

**DISEÑO DE UNA APLICACIÓN WEB PARA EL PROCESO DE CARTERIZACIÓN DE
LA EMPRESA COLOMBIA TELECOMUNICACIONES S.A**

ANDREY JULIAN SOTELO MONCADA

**UNIVERSIDAD SANTO TOMÁS
FACULTAD DE INGENIERIA DE TELECOMUNICACIONES
BOGOTÁ D.C.
2023**

**DISEÑO DE UNA APLICACIÓN WEB PARA EL PROCESO DE CARTERIZACIÓN DE
LA EMPRESA COLOMBIA TELECOMUNICACIONES S.A**

ANDREY JULIAN SOTELO MONCADA

**Proyecto de pasantía de grado en la empresa Colombia
Telecomunicaciones S.A para optar por el título de Ingeniero de
Telecomunicaciones**

Dirigido por:

ING. GERALD BREEK FUENMAYOR RIVADENEIRA

**UNIVERSIDAD SANTO TOMÁS
FACULTAD DE INGENIERIA DE TELECOMUNICACIONES
BOGOTÁ D.C.
2023**

Nota de aceptación

Firma del presidente del jurado

Firma del Jurado

Firma del Jurado

Bogotá D.C. 11, septiembre, 2023

RESUMEN

A lo largo del presente trabajo se realiza el diseño de un aplicativo web, el cual permita automatizar un proceso que se maneja actualmente dentro de la empresa Telefónica en el área B2B, para ello se realiza un diseño haciendo uso de las metodologías ágiles y de la metodología de cascada mejorado. Se inicia con una contextualización de cómo se maneja actualmente el proceso, a partir de allí se cuenta con un levantamiento de requerimientos necesarios para el correcto funcionamiento de la aplicación, esto permite el diseño de un modelo funcional del sistema, el cual incluye la determinación de los casos de uso, así como el análisis de los actores que intervienen en la aplicación para finalmente realizar el diseño de una base de datos que soporte el sistema y un modelado de las vistas que otorguen una idea más cercana a como se vera la aplicación si se aprueba su construcción.

Como se mencionó anteriormente, con el diseño de este aplicativo, se busca reducir el tiempo que implica la realización del proceso de carterización y también reducir los errores que se puedan presentar.

La metodología implementada en el trabajo fue la cascada mejorada, una metodología con bastantes años de uso, que sigue utilizando conceptos básicos, pero sigue siendo de gran ayuda cuando se aplica de manera correcta. Para el desarrollo del actual trabajo se utilizó hasta la fase dos, la cual es la fase de diseño. Y se apoyó con la metodología de SCRUM, la cual permite gestionar el proyecto basado en buenas prácticas.

Palabras clave

Diseño, automatización, aplicación web, ingeniería, software, carterización, metodologías ágiles.

ABSTRACT

The following document presents the design of a web application, its main objective is to automate an internal process that exists within the Telefonica company in the B2B area, within the work a design is made using agile methodologies, initially it is explained how the process currently works, then the requirements are collected and this allows a functional model of the system to be built, where the actors of the application and the use cases are determined, after that the creation of the base was done. of data and a modeling of the views of the application was made.

The creation of this application seeks to reduce the time used in the signage process and reduce errors that occur.

Keywords

Design, automation, web application, engineering, software, branding, agile methodologies.

Tabla de contenido

1	CONTEXTUALIZACIÓN.....	16
1.1	PLANTEAMIENTO DEL PROBLEMA.....	16
1.2	JUSTIFICACIÓN	18
1.3	OBJETIVOS	20
1.3.1	OBJETIVO GENERAL	20
1.3.2	OBJETIVOS ESPECIFICOS	20
1.4	ALCANCE	21
1.5	METODOLOGIA.....	22
1.6	CRONOGRAMA.....	26
2	MARCO TEORICO.....	27
2.1	ESTADO DEL ARTE.....	27
2.2	MARCO CONCEPTUAL.....	29
2.2.1	ARQUITECTURA CLIENTE-SERVIDOR	29
2.2.2	FRONT END	30
2.2.3	HTML	30
2.2.4	CSS:	31
2.2.5	JAVA SCRIPT:.....	32
2.2.6	FRAMEWORKS:.....	32
2.2.7	BACKEND:	34
2.2.8	PYTHON:.....	34

2.2.9	BASES DE DATOS	35
2.2.10	MODELOS DE BASES DE DATOS.....	37
2.2.11	MODELO ENTIDAD-RELACIÓN:.....	39
2.2.12	SQL (Structured Query Language):.....	40
2.2.13	SISTEMAS GESTORES DE BASES DE DATOS RELACIONALES:.....	41
2.2.14	BASES DE DATOS NO RELACIONALES.....	42
2.2.15	UML (Lenguaje Unificado de Modelado)	44
2.2.16	MODELOS DE DESARROLLO WEB	45
2.2.17	MOTOR DE REGLAS	52
2.2.18	MOCKUPS.....	53
2.2.19	FIGMA	53
3	DESARROLLO Y RESULTADOS.	53
3.1	ETAPA 1 ANALISIS	54
3.1.1	REQUERIMIENTOS	54
3.1.2	HISTORIA DE USUARIO	56
3.1.3	REQUISITOS FUNCIONALES.....	63
3.1.4	REQUISITOS NO FUNCIONALES	65
3.1.5	CASOS DE USO.....	66
3.2	ETAPA 2 DISEÑO	85
3.2.1	PRODUCT BACKLOG	86
3.2.2	SPRINT BACKLOG.....	87

3.2.3	DISEÑO DE BASE DE DATOS	88
3.2.4	VISTAS CONCEPTUALES DEL APLICATIVO	92
3.2.5	DIAGRAMA DE DESPLIEGUE	108
3.2.6	MODELO DE NAVEGACIÓN	109
3.2.7	CONSIDERACIONES DE ACCESIBILIDAD Y USABILIDAD	110
3.3	ANÁLISIS Y VALIDACIÓN DE RESULTADOS	111
4	CONCLUSIONES Y RECOMENDACIONES.....	113
4.1	CONCLUSIONES.....	113
4.2	RECOMENDACIONES	114
5	REFERENCIAS.....	115
6	APÉNDICES.	118
6.1	Apéndice A.....	118
6.2	Apéndice B.....	119

Lista de ilustraciones

Ilustración 1 Fases metodología en cascada	22
Ilustración 2 CRONOGRAMA.....	26
Ilustración 3 HTML, CSS, JS	30
Ilustración 4 FRONTEND Y BACKEND	34
Ilustración 5 Ranking lenguajes de programación.	35
Ilustración 6 MODELO RELACIONAL	38
Ilustración 7 MODELO ENTIDAD RELACIÓN	40
Ilustración 8 GESTORES DE BASES DE DATOS	42
Ilustración 9 MOELO CLAVE-VALOR.....	42
Ilustración 10 LENGUAJES NOSQL.....	43
Ilustración 11 METODOLOGIAS AGILES	47
Ilustración 12 Metodología de cascada mejorado	48
Ilustración 13 METODOLOGIA EN ESPIRAL	51
Ilustración 14 Contextualización carterización	56
Ilustración 15 Formato historia de usuario	57
Ilustración 16 Historia de usuario 01.....	57
Ilustración 17 Historia de usuario 02.....	58
Ilustración 18 Historia de usuario 03.....	58
Ilustración 19 Historia de usuario 04.....	58
Ilustración 20 Historia de usuario 05.....	59
Ilustración 21 Historia de usuario 06.....	59
Ilustración 22 Historia de usuario 07.....	59
Ilustración 23 Historia de usuario 08.....	60
Ilustración 24 Historia de usuario 09.....	60

Ilustración 25 Historia de usuario 10.....	60
Ilustración 26 Historia de usuario 11	61
Ilustración 27 Historia de usuario 12.....	61
Ilustración 28 Historia de usuario 13.....	61
Ilustración 29 Historia de usuario 14.....	62
Ilustración 30 Historia de usuario 15.....	62
Ilustración 31 Historia de usuario 16.....	62
Ilustración 32 Historia de usuario 17	63
Ilustración 33 Casos de uso Administrador	66
Ilustración 34 Casos de uso Consultador	73
Ilustración 35 Casos de uso modificador	77
Ilustración 36 Formato Product Backlog	86
Ilustración 37 Modelo relacional	89
Ilustración 38 BD Analistas	90
Ilustración 39 BD Clientes	90
Ilustración 40 BD Cuadrillas	90
Ilustración 41 BD tabla rompimiento cuadrillas-analistas	91
Ilustración 42 BD Reglas	91
Ilustración 43 BD Roles	91
Ilustración 44 BD Segmento	91
Ilustración 45 BD usuarios.....	91
Ilustración 46 BD usuarios Telefónica	92
Ilustración 47 Login de la aplicación	92
Ilustración 48 Página principal del administrador	93
Ilustración 49 Usuarios actuales	93
Ilustración 50 Modificar usuario	94

Ilustración 51 Confirmar modificar usuario.....	94
Ilustración 52 Eliminar usuario.....	95
Ilustración 53 Confirmación usuario eliminado.....	95
Ilustración 54 Agregar usuario	96
Ilustración 55 Confirmar agregar usuario.....	96
Ilustración 56 Opción consultar Admin.....	97
Ilustración 57 Estadística.....	97
Ilustración 58 Segmento a consultar.....	98
Ilustración 59 Servicio a consultar	98
Ilustración 60 Aplicativo móvil.....	99
Ilustración 61 Aplicativo fija.....	99
Ilustración 62 Carterización actual.....	100
Ilustración 63 Descargar carterización.....	100
Ilustración 64 Consultar analistas	101
Ilustración 65 Consultar cuadrillas	101
Ilustración 66 Página principal Consultador y Modificador	102
Ilustración 67 Página consultar del Modificador.....	103
Ilustración 68 Página carterización Modificador.....	103
Ilustración 69 Consulta de reglas.....	104
Ilustración 70 Nueva regla	104
Ilustración 71 Confirmación regla creada.....	105
Ilustración 72 Modificar regla.....	105
Ilustración 73 Confirmación regla modificada	106
Ilustración 74 Eliminar regla	106
Ilustración 75 Confirmación regla eliminada	107
Ilustración 76 Cerrar sesión.....	107

Ilustración 77 Diagrama de despliegue.....	109
Ilustración 78 Modelo de navegación	109

Lista de tablas

Tabla 1 Relación de objetivos con fases de modelo cascada mejorado.....	25
Tabla 2 Requisitos funcionales	63
Tabla 3 Requisitos no funcionales	65
Tabla 4 Caso de uso Login Admin	67
Tabla 5 Caso de uso alta de usuarios Admin	68
Tabla 6 Caso de uso consulta de usuarios Admin.....	68
Tabla 7 Caso de uso gestión de usuario Admin	69
Tabla 8 Caso de uso baja de usuario Admin.....	70
Tabla 9 Caso de uso consulta de carterización Admin	70
Tabla 10 Caso de uso Descarga documentos de carterización Admin.....	71
Tabla 11 Caso de uso Consulta de estadísticas Admin	72
Tabla 12 Caso de uso logout Admin	72
Tabla 13 Caso de uso Login consultador	73
Tabla 14 Caso de uso consulta de carterización consultador.....	74
Tabla 15 Caso de uso descarga documentos de carterización consultador	75
Tabla 16 Caso de uso Consulta de estadísticas Consultador	75
Tabla 17 Caso de uso Logout consultador.....	76
Tabla 18 Caso de uso Login Modificador	77
Tabla 19 Caso de uso consulta de carterización Modificador.....	78
Tabla 20 Caso de uso descarga documentos de carterización Modificador	79
Tabla 21 Caso de uso consulta de estadística Modificador.....	79
Tabla 22 Caso de uso subida documentos de carterización Modificador	80
Tabla 23 Caso de uso Consultar reglas de carterización Modificador	81

Tabla 24 Caso de uso crear reglas de carterización Modificador	81
Tabla 25 Caso de uso cambiar reglas de carterización Modificador	82
Tabla 26 Caso de uso eliminar reglas de carterización Modificador	83
Tabla 27 Caso de uso generador de carterización Modificador	84
Tabla 28 Caso de uso Logout Modificador	85
Tabla 29 Product Backlog	87
Tabla 30 Sprint Backlog	88
Tabla 31 Accesibilidad y usabilidad	110

INTRODUCCIÓN

Cada día el uso de la tecnología se hace más frecuente, al punto de que, en muchos hogares, la mayoría de los miembros cuentan con un producto de la tecnología como lo es un dispositivo móvil, el cual se ha convertido en parte de nuestra vida diaria. Esto demuestra la evolución de la era tecnológica, ahora bien, y como es lógico, las empresas también han mejorado su productividad con ayuda de las mejoras tecnológicas con las que se cuenta, y muchas de ellas son las aplicaciones.

Es por eso por lo que a lo largo del trabajo se busca explorar y comprender el funcionamiento de un proceso interno conocido como carterización, la cual se maneja dentro del área B2B en la empresa Telefónica.

Telefónica es una empresa multinacional, encargada de ofrecer servicios fijos, móviles y digitales. De hecho, logró posicionarse en el puesto 196 del ranking de empresas más grandes del mundo en el año de 2022 según la revista Forbes [1] entre otras muchas más características que refuerzan el concepto de empresa líder en el área de las telecomunicaciones en Latinoamérica y Europa, haciendo uso de herramientas como la fusión con otras compañías que ha elevado el portafolio de servicios [2].

Al ser una empresa líder en las telecomunicaciones, se hace indispensable el estar a la vanguardia de la mejoras tecnológicas. Dado esto se busca brindar una solución para aumentar la productividad basado en la reducción del tiempo de ejecución de la tarea. En otras palabras, se podría llamar como la automatización del proceso, y para lograr esto se propone el diseño de un software tipo aplicativo web que cumpla con esta función.

