñados para la recolección de datos y el monitoreo del personal. Los puntos clave discutidos durante la entrevista incluyeron:

➤ Requisitos Funcionales:

Se identificaron y definieron las funciones esenciales del sistema, tales como la captura de datos en tiempo real, la generación de informes y la gestión de perfiles de usuario con diferentes niveles de acceso.

➤ Plataformas y Tecnologías:

Durante la entrevista, se llegó a un consenso sobre las plataformas, tecnologías y lenguajes de programación a utilizar tanto en el desarrollo del sistema web como en la aplicación móvil. Además, se exploraron posibles integraciones con otras herramientas existentes.

➤ Diseño de Interfaz de Usuario:

Los requisitos de diseño fueron detalladamente revisados para asegurar una experiencia de usuario intuitiva y atractiva. Se discutieron preferencias de diseño, la estructura de la información y la adaptabilidad a dispositivos móviles.

➤ Seguridad y Privacidad:

Se abordaron las preocupaciones de seguridad y privacidad de los datos. Durante la entrevista, se acordaron medidas de seguridad robustas y estrategias para proteger la información sensible recopilada a través del sistema.

➤ Etapas de Desarrollo y Cronograma:

Un cronograma tentativo fue establecido durante la entrevista, identificando hitos clave y posibles fechas de entrega. Además, se discutieron las etapas de prueba y evaluación para garantizar la calidad del sistema.

➤ Capacitación y Soporte:

Se planificaron las necesidades de capacitación para los usuarios finales, y se acordó un plan de soporte técnico para garantizar la eficiencia continua del sistema después de su implementación.

➤ Escalabilidad y Futuras Mejoras:

Durante la entrevista, se exploraron posibles planes para la escalabilidad futura del sistema. Además, se discutieron ideas para futuras mejoras, asegurando que el sistema pueda adaptarse a cambios y crecimiento a largo plazo.