Para lograrlo es importante hacer un correcto levantamiento de requerimientos que sirvan como insumo para la construcción del diseño de la aplicación, a partir de allí se pueden realizar los demás pasos que brinden una idea clara del funcionamiento de la aplicación.

1 CONTEXTUALIZACIÓN.

1.1 PLANTEAMIENTO DEL PROBLEMA

Todas las grandes empresas están constituidas de diferentes áreas que se encargan de administrar y solventar los posibles inconvenientes que puedan surgir en los servicios que prestan. A su vez, cada área está constituido por una serie de profesionales los cuales cuentan con tareas específicas que desarrollan de manera óptima. Evidentemente, esta no es la excepción con Colombia telecomunicaciones S.A. o mejor conocida como Telefónica.

Ahora bien, varias de estas tareas ocupan gran porcentaje de tiempo de quien las realiza, y el mayor problema radica en que son tareas que se ejecutan con una frecuencia importante, por ejemplo, con una frecuencia diaria o semanal.

Una de las tareas más relevantes en el ámbito de la atención al cliente es la carterización, la cual, de manera simplificada, es la asignación de analistas de atención a cada uno de los clientes empresariales de Telefónica. Esta carterización es importante, porque al momento de que un cliente busca comunicarse con la empresa, a través de los diferentes canales de atención que se manejan, se debe dirigir la consulta al analista correspondiente según la carterización actual.

Sin embargo, la problemática central es el hecho de que este proceso es ejecutado de manera manual haciendo uso de herramientas como Excel, en donde está el consolidado de los analistas disponible. Intentar resolver esta situación traería beneficios importantes especialmente con el recurso del tiempo. Para profundizar más acerca de este proceso fue realizada una entrevista a uno de los encargados de la ejecución de la tarea (Ver apéndice A).

Dada la situación descrita, surge la pregunta ¿Es posible automatizar el proceso de carterización que se maneja dentro de Telefónica para reducir el tiempo de ejecución de la tarea y la tasa de errores que se puedan presentar?

La automatización de un proceso tiene tal importancia que representa diferentes beneficios a nivel empresarial, por ejemplo, según estudios realizados por McKinsey & Company se puede evidenciar que con la automatización de procesos es posible aumentar en un 30% la eficiencia y adicional a esto generar una reducción de costos en un 60%. [3]. Asimismo, se puede reducir la tasa de errores en cerca de un 20% y aumentar la calidad de los procesos [4]. Estos números reflejan cómo las empresas han logrado una migración lenta pero eficaz en la automatización de la mayoría de los procesos manuales y repetitivos que se utilizan.

1.2 JUSTIFICACIÓN

El tiempo es un recurso muy valorado en cualquier ámbito de la vida, pero cuando se está en el espacio laboral es aún máspreciado. Con esto en mente es normal que se recuerde una frase muy popular “el tiempo es dinero” la cual representa cómo esta variable inmaterial afecta la vida cotidiana. Debido a esto, cualquier herramienta que permita reducir el tiempo que conlleva ejecutar una tarea representa una ayuda inmediata.

Disminuir el tiempo de ejecución de una tarea no solo representa un beneficio a la empresa, sino que también le presenta al ejecutor. Por ejemplo, si se logra reducir el tiempo de una tarea al 60% del tiempo original, ya es una ventana de tiempo ganada que puede ser aprovechada en otras tareas.

Ahora bien, el hecho de ejecutar procesos de manera manual trae consigo una probabilidad de error que es necesario tener en cuenta, especialmente porque lleva de nuevo al punto anterior del manejo de tiempo. Cometer un error puede que no tenga una consecuencia importante en términos monetarios, de integridad o de seguridad, pero si puede representar el hecho de tener que volver a realizar una tarea y utilizar tiempo destinado a otras actividades.

Debido a esto, surge la posibilidad del diseño de un aplicativo web que permita reducir el tiempo en el cual se realiza el proceso de carterización. Adicional a esto, también se busca disminuir la tasa de errores que se puedan presentar.

El desarrollo de una aplicación se debe analizar desde diferentes perspectivas, una de ellas es el tema legal y como puede afectar al diseño de la solución planteada. Esta aplicación está enfocada únicamente para el uso de Telefónica, por diferentes razones, la primera por temas de confidencialidad, ya que se usarían datos de propiedad de la empresa, y adicionalmente este es un proceso que se maneja así dentro de Telefónica. En otras palabras, no se podría convertir en una aplicación comercial porque no les serviría a otras empresas. Dentro del ámbito legal

colombiano, no hay una ley específica que rijan el diseño y desarrollo de las aplicaciones, sin embargo, hay algunas leyes que deben ser tenidas en cuenta, por ejemplo la ley 603 del 2000, la cual es la ley de la propiedad intelectual y establece que el software que se desarrolle está protegida por los derechos de autor [5], adicionalmente también existe la ley Estatutaria 1581 de 2012 la cual regula el tratamiento de los datos personales en Colombia, y está relacionada directamente con el diseño del aplicativo ya que la información que se manejan son de los clientes [6].

De igual manera, es importante determinar el impacto ambiental que se pueda presentar, y aunque de manera directa es casi imperceptible, si lo causa de manera indirecta. Por ejemplo, el insumo de la aplicación, los datos, deben ser almacenados en algún servidor que consume energía y genera emisiones de efecto invernadero que aumentan la huella de carbono. Así mismo, los dispositivos físicos que usarían la aplicación con el tiempo se pueden convertir en un residuo que genera daño, adicional al hecho de no desarrollar la aplicación haciendo uso de buenas prácticas que permita garantizar un diseño sostenible y una eficiencia energética [7].

Finalmente, al ser un proyecto enfocado en el diseño de un aplicativo permite poner en práctica diferentes conocimientos adquiridos en la carrera, como la programación, manejo de bases de datos, y el diseño y desarrollo de aplicaciones, pero en esta ocasión aplicados a un entorno laboral.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

Diseñar un software que permita reducir el tiempo de ejecución del proceso de carterización que se maneja dentro de la empresa Telefónica en la sede Morato ubicada en Bogotá.

1.3.2 OBJETIVOS ESPECIFICOS

- Analizar el proceso de carterización de Telefónica, a través de la contextualización del funcionamiento actual del mismo.
- Determinar los requerimientos y requisitos que servirán de base para el diseño de la solución planteada.
- Elaborar el modelo arquitectural del software teniendo en cuenta los requerimientos y requisitos especificados.

1.4 ALCANCE

El proceso de carterización hace parte de diferentes áreas dentro del segmento de B2B de Telefónica, y es una actividad que en cierta medida puede ser consultada por cualquier miembro del área. Es por eso, que el proyecto está orientado a esa área con el diseño del software propuesto, el cual busca facilitar un proceso que afecta indirectamente a los usuarios y que tendría un uso importante ya que permite mejorar la calidad del servicio de atención al usuario, dado que este es un proceso que tiene un uso bastante frecuente con un promedio de 2 veces a la semana.

De igual manera, también se puede mencionar que el software propuesto podría mantenerse activo a lo largo del tiempo debido a que es una tarea que puede seguir siendo utilizada dentro de la compañía.

Ahora bien, la idea del software es determinar el modelo de atención que tendrá un cliente dependiendo del segmento en que se encuentre, entonces arrojará una serie de archivos que llevarán la información del cliente y quien lo atenderá. Una vez con estos archivos, ya se podrán cargar a las diferentes plataformas dependiendo de su uso.

No obstante, para el cumplimiento del proyecto, se hace uso de la metodología de cascada mejorado, la cual se compone de cinco fases y para dar cumplimiento con lo proyectado se utiliza hasta la segunda fase, conocida como fase de diseño. Las demás fases se pueden utilizar en caso de que se quiera dar continuidad al proyecto iniciado con la fase de implementación. La siguiente ilustración puede ayudar a comprender de mejor manera cuales son las fases que se utilizaron.

Ilustración 1 Fases metodología en cascada



FUENTE: Creación propia

1.5 METODOLOGIA

Teniendo en cuenta la finalidad del proyecto, la investigación que se decidió implementar fue de tipo aplicada, con la cual se garantizó hacer uso de los conocimientos obtenidos. Con este tipo de investigación se buscó obtener un resultado práctico y así dar solución a la falta de automatización del proceso de carterización [8].

Ahora bien, teniendo en cuenta los objetivos planteados, lo más recomendable fue el uso de una investigación de tipo aplicada y descriptiva apoyado de un método cualitativo dado que el objetivo principal del proyecto se basa en el diseño de un software.

Adicionalmente el uso de artículos académicos, documentos, ideas de especialistas, entre otros, se convirtieron en fuentes de información secundarias ya que otorgan herramientas adicionales a la hora del diseño de la aplicación.

De igual manera, es importante determinar cuál es el método de gestión de proyecto y desarrollo de software más conveniente a seguir dependiendo de las necesidades actuales. Debido a ello se logró determinar que el método a utilizar para la gestión del proyecto es la

metodología ágil de scrum, pero también se utilizó en conjunto con el método en cascada mejorado para desarrollo de software, el cual es un modelo en el que se trabaja de manera secuencial, logrando que una etapa inicie después de finalizar la anterior y realizando retroalimentación entre fases. Con esto se buscó garantizar el desarrollo del trabajo de una manera ideal haciendo uso de una de las metodologías más usadas en la actualidad, y que permite tener un resultado favorable.

Así mismo es importante resaltar que a lo largo del proyecto se abordaron las dos primeras fases del método en cascada mejorado (Análisis y diseño), esto se debe a que el proyecto se enfocó en el diseño de un aplicativo web y si finalmente se decide implementar se pueden utilizar las fases siguientes [9].

Durante la primera fase del método de cascada mejorado, la fase de análisis fue necesario hacer el levantamiento de requerimientos, lo cual son las necesidades con las que cuenta el usuario y por lo tanto necesita ser solventadas. Esta información fue recogida con base al reporte otorgado por el personal encargado de realizar el proceso y así se convirtió en la fuente de información primaria a la hora del diseño del aplicativo.

Para la recolección de información se utilizaron principalmente las reuniones con los encargados del proceso, ya que en este caso ellos actúan como el cliente, adicionalmente la observación ocular del proceso es otra técnica que también se utilizó, de igual manera para el diseño de la aplicación también se hizo uso de herramientas como búsqueda en repositorios académicos y el uso de tipo digital para la gestión de proyectos. Este proceso de levantamiento de requerimientos tuvo un tiempo predeterminado de alrededor de dos semanas. Sin embargo, al hacer uso de la metodología ágil de Scrum se buscaba estar en constante retroalimentación con el personal que cumple las veces de cliente.

Para la segunda fase, fase de diseño, se incluyó la creación de prototipos de interfaces de usuario (mockups) además de la elaboración de diferentes tipos de diagramas como el diagrama de despliegue o el modelo de navegación, figuras, tablas, esquemas. Para ello se utilizaron plataformas de acceso gratuito disponibles en la red, algunas de las elecciones fueron NinjaMock, Figma, Draw.io o incluso aplicaciones de escritorio como Excel, Power Point entre otros. El uso de estas aplicaciones permitió dar cumplimiento a la fase de diseño propuesta gracias al uso de la metodología de cascada mejorado.

En cuanto a la validación del diseño, se utilizaron principalmente test o encuestas con los encargados del proceso. Aunque el proceso es manejado por varias personas, principalmente son 2 los que desarrollan la mayor parte, a ellos dos se lograron realizar las encuestas para medir la satisfacción con el diseño presentado.

Para relacionar los objetivos con las fases de modelo utilizado, se presenta la siguiente tabla, la cual permite dar mayor claridad de la información presentada. En la tabla se encuentran los objetivos del proyecto, las fases del modelo en cascada mejorado, los Sprint (gracias a la metodología de SCRUM) con los resultados obtenidos.

Tabla 1 Relación de objetivos con fases de modelo cascada mejorado

O. General	Objetivos	Fases modelo Cascada	Sprint(scrum)	Resultados
Diseñar un software que permita reducir el tiempo de ejecución del proceso de carterización que se maneja dentro de la empresa Telefónica en la sede Morato ubicada en Bogotá.	Analizar el proceso de carterización de Telefónica, a través de la contextualización del funcionamiento actual del mismo.	Análisis	1	Se da cumplimiento al objetivo a través de una especificación completa de cómo se maneja la carterización dentro de Telefónica, a partir de eso se puede empezar a construir el aplicativo
	Determinar los requerimientos y requisitos que servirán de base para el diseño de la solución planteada.	Análisis	1	Se realiza el levantamiento de requerimientos haciendo uso de la herramienta historias de usuario, esto permite que también se desarrollen los requisitos del aplicativo

	Elaborar el modelo arquitectural del software teniendo en cuenta los requerimientos y requisitos especificados	Diseño	2,3	Se obtienen diferentes tipos de modelos que dan un sentido completo a la aplicación, por ejemplo, se obtiene el diagrama de despliegue, modelo de navegación, diagrama de bases de datos, vistas conceptuales del aplicativo (Mockups)
--	--	--------	-----	--

FUENTE: Creación propia

1.6 CRONOGRAMA

El cronograma ejecutado fue el siguiente, el cual tiene en cuenta las tareas a realizar para alcanzar los objetivos propuestos y también la duración:

Ilustración 2 CRONOGRAMA

Etapa	Actividad	mes 1				mes 2				mes 3				mes 4				mes 5				mes 6		
		S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3
I	Planteamiento del proyecto																							
II	Determinación de la necesidad de la realización del proyecto																							
III	Definición de objetivos																							
IV	Delimitación de alcance del proyecto																							
V	Definición de metodología																							
VI	Inicio de recolección de información																							
VII	Sprint planning																							
VIII	Especificación de requerimientos																							
IX	Determinación de requisitos funcionales y no funcionales																							
X	Identificación de actores asociados al proceso, incluyendo sus funciones																							
XI	Especificación de casos de uso																							
XII	Sprint review																							
XIII	Sprint planning																							
XIV	Diseño de modelo de bases de datos																							
XV	Creación de mockups para el modelado de la aplicación																							
XVI	Creación de diagrama de despliegue																							
XVII	Diseño de modelo de navegación																							
XVIII	Consideraciones de accesibilidad y usabilidad																							
XIX	Presentación de diseño a los encargados																							
XX	Conclusiones y recomendaciones																							

FUENTE: Creación propia

2 MARCO TEORICO.

2.1 ESTADO DEL ARTE

A continuación, se relacionaron una serie de trabajos en los cuales se proponen, diseñan y/o desarrollan aplicativos para dar solución a sus respectivas problemáticas, algunos de estas propuestas fueron realizados dentro de empresas con gran recorrido nacional e internacional, y da un punto de partida para entender la situación actual del desarrollo de software.

- Diseño de plataforma web para el apoyo y soporte en el proceso de gestión de incidentes en una red móvil de una empresa de telecomunicaciones:

En este trabajo se desarrolla una solución que consiste en realizar una actualización al software que utilizan para la gestión de las estaciones base, adicional se debe incluir una serie de nuevos requerimientos solicitados [10].

- Implementación de solución de Software Plaza Core para gestión empresarial y operativa de aplicaciones omnicanal:

En este caso, se construyó una aplicación web que se enfoca en e-commers basa en Woocommerce, la cual es una aplicación líder en el mercado, adicionalmente realizan una integración con un software de ERP [11].

- Diseño de un aplicativo web para mejorar el control, la producción y el rendimiento de los ingenieros en el área de documentación y transporte en el proyecto Datafill en ZTE Colombia S.A.S:

La solución planteada consiste en diseñar un software de sistemas de información, el cual busca apoyar el proyecto Datafill de la empresa ZTE, con la implementación de este aplicativo se busca mejorar la producción y el control de los participantes del proyecto Datafill [12].

- Diseño e implementación frontend de un software libre para el monitoreo y análisis de información en bases de datos enfocado a contribuir con el crecimiento de pequeñas y medianas empresas:

En este proyecto se construye un software libre de uso enfocado en pequeñas y medianas empresas el cual permite la administración y el análisis de información de las bases de datos relacionales y no relacionales, este software es ajustable dependiendo de las necesidades de cada empresa [13].

- Propuesta de un sistema de información para mejorar los procesos del área de logística en Orange Business Services:

Se plantea una solución y es el diseño y construcción de un sistema de información, esto permite organizar y actualizar todos los procesos del área de logística como el registro, control y almacenamiento. [34]

- Desarrollo de un prototipo de un sistema de información para fincas productoras de café:

En este proyecto se desarrolla un prototipo que tiene como finalidad implementar un sistema de información en una finca productora de café ubicada en el departamento de Caldas, Colombia [35].

- Desarrollo de un sistema web de comercio electrónico B2C, para la promoción, compra on-line y gestión de stock de artículos de cuero:

Este proyecto logra el diseño, desarrollo e implementación de un comercio electrónico de una tienda de calzado y artículos de cuero llamada EMI, para ello dan todo un contexto teórico del problema para luego iniciar con el diseño y posterior desarrollo [36].

- Implementación de una herramienta web para la administración de proyectos, investigaciones, tesis e ideas:

Se propone la implementación de una herramienta web, que ayude a los estudiantes de la UNAB a conocer cuales proyectos de grado se han realizado anteriormente, esto incluye un sistema de gestión para conocer los proyectos que se están trabajando actualmente, en donde se evidencia información como el estado, avances realizados o los resultados obtenidos [31].

2.2 MARCO CONCEPTUAL

A lo largo del marco teórico, se hará referencia a conceptos importantes para tener en cuenta durante el proyecto, en ella se especificarán su significado, entre otra información que puede ser relevante para el lector. La mayoría de estos conceptos fueron aplicados a lo largo del documento y otros están más enfocados en conocimiento y que pueden servir para implementaciones futuras.

2.2.1 ARQUITECTURA CLIENTE-SERVIDOR

Esta es una arquitectura que permite que una maquina actuando como el cliente pueda obtener acceso a una información contenida en una máquina que actúa como servidor aun si son entornos multiplataforma. De manera más clara el funcionamiento se podría resumir así: Un cliente envía un mensaje a un servidor para solicitar un servicio (se realiza una petición) y el servidor responde enviando uno o más mensajes con la respuesta (da acceso al servicio).

Por el lado del cliente, está encargado de crear los requerimientos y solicitudes que tenga al servidor por medio de una capa de interfaz gráfica (GUI) o también conocida como Front end.

Por el lado del servidor puede atender diferentes solicitudes de clientes ya que ahí se almacenan los datos y reglas de la aplicación, a esto se le conoce como Back end [37].

2.2.2 FRONT END

Es el componente encargado de crear y almacenar las interfaces del software, esto quiere decir que es la parte visual del programa y la que está en constante interacción con el usuario, eso permite mejorar la experiencia del cliente ya que se pueden agregar letras, imágenes, colores y de más herramientas dinámicas, adicionalmente el frontend es el encargado de realizar las peticiones al backend. Para desarrollar esta parte generalmente se utilizan herramientas como HTML, CSS y JavaScript, las cuales en conjunto dan la mejor experiencia al cliente, su relación se puede visualizar de la siguiente manera [14]:

Ilustración 3 HTML, CSS, JS



TOMADO DE: <https://www.cursosgis.com/como-integramos-los-lenguajes-html-css-y-javascript/>

De igual manera también existen algunos frameworks que aumentan las posibilidades de interacción en la interfaz como, por ejemplo, Vue.js, Angular, React, Bootstrap, entre otros.

2.2.3 HTML

Sus siglas significan Hyper Text Markup Language, lo cual significa lenguajes de marcas de hipertexto, y es un tipo de lenguaje de marcado que brinda la posibilidad de estructurar una página web haciendo uso de etiquetas, este lenguaje puede ser interpretado por los navegadores y los sistemas operativos y así mostrar el formato que se desea. Este lenguaje es uno de los más

usados en la actualidad y debe ser utilizado a través de un editor de texto como visual studio o sublime text. Cada versión ofrece una mejora respecto a la anterior y actualmente se usa la versión HTML5.

El inicio de HTML se dio gracia a Tim Berners-Lee, un físico que trabajaba para CERN y que en 1989 quiso vincular una página a otra, esto permitiría el acceso a más información y permitió que naciera la idea del sistema del hipertexto. La versión actual que se maneja (HTML5) significo un cambio muy grande para el desarrollo Front End porque trajo la posibilidad de incluir audio, video e incluso MathML, un lenguaje encargado de realizar fórmulas matemáticas.

La composición de una página HTML es la de un conjunto de etiquetas ordenadas jerárquicamente, donde se organizan y representan los componentes básicos de la página como secciones, encabezados, párrafos, entre otros, su extensión es la .html y cualquier navegador permite su visualización.

Como cualquier otro lenguaje, cuenta con ventajas y desventajas, por ejemplo, y como se mencionó anteriormente, puede ser ejecutado de manera nativa por cualquier navegador, también se puede integrar con el backend de manera sencilla y no está de más mencionar que es uno de los lenguajes más utilizados en el mercado, sin embargo, entre sus desventajas, la más importante es que son paginas estáticas y que para darle dinamismo es necesario utilizar JavaScript [15].

2.2.4 CSS:

Sus siglas significan Cascading Style sheets, lo cual significa hojas de estilo en cascada, este es un lenguaje usado para estilizar la estructura codificada en HTML, el W3C (World wide Consortium) se encarga de especificar cuáles son las hojas de estilo estándar que pueden ser utilizada por los navegadores. Este lenguaje está diseñado por niveles y perfiles, lo que significa que cada nivel nuevo se agrega encima del anterior lo que permite añadir nuevas funciones [16].

2.2.5 JAVA SCRIPT:

Es un lenguaje de programación altamente usado, y que en el ámbito del front end puede utilizarse junto HTML y CSS, así brindar dinamismo e interactividad en la página web, este lenguaje pertenece a la categoría de scripting y está basado en ECMAScript, lo cual tiene una ventaja y es que se puede ejecutar en el cliente o en el servidor.

Su uso ha logrado tener tanta fuerza que una gran cantidad de navegadores cuentan con motores para el renderizado de JavaScript, esto representa tener la ventaja de ejecutar líneas de comando en un documento HTML sin utilizar un programa externo que lo haga. Como se mencionó anteriormente, JavaScript permite darle dinamismo al front end, y gracias a este lenguaje es que se observa que algunas páginas cuentan con automatización de proceso como el paso de imágenes de forma automática, o el autocompletado de texto, herramientas de diversos tipos que mejoran la experiencia del usuario.

Como se mencionó anteriormente, JavaScript es un lenguaje que se puede utilizar tanto en el frontend como en el backend, esto es una de las ventajas principales con las que cuenta, adicionalmente cuenta con una buena cantidad de frameworks que facilitan el SPA (Single page applications). De igual manera cuenta con una serie de desventajas como es la seguridad, y es que suele ser propenso a utilizarse con motivos maliciosos porque el código se ejecuta en el usuario y no en el servidor, una de las maneras para contrarrestarlo es desactivar JavaScript para no tener brechas de información con la complicación de perder el dinamismo de la página web [17].

2.2.6 FRAMEWORKS:

Esta es una de las herramientas más utilizadas por los desarrolladores, y es que un framework es un marco de trabajo que agiliza procesos de desarrollo y reduce la necesidad de escribir código, se puede interpretar como una plantilla que se sigue y que tiene predefinidos

elementos que se usan ya sean en frontend como en el backend, el uso de un framework representa una ventaja porque garantiza las buenas prácticas en el desarrollo de un proyecto, también ayuda a reducir el tiempo que se consume en el desarrollo de la aplicación y proporciona mayor seguridad teniendo en cuenta que las principales amenazas ya fueron solucionadas [18].

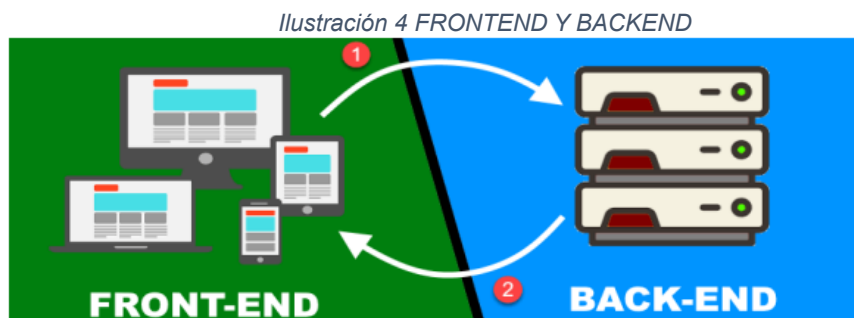
Como se mencionó anteriormente, los framework existen tanto para frontend como para backend y para mencionar algunos ejemplos hay los siguientes:

- React.js: Frameworks de JavaScript el cual se caracteriza por crear interfaces dinámicas e interactivas para los usuarios, permite organizar y reutilizar el código para ser eficiente y es de código abierto [18].
- Angular: Framework caracterizado por utilizar el diseño MVC (Modelo-vista-controlador) que permite generar enlaces de datos bidireccionales, enrutamiento, entre otros, es ideal para aplicaciones web robustas y escalables [18].
- Vue.js: Es un frameworks fácil de integrarse a proyectos existentes ya que su curva de aprendizaje es suave, además de que la sintaxis que utiliza es más simple, de igual manera es de código abierto [18].
- Express.js: Este framework es utilizado en aplicaciones web para Node.js, permite construir APIs de manera eficiente y rápida [18].
- Django: está basado en Python y permite el desarrollo de alto nivel debido a la eficiencia que maneja y en la posibilidad de la reutilización de código, trae incorporado herramientas como la generación de formularios, administración de bases o la autenticación [18].
- Ruby on Rails: Es un framework basado en el lenguaje de Ruby, está enfocado en la productividad y simplicidad, tiene estructuras predefinidas para la gestión de las bases de datos y la generación de código [18].

2.2.7 BACKEND:

Es el componente encargado de contener la lógica de la aplicación, en ella se codifica todos los procesos haciendo uso de un lenguaje programación, esta capa es la encargada de acceder al servidor y a las bases de datos, sin embargo, desde el punto de vista del cliente, esta capa es invisible ya que ellos no tienen acceso, siendo esta una de las ventajas principales de utilizar una arquitectura cliente – servidor. En pocas palabras el backend recibe las solicitudes de los clientes a través de la conexión con el front end, los procesa y envía la respuesta. Para el desarrollo de backend existen varios lenguajes como Python, Ruby, PHP, Java, entre otros lenguajes que establecen arquitecturas robustas y con alto nivel de seguridad, de igual manera se cuentan con frameworks como express.js, Django o Ruby on Rails.

La siguiente imagen relaciona el front end y el back end, donde se observa que el front end se ubica en los dispositivos de los clientes y el backend se ubica en los servidores [19].



TOMADO DE: <https://www.campusmvp.es/recursos/post/que-necesitas-saber-para-ser-un-buen-desarrollador-web-front-end.aspx>

2.2.8 PYTHON:

Es un lenguaje de programación desarrollado en Países bajos en los años 80, es un lenguaje de alto nivel que busca que sus instrucciones sean similares al inglés y así se mas legible el código, se caracteriza por ser un lenguaje multiplataforma, es decir, puede ser utilizado en sistemas operativos como Apple, Windows, Linux, entre otros. También es un lenguaje de

programación multiparadigma que permite la programación orientada a objetos o la imperativa, es de código abierto lo que significa que no requiere un pago para poder ser utilizado, es de tipado dinámico y cuenta con una alta cantidad de librerías y frameworks, aun así, una de sus principales desventajas es el alto consumo de memoria que tiene, y el hecho de que no detecta errores al momento de la compilación.

Otra gran ventaja es el hecho de ser un lenguaje de propósito general, por eso se encuentra gran variedad de aplicaciones en diferentes campos, por ejemplo, análisis de datos, aplicaciones de escritorio, desarrollo web, seguridad, inteligencia artificial, entre otros [20].

Según el índice TIOBE para junio del 2023, Python es el lenguaje de programación más utilizado, título que ostenta hace bastante tiempo, y que es seguido de lenguajes como C, C++ o Java [21].

Ilustración 5 Ranking lenguajes de programación.

Jun 2023	Jun 2022	Change	Programming Language	Ratings	Change
1	1		 Python	12.46%	+0.26%
2	2		 C	12.37%	+0.46%
3	4	▲	 C++	11.36%	+1.73%
4	3	▼	 Java	11.28%	+0.81%
5	5		 C#	6.71%	+0.59%

TOMADO DE: <https://www.tiobe.com/tiobe-index/>

La actual versión que se maneja de Python es la 3.11 lanzada en octubre del 2022 y se proyecta una nueva versión para el año en curso.

2.2.9 BASES DE DATOS

Las bases de datos son un depósito de datos, la cual, generalmente cuenta con una estructura que permite evidenciar como están relacionadas, esta información puede ser utilizada por los usuarios que la requiera siempre que tengan acceso a ella. Ahora bien, para garantizar la seguridad de la información, los datos deben estar independientes de cualquier otra cosa, donde también se especifique la definición del dato y su tipo. Una base de datos generalmente

cuenta con un formato establecido que se repite, en ella se encuentran los campos o las mismas columnas, registros o las mismas filas y finalmente los archivos que es la colección de los registros.

Durante el proyecto, se propuso el uso de bases de datos relacionales, esto debido a que la información que se maneja en el proceso está estructurada, lo que hace que sea más sencillo su almacenamiento, manejo y extracción cuando se encuentra dentro de una base de datos, adicional a ello es mucho más intuitiva. No se recomendó el uso de bases de datos no relacionales debido a que estas no manejan una estructura fija, y para el presente caso, los datos ya están estructurados lo que haría que, en vez de facilitar, complicaría más el manejo de la información.

Las bases de datos cuentan con ciertas características como, por ejemplo, cada registro cuenta con una clave única e irrepetible (Primary Key) esto permite que cada registro sea único, en otras palabras, evita que haya datos duplicados a la hora de hacer una consulta, algunas de las otras características se mencionan a continuación:

- Ingreso de varios usuarios de manera simultanea
- Acceso seguro a la información
- Posibilidad de realizar backup para recuperación de información
- Integridad de los datos
- Mínima redundancia
- Uso mediante lenguajes de programación estándar

Lo que se conoce como bases de datos se puede dividir en dos partes, en la primera se encuentran lo que son los datos, toda esta información obtenida de los usuarios y/o aplicaciones y que se almacena de manera estructurada y organizada, y por otra parte están los DBMS (DataBase Management System) los cuales son softwares especializados en el manejo y gestión de los datos y la información almacenada, es decir, estos son los programas que se usan para crear, leer modificar o borrar la información que se encuentra en una base de datos [19].

2.2.10 MODELOS DE BASES DE DATOS

Existen diferentes tipos de modelos, los cuales se diferencian principalmente por la representación lógica de los datos.

2.2.10.1 Base de datos jerárquica

Para el almacenamiento y gestión de la información se utiliza un patrón en forma de árbol, donde su orden depende de la importancia. Para este modelo se cuenta principalmente con 3 elementos:

- Raíz: Es el nivel superior y a partir del cual se desprenden los diferentes nodos, en este nivel se ubican el origen de los datos
- Padre: Son los nodos que están en el mismo nivel y tienen la misma importancia, a partir de ellos se desprenden los demás nodos de niveles inferiores.
- Hijo: Son los nodos que nacen a partir de la división del nodo padre y por lo tanto se encuentra en un nivel menor.

2.2.10.2 Base de datos estructurada en red

Tiene una estructura parecida al modelo jerárquico, sin embargo, se diferencia principalmente en que los nodos hijos pueden tener varios nodos padres en una relación de mucho a muchos, esta diferencia genera un beneficio importante y es que evita la redundancia de los datos.

2.2.10.3 Base de datos orientada a objetos

En este caso se basan en los lenguajes orientados a objetos, esto crea estructura de datos mucho más complejas pero que genera una ventaja, y es que elimina las barreras que hay al momento de desarrollar una aplicación y la gestión de sus datos.

2.2.10.4 Base de datos estática:

Estas bases de datos permiten almacenar la información de datos que han sido recolectados a través del tiempo, aunque únicamente se puede realizar consultas, con esta información ya se puede analizar y estudiar los datos.

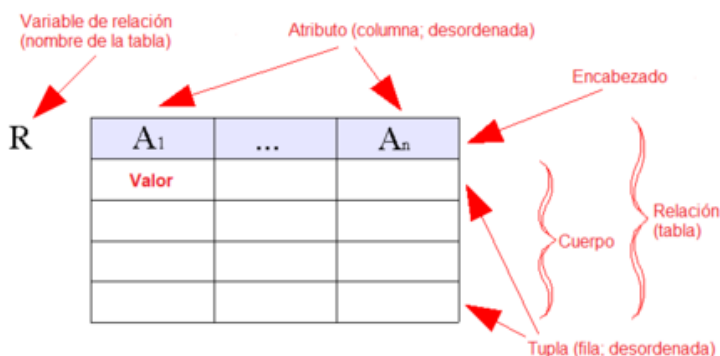
2.2.10.5 Base de datos dinámica:

Su principal diferencia con respecto a las bases de datos estáticas es que estas permiten hacer las operaciones CRUD, operaciones conocidas en el manejo de bases de datos y que permiten crear (C), leer (R), Actualizar (U) y eliminar (D) los datos que se encuentre en la base de datos.

2.2.10.6 Bases de datos relacionales:

Estas son las bases de datos más comunes, donde tanto los atributos como las entidades se pueden almacenar en tablas y así se convierten en registros o fila y en los campos o las columnas, el manejo de este elemento es más sencillo ya que se cuenta con un identificador único para cada registro.

Ilustración 6 MODELO RELACIONAL



TOMADO DE: https://www.wikiwand.com/es/Modelo_relacional

2.2.11 MODELO ENTIDAD-RELACIÓN:

Conocido como el modelo ER, es altamente utilizado en la actualidad ya que es de fácil uso e interpretación, modela los datos haciendo uso de diagrama y simbología, este modelo se encarga de representar gráficamente los elementos, también especificando las propiedades y relaciones entre los objetos. Está compuesto principalmente por las entidades, atributos, relaciones y tipos de relaciones [19].

En el caso de las entidades se hace referencia a los objetos que pueden ser incluidos en una base de datos, acá hay dos tipos de entidades:

- Los objetos con existencia física o con entidad concreta, los cuales justamente son las cosas que existen y son tangibles, como un lápiz, una moto, un celular, etc.
- Los objetos con existencia conceptual o con entidad abstracta, son los elementos intangibles como los grados escolares.

Su representación en el diagrama es a través de los rectángulos.

Por otro lado, también están los atributos, los cuales son las cualidades de las entidades, son las propiedades y con ellas se puede dar información relevante sobre cada entidad, existen diferentes tipos de atributos:

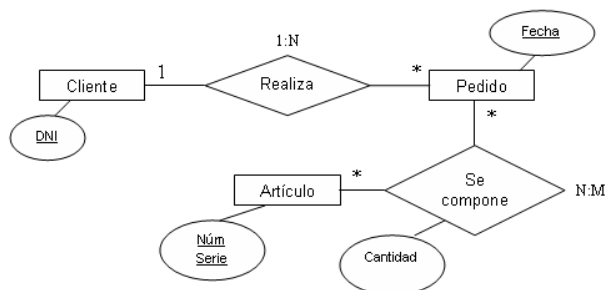
- Atributo clave: Es un valor único el cual permite reconocer a un único registro, es un valor que no se debe repetir, en las DBMS se suele conocer como Primary Key o llave primaria.
- Clave externa o foránea: Este atributo es utilizado cuando relaciono dos entidades, en las BDMS se conoce como Foreign Key o clave foránea y puedo relacionar las entidades únicamente haciendo uso de la clave principal de la otra entidad a la que voy a relacionar.

En este modelo también existen las relaciones, y son los enlaces que se forman de la asociación entre diferentes entidades, para que estas relaciones se formen, las entidades deben compartir ciertos atributos, y esta representación en el diagrama se hace mediante un rombo

De igual manera existen varios tipos de relaciones los cuales se pueden llamar relaciones de cardinalidad y son los siguientes:

- Relación uno a uno (1:1): Es el caso cuando un elemento de la entidad A solo se relaciona con otro elemento de la entidad B
- Relación una a muchos (1: N): Es el caso cuando un elemento de la entidad A puede estar relacionada con uno o varios elementos de la entidad B
- Relación muchos a muchos (N:M): es el caso cuando varios elementos de la entidad A se relacionan con varios elementos de la entidad B

Ilustración 7 MODELO ENTIDAD RELACIÓN



TOMADO DE: https://es.wikipedia.org/wiki/Modelo_entidad-relaci%C3%B3n

2.2.12 SQL (Structured Query Language):

Este es un lenguaje común de consultas con el cual se puede manejar y hacer uso de las bases de datos, este lenguaje también es enunciativo y es utilizado en las bases de datos

relacionales y cuenta con diferentes tipos de operaciones para administrar la información que contienen, específicamente tres lenguajes con propósitos diferentes:

- DML (Data Definition Lenguaje) Encargado de llenar la información de las tablas
- DCL (Data Control Lenguaje) Otorga los permisos de entrada a las demás bases de datos
- DDL (Data control Lenguaje) Se encarga de definir la estructura de la tabla, sin embargo, no llena la tabla

2.2.13 SISTEMAS GESTORES DE BASES DE DATOS RELACIONALES:

Como se mencionó anteriormente, existen programas encargados de gestionar las bases de datos y con la capacidad de cumplir con las operaciones que puede hacer una base de datos, para el caso de las bases de datos relacionales estas son las principales:

- MySQL: Es uno de los gestores más utilizados por los sitios web debido a su alta velocidad en el manejo de la información, sin embargo, cuenta con la desventaja de que no se recomienda con grandes cantidades de datos.
- Microsoft SQL Server: Se caracteriza por permitir el acceso a varios usuarios de manera simultánea sin ningún problema, pero su principal limitación radica en que es una aplicación de pago.
- SQLite: Es una biblioteca de C que es altamente utilizada en aplicaciones pequeñas, es de código abierto y su limitación es que no admite grandes cantidades de datos.
- Oracle: Uno de los softwares más populares y usado por las grandes organizaciones ya que permite controlar y administrar bases de datos sin importar la cantidad, sin embargo, también es un software de pago.
- MariaDB: Es basado en MYSQL, lo que significa que comparten la mayoría de las características y limitaciones

- PostgreSQL: Es un software bastante utilizado ya que es de código abierto y combina diferentes características a la hora del almacenamiento y la seguridad de los datos, lo que lo convierte en un software ideal en la mayoría de los casos

Ilustración 8 GESTORES DE BASES DE DATOS



TOMADO DE: <https://www.diegocalvo.es/wp-content/uploads/2017/11/Base-de-Datos-Relacionales.png>

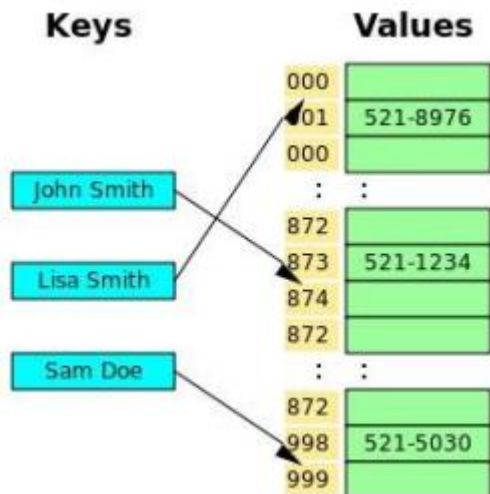
2.2.14 BASES DE DATOS NO RELACIONALES

Otro tipo de bases que se suelen manejar son conocidas como no relacionales o NoSQL, en este caso no se encuentra con una estructura de datos definida, lo cual en diferentes casos es bastante conveniente y se suele integrar en entorno aptos y operativos en todo momento, y también se utiliza debido a que maneja grandes cantidades de datos. De igual manera y como su nombre lo dice no utiliza lenguaje SQL [19].

2.2.14.1 Modelo Clave-valor:

Este modelo es el más utilizado debido a su sencillez, se caracteriza porque cada elemento se identifica como un conjunto de pares clave- valor en donde cada clave debe ser un identificador único. En cuanto a su contenido no hay problema porque tanto las claves como el valor pueden ser objetos simples o compuestos, hacer uso de este modelo trae como beneficio una recuperación de la información de manera muy rápida y efectiva tanto en lectura como escritura [23].

Ilustración 9 MOELO CLAVE-VALOR



TOMADO DE: <https://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>

Al igual que con las bases de datos relacionales, hay diferentes softwares que permiten gestionar de manera correcta a las bases de datos no relacionales [23]:

- Cassandra: Es un software NoSQL que se caracteriza por una alta velocidad y desempeño a la hora de manejar los datos incluido la importación y exportación. En este caso cuenta con un lenguaje propio de consultas conocido como CQL (Cassandra Query Language) y fue diseñada en Java.
- MongoDB: Es de código abierto lo cual representa un gran beneficio, adicional es de tipo orientada a documentos, y se caracteriza por manejar esquemas de datos diferentes para cada caso al punto de que ningún esquema de datos sea igual al resto de registros almacenados, para almacenar información utiliza un lenguaje conocido como BSON el cual es una evolución de JSON, pero en este se pueden almacenar datos binarios, dadas estas razones MongoDB es uno de los más utilizados.

Ilustración 10 LENGUAJES NOSQL



TOMADO DE: <https://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>

2.2.15 UML (Lenguaje Unificado de Modelado)

Este lenguaje está basado en la construcción de diagramas para la especificación, visualización, construcción y documentación de cualquier sistema, especialmente del software. Las ventajas de utilizarlos radican en el hecho de que es una simplificación de la realidad, con esto se puede comprender de mejor manera que es lo que se va a desarrollar y así obtener una visión global hacia donde se debe dirigir, UML está orientado a objetos por eso es más común usarlo cuando los softwares son orientados a objetos. Algunos de los diagramas más comunes que se utilizan son [25]:

Según su estructura:

- Diagrama de objetos
- Diagrama de paquetes
- Diagramas de clases
- Diagrama de despliegue
- Diagrama de componentes

- Diagrama de estructura

De comportamiento:

- Diagrama de secuencia
- Diagrama de máquina de estados
- Diagrama de casos de uso
- Diagrama de actividad
- Diagrama de tiempos
- Diagrama de comunicación
- Diagrama global de interacciones

Como se puede observar, UML contiene 13 tipos de diagrama, sin embargo, en el presente trabajo se utilizaron los diagramas de casos de uso, diagrama de despliegue y diagrama de actividades o también conocido como modelo de navegación.

2.2.16 MODELOS DE DESARROLLO WEB

El desarrollo de software es más que escribir solo código, requiere entender cuáles son las necesidades reales del cliente y de esta manera determinar si la creación de un software es lo más conveniente según su necesidad, con esto en mente surgen las metodologías las cuales son un conjunto de herramientas y técnicas las cuales ayudan a diseñar, planificar, desarrollar, implementar y mantener un software [24].

Como la mayoría de las veces, existen diferentes metodologías cada una cuenta con ventajas y desventajas:

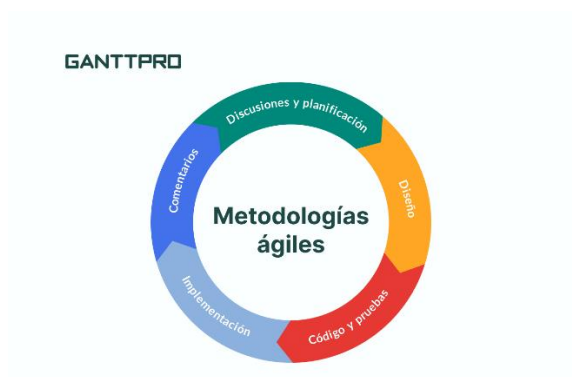
2.2.16.1 Metodologías ágiles para la gestión de proyectos:

Se fundamenta principalmente en el trabajo en equipo y la iteración esto permite reducir el tiempo de entrega de un producto sin dejar de lado la alta calidad, hay diferentes metodologías y a continuación se explican las principales:

- Metodología Kanban: Nació en los años cuarenta gracias a la industria automotriz japonesa y su nombre se deriva de la palabra “Tablero”, el objetivo principal de esta metodología es alcanzar un flujo de trabajo alto en donde se priorice la alta calidad del software, para ello utilizan una técnica conocida como limitación del trabajo en progreso (WIP) en donde no se puede comenzar nuevos trabajos hasta no terminar el actual, esto lo hacen para evitar la sobrecarga de trabajo y mejorar el flujo.
- Metodología Scrum: Esta es una metodología que se caracteriza por la entrega de productos de alta calidad haciendo uso de procesos iterativos e incrementales, para ello el equipo de trabajo se divide en subgrupos que trabajan de manera autónoma en partes pequeñas del producto, finalmente el equipo total de trabajo se reúne para discutir y compartir los resultados de cada tarea, en scrum se maneja un ciclo de vida que consta de lo siguiente:
 - Product Backlog: Son todos los requisitos que debe llevar el producto para que satisfagan la necesidad del cliente
 - Sprint Backlog: Son listas de tareas que se deben desarrollar para cumplir con los requerimientos y que se realizan dentro de un determinado tiempo conocido como sprint
 - Desarrollo: Es el cumplimiento de las tareas asignadas a cada subgrupo de trabajo

- Prueba: Son la realización de las diferentes pruebas para comprobar el resultado y si cumple con lo solicitado
- Implementación: Es la puesta en funcionamiento y posterior entrega al cliente.

Ilustración 11 METODOLOGIAS AGILES



TOMADO DE: <https://blog.ganttpro.com/es/introduccion-a-las-metodologias-agiles-1/>

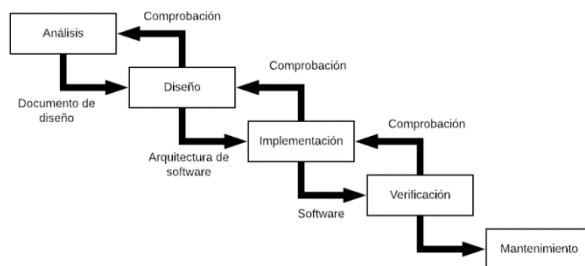
2.2.16.2 Metodologías tradicionales

- Metodología de cascada mejorado:

La principal característica de esta metodología es que sigue unos pasos muy estrictos y que además se deben cumplir en orden esto causa que el margen del equipo de desarrollo sea muy pequeño, sin embargo, es una metodología que sigue siendo utilizada en proyectos grandes cuando cuenta con una planificación muy detallada, tanto en su análisis de requerimientos como en el diseño y en el que no se espera que haya muchos cambios.

Como se aprecia en la siguiente imagen, está constituida principalmente por 5 fases: Análisis, diseño, implementación, verificación y mantenimiento, y se observa cómo va en tipo de cascada ya que una fase depende de la anterior [31].

Ilustración 12 Metodología de cascada mejorado



TOMADO DE:

https://repository.unab.edu.co/bitstream/handle/20.500.12749/14408/2021_Tesis_Pedro_Jota_Regueros.pdf?sequence=1

- ✓ **Análisis:** Como su nombre lo indica, esta etapa está enfocada en la realización del análisis y definición de los requerimientos y requisitos, para lograr esto se hacen reuniones con los clientes y/o usuarios ya que ellos son el insumo principal para determinar las necesidades y restricciones del sistema.

Para el presente proyecto se realizaron reuniones con los miembros encargados de ejecutar la tarea de Carterización, donde explicaron el funcionamiento del mismo y cuáles eran las principales dificultades.

- ✓ **Diseño:** Esta fase permite diseñar el sistema diferenciando elementos de software y hardware necesarios a partir de la recolección de la información del paso anterior.

En este paso se realizó la construcción del modelo de base datos, la creación de los prototipos de interfaces de usuario, el modelo de navegación y el diagrama de despliegue.

- ✓ **Implementación:** Esta fase ejecuta la fase de diseño a través de la programación del software, posterior a ello se realiza la búsqueda de errores y las pruebas respectivas. La programación del software se realiza en componentes separados, así como las pruebas para lograr una integración gradual al software final

- ✓ **Verificación:** Esta fase permite la integración del software en un ambiente específico que permite medir el rendimiento de este, gracias a esto se realizan unas pruebas de aceptación que determinan si el software cumple con las exigencias mínimas requeridas que se definieron anteriormente.
- ✓ **Mantenimiento:** Una vez el software está listo se hace entrega de este, pero igual se debe estar pendiente en caso de algún tipo de fallas no descubiertas, además permite mejorar el sistema para darle mayor robustez.

El uso de esta metodología presenta algunos beneficios como:

- ✓ Producción de resultados predecibles que cumplan con lo solicitado por el usuario
- ✓ Ser una metodología ideal en los casos que se conoce claramente lo que busca el cliente y no se presenten grandes cambios.

También presenta unas desventajas como:

- ✓ Es difícil la introducción de cambios
- ✓ El producto se observa hasta el final cuando ya esté terminado
- ✓ Los requerimientos se deben fijar desde un principio

Debido a los diferentes beneficios que trae, este fue el método que se utilizó para el desarrollo del proyecto también la decisión se basó en que el proyecto cuenta con un solo miembro y este tipo de metodologías favorecen a las personas que trabajan sin equipo. Sin embargo, se mencionan diferentes tipos de metodologías a manera de ampliación del tema para el lector.

A lo largo de la definición de los pasos, se evidencio como el proyecto utilizo la respectiva fase para cumplir su objetivo, se puso en práctica las dos primeras fases debido a que el proyecto cubre el diseño de una aplicación web, para poner en marcha las demás fases ya se debe tener

en cuenta la retroalimentación de la empresa Telefónica para determinar su satisfacción con el diseño propuesto y su posterior aprobación.

- Metodología DevOps:

Cuando el tiempo de producción del software no es muy grande esta metodología puede ser ideal, ya que se orienta a la colaboración y coordinación entre el área de desarrollo y de operaciones para así tener un lanzamiento rápido sin descuidar la calidad del producto final.

- Metodología Lean:

Esta metodología está basada en la filosofía Lean en la cual el objetivo es mejorar la calidad y la eficiencia del producto acompañado de una reducción del tiempo, para esto se basan en el principio de entregar lo más valioso lo antes posible y minimizar los desperdicios que se puedan generar.

- Metodología en espiral

Su enfoque iterativo e incremental la convierte en una metodología interesante para cualquier proyecto, en ella se aborda el riesgo de manera sistemática y así minimizándolo, adicionalmente otorga la integración del desarrollo, las pruebas y la validación del producto, a su vez funciona como un marco referencial para el proyecto y con cada iteración se vuelve más robusto el producto que se entrega, se divide en cuatro fases:

- Inicio
- Crecimiento
- Maduración
- Declive

A su vez cada fase se divide en las siguientes subfases:

- Planificación
- Diseño
- Implementación
- Prueba

Ilustración 13 METODOLOGIA EN ESPIRAL



TOMADO DE: R. S. Pressman, Ingeniería del Software - Un Enfoque Practico 7ma Edicion. McGraw-Hill Companies.

- Metodología de prototipo

En esta metodología se desarrolla un prototipo que permita entender de mejor manera cuales son las necesidades del cliente, a partir de allí se empieza construir el software completo, con esto se busca reducir el costo del proyecto y el tiempo de ejecución.

- Desarrollo rápido de aplicación (RAD)

Es este caso se minimiza el tiempo que se consume en etapas con el diseño, análisis, codificación para producir un prototipo funcional en donde lo fundamental es gastar el menor tiempo posible

- Metodología de programación extrema (XP)

En este caso se busca producir software de muy alta calidad, eficiente, y que sea adaptable al cambio, para ello se utilizan un conjunto de prácticas de desarrollo el cual es iterativo e incremental, algunas de las practicas que se utilizan son la programación en parejas, planificación muy corta, pruebas rápidas y extremas entre otros.

2.2.17 MOTOR DE REGLAS

En diversas aplicaciones, es necesario cambiar o modificar la lógica de una aplicación según sea conveniente, pero para esto es necesario hacer uso de una persona especializada en el tema (programador) lo cual representa una demanda de tiempo importante así como un gasto monetario adicional, para estos casos surgen los motores de regla de negocio (BRE) los cuales son componentes de software que permiten agregar, modificar o eliminar las reglas de negocio sin tener conocimientos como programador. Para el funcionamiento del motor se debe separar la ejecución del código de reglas de los demás componentes del sistema, para que así actúe como un componente independiente y pueda efectuar los cambios sin afectar a las demás normas del sistema [29].

Un motor de reglas completo suele contar con las siguientes características:

- Repositorio de reglas: Base de datos donde se almacenarán las reglas de negocio creadas por los usuarios
- Editor de reglas de negocio: Se debe construir un interfaz de usuario intuitiva para el usuario, en donde prime la claridad para que el usuario tenga facilidad para la construcción de las reglas.
- Informe de los componentes: Una interfaz de usuario intuitiva para que los usuarios consulten de manera clara las normas actuales.

2.2.18 MOCKUPS

Son fotomontajes que dan una idea clara de cómo se vería el producto final terminado, orienta al diseñador o programador a cuáles son las modificaciones que debe hacer para cumplir con las expectativas del cliente, los beneficios de usar mockups son de incrementar la satisfacción del cliente y ahorrar tiempo y dinero en el desarrollo del proyecto [26].

2.2.19 FIGMA

Es un software de edición grafica ideal para la creación de mockups y cualquier tipo de diseños gráficos que den una idea clara de cómo se podría ver el producto terminado [27].

3 DESARROLLO Y RESULTADOS.

Como se aclaró en la sección de la metodología, se utilizó el modelo en cascada mejorado junto con la metodología ágil de scrum para el desarrollo del proyecto. Las razones de esta elección están basadas en lo siguiente:

- ✓ Usar estas metodologías permite una determinación clara de los requisitos
- ✓ El modelo de desarrollo en cascada mejorado permite la retroalimentación de las fases anteriores.
- ✓ Este modelo sigue siendo muy utilizado a pesar de su longevidad debido a los buenos resultados que se obtienen
- ✓ Al ser un proyecto de una sola persona, permite una fácil comprensión de su funcionamiento.
- ✓ La integración de estas dos metodologías permite tener un flujo de trabajo controlado mientras se cumple con las fases.

La metodología de cascada mejorado se caracteriza por tener 5 fases principales: Análisis, diseño, implementación, verificación y mantenimiento. Para el actual proyecto se

tuvieron en cuenta las dos primeras fases. A continuación, se describe como fueron aplicadas y que actividades fueron desarrolladas:

3.1 ETAPA 1 ANALISIS

Esta etapa se caracteriza por tener como propósito principal identificar las necesidades del aplicativo, esto se suele conocer como levantamiento de requerimientos y teniendo en cuenta esto y con base a las metodologías que se utilizan se determinó utilizar la herramienta de historias de usuario o user story, la cual es un formato que resume la necesidad del cliente y que deben ser cubiertas a través de funcionalidades del software.

El levamiento de requerimientos se hizo de manera conjunta entre el equipo de desarrollo y el cliente, el cual expuso sus necesidades para que sean cubiertas. Para el proyecto actual no se contó con un cliente como tal, sino que se utilizó como insumo de información a los compañeros encargados de realizar el proceso, para ello se realizaron algunas reuniones haciendo uso de herramientas de comunicación interna como Teams y otras reuniones informales a través del diálogo directo.

3.1.1 REQUERIMIENTOS

Dentro del área de B2B de Telefónica hay un proceso importante conocido como carterización el cual se encarga de asignar un modelo de atención a un determinado cliente, en otras palabras, si un determinado cliente quiere comunicarse con Telefónica tiene una persona que le va a colaborar con todas las solicitudes que necesiten. El proceso tiene las siguientes características:

- ✓ A los encargados del proceso les comparten una base de clientes con toda la información de este, por ejemplo, NIT, nombre, segmento, promedio de facturación, etc.

- ✓ También se comparte una base con los analistas disponibles, esta base debe ser segmentada en diferentes cuadrillas o tríos, las cuales se le asignan un número ya sea para el servicio que está destinada o por el aliado que atiende.
- ✓ Dentro del área se manejan 3 segmentos de atención principales conocidos como Corporaciones, empresas y Pymes, cada nueva empresa que contrata servicios de telefónica se asigna a uno de estos segmentos dependiendo de su facturación, cantidad de servicios contratados, entre otros aspectos que se definen de manera interna en la compañía
- ✓ La caracterización se hace principalmente a estos 3 segmentos de manera separa, esto debido a que cada segmento cuenta con sus propias reglas, por ejemplo, los analistas disponibles para hacer la atención de pymes son diferentes a los analistas que realizan la atención en corporaciones.
- ✓ Adicional a los segmentos de atención, también se debe tener en cuenta el tipo de servicio con el que cuentan, pueden ser productos de fija, productos de móvil o productos convergentes que es cuando se contratan ambos tipos de servicio.
- ✓ Dentro de la compañía se hace uso de algunos aplicativos para la atención al usuario, estos son los canales de atención, que a su vez se dividen en canales tradicionales y canales digitales, en los canales tradicionales se encuentra Scriba y en los digitales está Mcare e IVR.
- ✓ Como se mencionó anteriormente hay servicios de fija los cuales son atendidos a través de Scriba, en el caso de móvil se atienden en Mcare y el caso de convergente se atiende a través del IVR.
- ✓ Esta información es de alta importancia porque existen diferentes tipos de caracterización los cuales varían en sus reglas, en sus clientes, y en su modelo de atención.

- ✓ Actualmente esta información se maneja a través de archivos Excel y todo su proceso ocurre haciendo uso de esta herramienta, sin embargo, su principal problema radica en el cuidado que se debe tener para evitar que los clientes queden asignados a personal incorrecto o que incluso no queden asignados, entre otros problemas que al final representan un alto uso de tiempo.

A continuación, se adjunta una imagen proveniente de un archivo de Excel el cual contiene a los usuarios y toda la información importante de ellos, en ese archivo es donde se adjunta el modelo de atención, en pocas palabras, agrega el nombre del analista que lo va a atender y la información de este. Por temas de confidencialidad de la empresa y el tratamiento de datos, se censura la información, sin embargo, la imagen sirve de contexto para entender cómo se realiza el proceso.

Ilustración 14 Contextualización carterización

	A	B	C	D	E	F	G	H	I	J	K	L	M
	NIT	NIT_ORG	M Salesforce	SEGMENTO	SEGMENTO ATENCIÓN	SEGMENTO SALESFORCE	SUBSEGMENTO LOCAL	SECTOR (ESTRUCTURA)	MULTINACIONAL	Actividad Económica	TIPO CLIENTE	Cliente en More	PERIODO
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													
21													
22													
23													
24													
25													
26													
27													
28													
29													
30													
31													
32													
33													
34													
35													
36													
37													
38													
39													
40													
41													
42													
43													
44													
45													
46													
47													
48													
49													
50													

Fuente: Creación propia

3.1.2 HISTORIA DE USUARIO

A partir de esta información recogida, se procede a enlistar las user story, las cuales facilitaran el entendimiento de los requerimientos solicitados. Para las historias de usuario se usará el siguiente formato:

Ilustración 15 Formato historia de usuario

HISTORIA DE USUARIO			
ID		USUARIO	
NOMBRE			
PRIORIDAD			
DESCRIPCIÓN			
VALIDACIÓN			

Fuente: Creación propia

Donde:

- ✓ **ID:** Es un identificador único para cada historia de usuario.
- ✓ **Usuario:** Usuario al cual le pertenece la historia de usuario.
- ✓ **Nombre:** Nombre corto que describe la historia de usuario.
- ✓ **Prioridad:** Es la importancia que tiene la historia de usuario para el desarrollo del proyecto, puede ser de valor: Alta, media o baja.
- ✓ **Descripción:** Es la descripción de la idea principal del usuario.
- ✓ **Validación:** Condiciones que deben cumplirse para que la historia de usuario se logre finalizar.

A continuación, se presenta la historia de usuario número 1:

Ilustración 16 Historia de usuario 01

HISTORIA DE USUARIO			
ID	01	USUARIO	Administrador
NOMBRE	Alta de usuarios		
PRIORIDAD	Alta		
DESCRIPCIÓN	Como administrador quiero determinar cuales usuarios pueden ingresar a la aplicación		
VALIDACIÓN	*Quiero agregar nuevos usuarios para que puedan entrar a la aplicación *Para agregar nuevos usuarios voy a utilizar el usuario de red del trabajador que desea tener acceso a la aplicación *La contraseña de ingreso de cada colaborador también será la misma que la utilizada en el usuario de red		

Fuente: Creación propia

Se presenta la historia de usuario número 2:

Ilustración 17 Historia de usuario 02

HISTORIA DE USUARIO			
ID	02	USUARIO	Administrador
NOMBRE	Consulta de usuarios		
PRIORIDAD	Media		
DESCRIPCIÓN	Como administrador quiero conocer cuales usuarios tienen acceso a la aplicación		
VALIDACIÓN	*Quiero un listado de todos los usuarios que tienen acceso, donde se detalle: -Fecha de alta en la aplicación -Fecha del ultimo ingreso -Perfil en la aplicación		

Fuente: Creación propia

Se presenta la historia de usuario número 3:

Ilustración 18 Historia de usuario 03

HISTORIA DE USUARIO			
ID	03	USUARIO	Administrador
NOMBRE	Baja de usuarios		
PRIORIDAD	Alta		
DESCRIPCIÓN	Como administrador quiero determinar cuales usuarios ya no pueden ingresar a la aplicación		
VALIDACIÓN	*Quiero dar de baja a los usuarios que ya no necesitan tener acceso a la aplicación *En el listado de los usuarios actuales, necesito un botón que permita eliminar los permisos de los usuarios *Es necesario la confirmación de la acción		

Fuente: Creación propia

Se presenta la historia de usuario número 4:

Ilustración 19 Historia de usuario 04

HISTORIA DE USUARIO			
ID	04	USUARIO	Administrador
NOMBRE	Consulta de carterización		
PRIORIDAD	Baja		
DESCRIPCIÓN	Como administrador quiero ver la carterización actual		
VALIDACIÓN	*Quiero tener una previsualización del archivo donde esta la carterización *Quiero tener acceso a todas las carterizaciones		

Fuente: Creación propia

Se presenta la historia de usuario número 5:

Ilustración 20 Historia de usuario 05

HISTORIA DE USUARIO			
ID	05	USUARIO	Administrador
NOMBRE	Consulta de estadísticas		
PRIORIDAD	Baja		
DESCRIPCIÓN	Como administrador quiero ver las estadísticas de los usuarios carterizados		
VALIDACIÓN	*Quiero tener acceso a las estadísticas donde se reflejen datos como cantidad de usuarios, cantidad de datos por aliados, entre otros. *Las estadísticas deben aparecer dependiendo de los segmentos de atención .		

Fuente: Creación propia

Se presenta la historia de usuario número 6:

Ilustración 21 Historia de usuario 06

HISTORIA DE USUARIO			
ID	06	USUARIO	Administrador
NOMBRE	Descarga de carterización		
PRIORIDAD	Baja		
DESCRIPCIÓN	Como administrador quiero descargar los archivos donde esta la carterización		
VALIDACIÓN	*Quiero tener un botón que permita descargar la carterización que estoy consultando *El archivo debe ser de tipo (.TXT) o (.XLSX)		

Fuente: Creación propia

Se presenta la historia de usuario número 7:

Ilustración 22 Historia de usuario 07

HISTORIA DE USUARIO			
ID	07	USUARIO	Consultador
NOMBRE	Consulta de carterización		
PRIORIDAD	Alta		
DESCRIPCIÓN	Como consultador quiero ver la carterización actual		
VALIDACIÓN	*Quiero tener una previsualización del archivo donde esta la carterización *Quiero tener acceso a todas las carterizaciones		

Fuente: Creación propia

Se presenta la historia de usuario número 8:

Ilustración 23 Historia de usuario 08

HISTORIA DE USUARIO			
ID	08	USUARIO	Consultador
NOMBRE	Consulta de estadísticas		
PRIORIDAD	Media		
DESCRIPCIÓN	Como consultador quiero ver las estadísticas de los usuarios carterizados		
VALIDACIÓN	*Quiero tener acceso a las estadísticas donde se reflejen datos como cantidad de usuarios, cantidad de datos por aliados, entre otros. *Las estadísticas deben aparecer dependiendo de los segmentos de atención .		

Fuente: Creación propia

Se presenta la historia de usuario número 9:

Ilustración 24 Historia de usuario 09

HISTORIA DE USUARIO			
ID	09	USUARIO	Consultador
NOMBRE	Descarga de carterización		
PRIORIDAD	Media		
DESCRIPCIÓN	Como consultador quiero descargar los archivos donde esta la carterización		
VALIDACIÓN	*Quiero tener un botón que permita descargar la carterización que estoy consultando *El archivo debe ser de tipo (.TXT) o (.XLSX)		

Fuente: Creación propia

Se presenta la historia de usuario número 10:

Ilustración 25 Historia de usuario 10

HISTORIA DE USUARIO			
ID	10	USUARIO	Modificador
NOMBRE	Generador de carterización		
PRIORIDAD	Alta		
DESCRIPCIÓN	Como modificador quiero generar la carterización		
VALIDACIÓN	*Quiero tener la posibilidad de subir el archivo con los clientes a carterizar *Quiero tener la posibilidad de subir el archivo con los analistas disponibles *El archivo debe ser de tipo (.TXT) o (.XLSX)		

Fuente: Creación propia

Se presenta la historia de usuario número 11:

Ilustración 26 Historia de usuario 11

HISTORIA DE USUARIO			
ID	11	USUARIO	Modificador
NOMBRE	Crear reglas de carterización		
PRIORIDAD	Alta		
DESCRIPCIÓN	Como modificador quiero crear las reglas de la carterizacion		
VALIDACIÓN	*Quiero establecer las reglas con las que va a carterizar y así el software sepa a donde va a asignar cada analista		

Fuente: Creación propia

Se presenta la historia de usuario número 12:

Ilustración 27 Historia de usuario 12

HISTORIA DE USUARIO			
ID	12	USUARIO	Modificador
NOMBRE	Consultar las reglas de carterización		
PRIORIDAD	Alta		
DESCRIPCIÓN	Como modificador quiero consultar cuales son las reglas de la carterizacion		
VALIDACIÓN	*Quiero tener una lista con las reglas actuales que se manejan para el segmento que estoy consultando *Adicional a la regla se debe tener registro de cuando fueron creadas		

Fuente: Creación propia

Se presenta la historia de usuario número 13:

Ilustración 28 Historia de usuario 13

HISTORIA DE USUARIO			
ID	13	USUARIO	Modificador
NOMBRE	Cambiar reglas de carterización		
PRIORIDAD	Alta		
DESCRIPCIÓN	Como modificador quiero cambiar las reglas de la carterizacion		
VALIDACIÓN	*Quiero modificar las reglas con las que va a carterizar que están previamente creadas		

Fuente: Creación propia

Se presenta la historia de usuario número 14:

Ilustración 29 Historia de usuario 14

HISTORIA DE USUARIO			
ID	14	USUARIO	Modificador
NOMBRE	Eliminar reglas de carterización		
PRIORIDAD	Alta		
DESCRIPCIÓN	Como modificador quiero eliminar las reglas de la carterizacion		
VALIDACIÓN	*Quiero eliminar las reglas con las que va a carterizar que están previamente creadas		

Fuente: Creación propia

Se presenta la historia de usuario número 15:

Ilustración 30 Historia de usuario 15

HISTORIA DE USUARIO			
ID	15	USUARIO	Modificador
NOMBRE	Consulta de carterización		
PRIORIDAD	Alta		
DESCRIPCIÓN	Como Modificador quiero ver la carterización actual		
VALIDACIÓN	*Quiero tener una previsualización del archivo donde esta la carterización *Quiero tener acceso a todas las carterizaciones		

Fuente: Creación propia

Se presenta la historia de usuario número 16:

Ilustración 31 Historia de usuario 16

HISTORIA DE USUARIO			
ID	16	USUARIO	Modificador
NOMBRE	Consulta de estadísticas		
PRIORIDAD	Alta		
DESCRIPCIÓN	Como Modificador quiero ver las estadísticas de los usuarios carterizados		
VALIDACIÓN	*Quiero tener acceso a las estadísticas donde se reflejen datos como cantidad de usuarios, cantidad de datos por aliados, entre otros. *Las estadísticas deben aparecer dependiendo de los segmentos de atención.		

Fuente: Creación propia

Se presenta la historia de usuario número 17:

Ilustración 32 Historia de usuario 17

HISTORIA DE USUARIO			
ID	17	USUARIO	Modificador
NOMBRE	Descarga de carterización		
PRIORIDAD	Alta		
DESCRIPCIÓN	Como Modificador quiero descargar los archivos donde esta la carterización		
VALIDACIÓN	*Quiero tener un botón que permita descargar la carterización que estoy consultando *El archivo debe ser de tipo (.TXT) o (.XLSX)		

Fuente: Creación propia

3.1.3 REQUISITOS FUNCIONALES

A continuación, se determinan cuáles son los requisitos funcionales de la aplicación, estos se determinan a partir de la información otorgada por los encargados del proceso.

Tabla 2 Requisitos funcionales

Código	Clase	Requerimiento	Descripción	Restricción	Responsable
RF1	Funcional	Alta de usuarios	Cuando es necesario que una nueva persona tenga acceso a la aplicación, se le debe conceder su respectivo permiso, también se le debe asignar un perfil y dependiendo de esto el nuevo usuario tendrá diferentes tipos de permisos.	El usuario se debe encontrar registrado en la base de datos de Telefónica y contar con correo y usuario de red	Administrador
RF2	Funcional	Consulta de usuarios	En ocasiones, es necesario conocer cuáles son los usuarios que pueden ingresar a la aplicación, para esto se debe contar con la posibilidad de conocer cuáles son, su perfil y su fecha de ingreso.	EL usuario se debió dar de alta previamente	Administrador

RF3	Funcional	Baja de usuarios	Cuando un usuario no necesita más ingresar a la aplicación, lo mejor es quitarle el acceso para evitar fugas de información.	EL usuario se debió dar de alta previamente	Administrador
RF4	Funcional	Consulta de carterización	Se debe permitir el acceso a la carterización actual sin importar el segmento	Ninguna	Administrador, consultador y modificador
RF5	Funcional	Descarga de carterización	Se cuenta con la opción de descargar la carterización que se está visualizando para consultar la información de manera completa	Ninguna	Administrador, consultador y modificador
RF6	Funcional	Consulta de estadísticas	La aplicación debe contar con una sección donde se muestren diferentes tipos de estadísticas que sirvan de información para el lector	Ninguna	Administrador, consultador y modificador
RF7	Funcional	Generador de carterización	Se debe tener la opción de generar la carterización para eso se debe validar que estén los archivos necesarios cargados en la aplicación al igual que configuradas las reglas	Únicamente lo puede desarrollar el modificador	Modificador
RF8	Funcional	Crear reglas de carterización	Se necesita la opción de crear reglas para así determinar la carterización de manera correcta	Únicamente lo puede desarrollar el modificador	Modificador
RF9	Funcional	Consultar reglas de carterización	Se debe tener la opción de consultar cuales reglas se aplican actualmente	Únicamente lo puede desarrollar el modificador	Modificador
RF10	Funcional	Modificar reglas de carterización	Las reglas que fueron creadas también se pueden modificar	Únicamente lo puede desarrollar el modificador	Modificador
RF11	Funcional	Eliminar reglas de carterización	También se pueden eliminar esas reglas que ya no son necesarias	Únicamente lo puede	Modificador

				desarrollar el modificador	
--	--	--	--	----------------------------	--

Fuente: Creación propia

3.1.4 REQUISITOS NO FUNCIONALES

Los siguientes requisitos son clasificados como no funcionales debido a que no son estrictamente necesarios para el funcionamiento de la aplicación web. Sin embargo, son atributos de calidad, que buscan mejorar la experiencia general del usuario una vez entra en contacto con la aplicación.

Tabla 3 Requisitos no funcionales

Código	Clase	Subclase	Descripción
AC1	No funcional	Accesibilidad	Algunos usuarios pueden contar con algún tipo de discapacidad de tipo auditiva o visual. Para el primer caso no hay ningún tipo de afectación ya que la primera versión del sistema no cuenta con medios audiovisuales. Sin embargo, para el segundo caso, si se sugiere la inclusión de herramientas de voz.
AC2	No funcional	Accesibilidad	Se debe contar con un diseño intuitivo que permita al usuario llegar rápidamente a lo que el desea.
AC3	No funcional	Rendimiento	La aplicación web debe mostrar un buen rendimiento basado en la disponibilidad, se recomienda que el servidor este activo 24*7 y que sea lo suficientemente bueno para que soporte el alto tráfico de datos que puede llegar a tener
AC4	No funcional	Seguridad	Se debe diseñar la aplicación web basado en buenas prácticas que priorice el cuidado de la información, evite posibles fugas y este al tanto de ataques cibernéticos
AC5	No funcional	Compatibilidad	La aplicación web debe poder desplegarse correctamente en el hardware (computadores) que maneja la empresa, adicional a ser compatible con las demás herramientas que se manejan.

Fuente: Creación propia

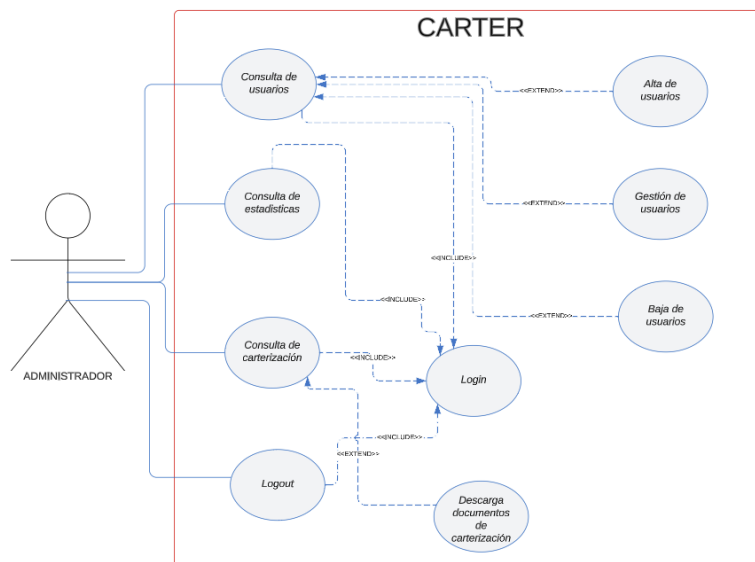
3.1.5 CASOS DE USO

El siguiente paso para el diseño del software es la construcción de los casos de uso, la cual es una técnica que permite recopilar los requisitos del software que se desea construir. En los casos de uso se especifica los escenarios en los cuales el sistema interactúa con el usuario para así conseguir un objetivo, para esto se suele utilizar un lenguaje más cercano al usuario sin utilizar tanta terminología técnica [28].

El tipo de caso de uso que se está utilizando es el de UML, el cual tiene una ventaja al tradicional en informar como son las dependencias entre casos de uso, para esto utiliza dos tipos de asociaciones, incluye y extend [32].

A continuación, se presenta los casos de uso de los diferentes actores de la aplicación, iniciando con administrador:

Ilustración 33 Casos de uso Administrador



Fuente: Creación propia

3.1.5.1 CASOS DE USO DETALLADO PERFIL ADMINISTRADOR

El primer perfil que hay es el administrador, el cual cuenta con una función principal la cual es la administración de los usuarios que tienen acceso a la aplicación, a continuación, se detallan sus casos de uso:

Caso de uso para el login del administrador:

Tabla 4 Caso de uso Login Admin

LOGIN			
Precondiciones	El usuario debe estar registrado dentro de la base de datos de la empresa		
	La base de datos de la aplicación debe estar enlazada con la base de datos de la empresa		
Descripción	Para tener ingreso al aplicativo, el usuario debe digitar los datos de autenticación que tiene para hacer uso de los otros aplicativos de la empresa *Usuario de red y contraseña		
Secuencia	Paso	Acción	
	1	Usuario entra a la dirección web de la aplicación	
	2	Sistema pide los datos de ingreso	
	3	Usuario ingresa los datos solicitados	
	4	Sistema valida si los datos ingresados son correctos	
	5	Usuario ingresa al aplicativo	
Postcondición	El usuario ha ingresado a la aplicación correctamente		
Excepciones	Paso	Acción	
	4	Datos de autenticación invalida	
		E.1	El usuario de red es incorrecto
		E.2	La contraseña es incorrecta
Comentarios	Para validar los datos de autenticación se verifica con la información que se encuentra en la base de datos de la empresa		

Fuente: Creación propia

Caso de uso para el alta de usuarios por parte del administrador:

Tabla 5 Caso de uso alta de usuarios Admin

ALTA DE USUARIOS		
Precondiciones	El administrador debe estar logueado en la aplicación	
Descripción	El administrador esta encargado de permitir el ingreso de nuevos usuarios a la aplicación	
Secuencia	Paso	Acción
	1	El administrador solicita el usuario de red de la persona a la cual le va a otorgar el acceso
	2	Administrador ingresa a la sección "Nuevos usuarios"
	3	Se ingresa el usuario de red al sistema
	4	Sistema consulta en la base de datos si existe el usuario de red y devuelve el resultado
	5	Administrador selecciona el tipo de perfil dentro de la aplicación
	6	Administrador confirma la acción
	7	El sistema agrega el nuevo usuario a la base de datos
Postcondición	El administrador da de alta al usuario para que pueda acceder a la aplicación	
Excepciones	Paso	Acción
	2	Usuario no le aparece la sección "Usuarios activos"
	E.1	Se debe validar que el usuario tenga perfil de administrador
	4	Usuario de red no existe
	E.1	Se valida que este correctamente escrito el usuario de red
	E.2	Validación interna dentro de la empresa para validar si el usuario existe
Comentarios	La validación de datos se realiza con la base de datos de la empresa para garantizar que solo empleados accedan a la aplicación	

Fuente: Creación propia

Caso de uso para la consulta de usuarios por parte del administrador:

Tabla 6 Caso de uso consulta de usuarios Admin

CONSULTA DE USUARIOS		
Precondiciones	El administrador debe estar logueado en la aplicación	
Descripción	El administrador puede consultar los usuarios activos dentro de la aplicación, adicional a ello puede conocer el tipo de perfil que tiene y el ultimo acceso a la aplicación	
Secuencia	Paso	Acción

	1	El administrador ingresa a la sección de "Usuarios activos"	
	2	Sistema devuelve la ventana donde se encuentran almacenados	
Postcondición	El administrador consulta los usuarios activos, tipo de perfil y fecha de ultimo ingreso.		
Excepciones	Paso	Acción	
	1	Usuario no le aparece la sección "Usuarios activos"	
		E.1	Se debe validar que el usuario tenga perfil de administrador
Comentarios	Administrador no tiene restricción para la consulta de usuarios		

Fuente: Creación propia

Caso de uso para la gestión de usuarios por parte del administrador:

Tabla 7 Caso de uso gestión de usuario Admin

GESTIÓN DE USUARIOS			
Precondiciones	El administrador debe estar logueado en la aplicación		
Descripción	El administrador puede cambiar el tipo de perfil de los usuarios activos de la aplicación		
Secuencia	Paso	Acción	
	1	El administrador ingresa a la sección de "Usuarios activos"	
	2	El sistema devuelve la ventana donde se encuentran almacenados	
	3	El administrador modifica el tipo de perfil del usuario seleccionado	
	4	El administrador confirma la selección	
	5	El sistema modifica el perfil del usuario seleccionado dentro de la base de datos	
Postcondición	El usuario ahora ingresa a la aplicación con el nuevo perfil		
Excepciones	Paso	Acción	
	1	Usuario no le aparece la sección "Usuarios activos"	
		E.1	Se debe validar que el usuario tenga perfil de administrador

Comentarios	Se debe modificar el usuario en la base de datos para que ahora pueda acceder según el perfil establecido
--------------------	---

Fuente: Creación propia

Caso de uso para dar de baja a los usuarios por parte del administrador:

Tabla 8 Caso de uso baja de usuario Admin

BAJA DE USUARIOS			
Precondiciones	El administrador debe estar logueado en la aplicación		
Descripción	El administrador puede denegar el ingreso de usuarios a la aplicación eliminando el acceso que tienen		
Secuencia	Paso	Acción	
	1	El administrador ingresa a la sección de "Usuarios activos"	
	2	El sistema devuelve la ventana donde se encuentran almacenados	
	3	El administrador busca el usuario y selecciona la opción de eliminar	
	4	El administrador confirma la selección	
	5	El sistema elimina el usuario dentro de la base de datos	
Postcondición	El usuario ahora tiene el acceso restringido a la aplicación		
Excepciones	Paso	Acción	
	1	Usuario no le aparece la sección "Usuarios activos"	
		E.1	Se debe validar que el usuario tenga perfil de administrador
Comentarios	Se debe eliminar el usuario en la base de datos para que ahora no pueda acceder al aplicativo		

Fuente: Creación propia

Caso de uso para la consulta de la carterización por parte del administrador:

Tabla 9 Caso de uso consulta de carterización Admin

CONSULTA DE CARTERIZACIÓN			
Precondiciones	El administrador debe estar logueado en la aplicación		
Descripción	El administrador puede consultar la carterización actual accediendo a todas las opciones disponibles		
Secuencia	Paso	Acción	
	1	El administrador ingresa a la sección de "Carterización"	

	2	Sistema devuelve la ventana donde se encuentran almacenados
	3	Administrador selecciona la carterización que quiere visualizar navegando en la aplicación
Postcondición	El administrador consulta la carterización que desea observar	
Excepciones	Paso	Acción
	N/A	N/A
	E.1	N/A
Comentarios	Administrador no tienen ninguna restricción para la consulta	

Fuente: Creación propia

Caso de uso para la descarga de la carterización por parte del administrador:

Tabla 10 Caso de uso Descarga documentos de carterización Admin

DESCARGA DOCUMENTOS DE CARTERIZACIÓN		
Precondiciones	El administrador debe estar logueado en la aplicación	
Descripción	El administrador puede descargar la carterización actual de la opción que este consultando	
Secuencia	Paso	Acción
	1	El administrador ingresa a la sección de "Carterización"
	2	Sistema devuelve la ventana donde se encuentran almacenados
	3	Administrador selecciona la carterización que quiere visualizar navegando en la aplicación
	4	Sistema devuelve la ventana donde se visualiza una cantidad limitada de registros de la carterización, adicional se muestra la opción de descarga
	5	Administrador acciona la opción de descargar
	6	Administrador confirma la acción
	7	Sistema genera la descarga del archivo
Postcondición	El administrador tiene acceso total al archivo donde se encuentra la carterización que desea observar	
Excepciones	Paso	Acción
	N/A	N/A
	E.1	N/A
Comentarios	El archivo de descarga es (.TXT)	

Fuente: Creación propia

Caso de uso para la consulta de estadísticas por parte del administrador:

Tabla 11 Caso de uso Consulta de estadísticas Admin

CONSULTA DE ESTADISTICAS			
Precondiciones	El administrador debe estar logueado en la aplicación		
Descripción	El administrador puede consultar las estadísticas referentes a la carterización en sus diferentes segmentos		
Secuencia	Paso	Acción	
	1	El administrador ingresa a la sección de "Estadísticas"	
	2	Sistema devuelve la ventana donde se encuentran almacenados	
Postcondición	El administrador consulta las estadísticas		
Excepciones	Paso	Acción	
	N/A	N/A	
		E.1	N/A
Comentarios	Administrador no tienen ninguna restricción para la visualización de la sección de "Estadísticas"		

Fuente: Creación propia

Caso de uso para el logout por parte del administrador:

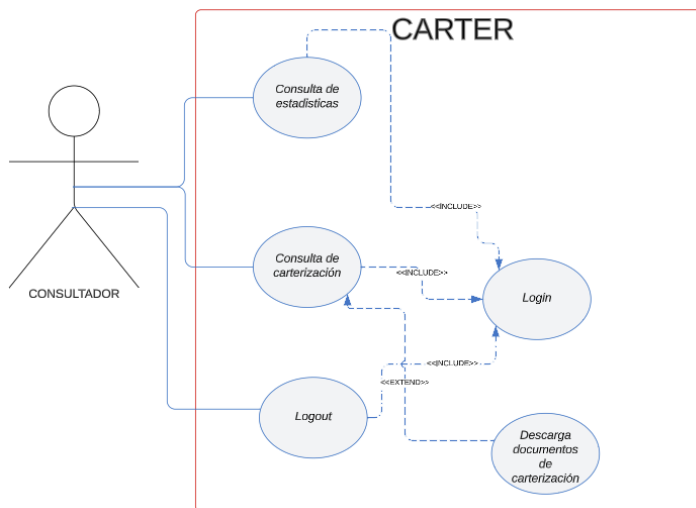
Tabla 12 Caso de uso logout Admin

LOGOUT			
Precondiciones	El administrador debe estar logueado en la aplicación		
Descripción	El administrador desea salir del aplicativo		
Secuencia	Paso	Acción	
	1	Administrador acciona la opción de "Cerrar sesión"	
	2	Sistema dirige al usuario a la sección de login	
Postcondición	El usuario ha salido de la aplicación correctamente		
Excepciones	Paso	Acción	
	N/A	N/A	
		E.1	N/A
Comentarios	Para volver a acceder a las funciones es necesario volver a loguearse		

Fuente: Creación propia

El siguiente perfil corresponde al consultador, el cual tiene la posibilidad de conocer y visualizar como está la carterización actualmente.

Ilustración 34 Casos de uso Consultador



Fuente: Creación propia

3.1.5.2 CASOS DE USO DETALLADO PERFIL CONSULTADOR

A continuación, se describen los casos de uso que pertenecen al consultador:

Caso de uso el login por parte del consultador:

Tabla 13 Caso de uso Login consultador

LOGIN		
Precondiciones	El usuario debe estar registrado dentro de la base de datos de la empresa	
	La base de datos de la aplicación debe estar enlazada con la base de datos de la empresa	
Descripción	Para tener ingreso al aplicativo, el usuario debe digitar los datos de autenticación que tiene para hacer uso de los otros aplicativos de la empresa *Usuario de red y contraseña	
Secuencia	Paso	Acción
	1	Usuario entra a la dirección web de la aplicación
	2	Sistema pide los datos de ingreso

	3	Usuario ingresa los datos solicitados		
	4	Sistema valida si los datos ingresados son correctos		
	5	Usuario ingresa al aplicativo		
Postcondición	El usuario ha ingresado a la aplicación correctamente			
Excepciones	Paso	Acción		
	4	Datos de autenticación invalida		
		E.1	El usuario de red es incorrecto	
		E.2	La contraseña es incorrecta	
Comentarios	Para validar los datos de autenticación se verifica con la información que se encuentra en la base de datos de la empresa			

Fuente: Creación propia

Caso de uso la consulta de carterización parte del consultador:

Tabla 14 Caso de uso consulta de carterización consultador

CONSULTA DE CARTERIZACIÓN			
Precondiciones	El consultador debe estar logueado en la aplicación		
Descripción	El consultador puede revisar la carterización actual accediendo a todas las opciones disponibles		
Secuencia	Paso	Acción	
	1	El consultador ingresa a la sección de "Carterización"	
	2	Sistema devuelve la ventana donde se encuentran almacenados	
	3	Consultador selecciona la carterización que quiere visualizar navegando en la aplicación	
Postcondición	El consultador visualiza el archivo de carterización según el aplicativo escogido		
Excepciones	Paso	Acción	
	N/A	N/A	
		E.1	N/A
Comentarios	Consultador no tienen ninguna restricción para la visualización de los archivos		

Fuente: Creación propia

Caso de uso la descarga de la carterización por parte del consultador:

Tabla 15 Caso de uso descarga documentos de carterización consultador

DESCARGA DOCUMENTOS DE CARTERIZACIÓN			
Precondiciones	El consultador debe estar logueado en la aplicación		
Descripción	El consultador puede descargar la carterización actual de la opción que este revisando		
Secuencia	Paso	Acción	
	1	El consultador ingresa a la sección de "Carterización"	
	2	Sistema devuelve la ventana donde se encuentran almacenados	
	3	Consultador selecciona la carterización que quiere visualizar navegando en la aplicación	
	4	Sistema devuelve la ventana donde se visualiza una cantidad limitada de registros de la carterización, adicional se muestra la opción de descarga	
	5	Consultador acciona la opción de descargar	
	6	Consultador confirma la acción	
	7	Sistema genera la descarga del archivo	
Postcondición	El consultador tiene acceso total al archivo donde se encuentra la carterización que desea observar		
Excepciones	Paso	Acción	
	N/A	N/A	
		E.1	N/A
Comentarios	El archivo de descarga es (.TXT)		

Fuente: Creación propia

Caso de uso la consulta de estadísticas por parte del consultador:

Tabla 16 Caso de uso Consulta de estadísticas Consultador

CONSULTA DE ESTADISTICAS			
Precondiciones	El consultador debe estar logueado en la aplicación		
Descripción	El consultador puede revisar las estadísticas referentes a la carterización en sus diferentes segmentos		
Secuencia	Paso	Acción	
	1	El consultador ingresa a la sección de "Estadísticas"	

	2	Sistema devuelve la ventana donde se encuentran almacenados	
Postcondición	El consultador visualiza las estadísticas		
Excepciones	Paso	Acción	
	N/A	N/A	
		E.1	N/A
Comentarios	Consultador no tienen ninguna restricción para la visualización de la sección de "Estadísticas"		

Fuente: Creación propia

Caso de uso el logout por parte del consultador:

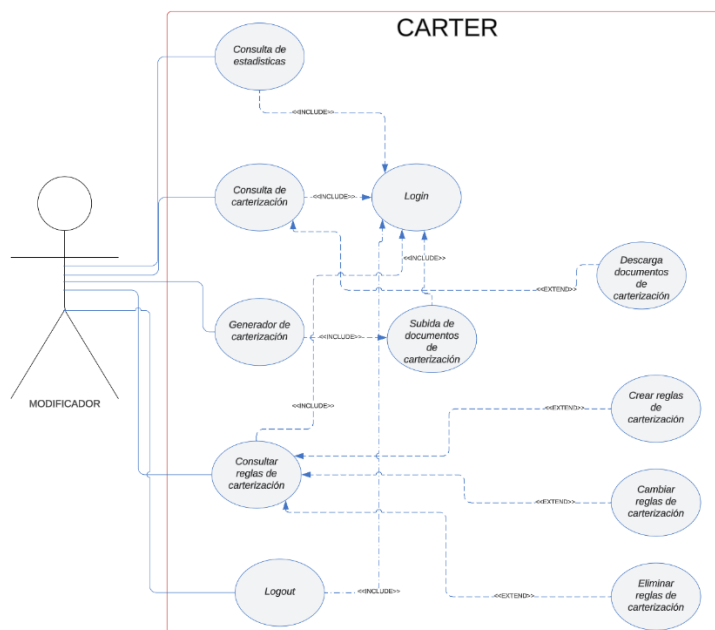
Tabla 17 Caso de uso Logout consultador

LOGOUT			
Precondiciones	El consultador debe estar logueado en la aplicación		
Descripción	El consultador desea salir del aplicativo		
Secuencia	Paso	Acción	
	1	Consultador acciona la opción de "Cerrar sesión"	
	2	Sistema dirige al usuario a la sección de login	
Postcondición	El usuario ha salido de la aplicación correctamente		
Excepciones	Paso	Acción	
	N/A	N/A	
		E.1	N/A
Comentarios	Para volver a acceder a las funciones es necesario volver a loguearse		

Fuente: Creación propia

Finalmente se presenta el diagrama de los casos de uso pertenecientes al tercer perfil de la aplicación, el cual es el modificador.

Ilustración 35 Casos de uso modificador



Fuente: Creación propia

3.1.5.3 CASOS DE USO DETALLADO PERFIL MODIFICADOR

A continuación, se describen los casos de uso que pertenecen al modificador, este perfil es el que más permisos contiene debido a que puede consultar y modificar la carterización:

Caso de uso el login por parte del modificador:

Tabla 18 Caso de uso Login Modificador

LOGIN		
Precondiciones	El usuario debe estar registrado dentro de la base de datos de la empresa	
	La base de datos de la aplicación debe estar enlazada con la base de datos de la empresa	
Descripción	Para tener ingreso al aplicativo, el usuario debe digitar los datos de autenticación que tiene para hacer uso de los otros aplicativos de la empresa *Usuario de red y contraseña	
Secuencia	Paso	Acción
	1	Usuario entra a la dirección web de la aplicación
	2	Sistema pide los datos de ingreso
	3	Usuario ingresa los datos solicitados
	4	Sistema valida si los datos ingresados son correctos

	5	Usuario ingresa al aplicativo		
Postcondición	El usuario ha ingresado a la aplicación correctamente			
Excepciones	Paso	Acción		
	4	Datos de autenticación invalida		
		E.1	El usuario de red es incorrecto	
		E.2	La contraseña es incorrecta	
Comentarios	Para validar los datos de autenticación se verifica con la información que se encuentra en la base de datos de la empresa			

Fuente: Creación propia

Caso de uso la consulta de carterización por parte del modificador:

Tabla 19 Caso de uso consulta de carterización Modificador

CONSULTA DE CARTERIZACIÓN			
Precondiciones	El modificador debe estar logueado en la aplicación		
Descripción	El modificador puede revisar la carterización actual accediendo a todas las opciones disponibles		
Secuencia	Paso	Acción	
	1	El modificador ingresa a la sección de "Carterización"	
	2	Sistema devuelve la ventana donde se encuentran almacenados	
	3	Modificador selecciona la carterización que quiere visualizar navegando en la aplicación	
Postcondición	El modificador visualiza el archivo de carterización según el aplicativo escogido		
Excepciones	Paso	Acción	
	N/A	N/A	
		E.1	N/A
Comentarios	Modificador no tienen ninguna restricción para la visualización de los archivos		

Fuente: Creación propia

Caso de uso la descarga de la carterización por parte del modificador:

Tabla 20 Caso de uso descarga documentos de carterización Modificador

DESCARGA DOCUMENTOS DE CARTERIZACIÓN			
Precondiciones	El modificador debe estar logueado en la aplicación		
Descripción	El modificador puede descargar la carterización actual de la opción que este revisando		
Secuencia	Paso	Acción	
	1	El modificador ingresa a la sección de "Carterización"	
	2	Sistema devuelve la ventana donde se encuentran almacenados	
	3	Modificador selecciona la carterización que quiere visualizar navegando en la aplicación	
	4	Sistema devuelve la ventana donde se visualiza una cantidad limitada de registros de la carterización, adicional se muestra la opción de descarga	
	5	Modificador acciona la opción de descargar	
	6	Modificador confirma la acción	
	7	Sistema genera la descarga del archivo	
Postcondición	El modificador tiene acceso total al archivo donde se encuentra la carterización que desea observar		
Excepciones	Paso	Acción	
	N/A	N/A	
		E.1	N/A
Comentarios	El archivo de descarga es (.TXT)		

Fuente: Creación propia

Caso de uso la consulta de estadísticas por parte del modificador:

Tabla 21 Caso de uso consulta de estadística Modificador

CONSULTA DE ESTADISTICAS			
Precondiciones	El modificador debe estar logueado en la aplicación		
Descripción	El modificador puede revisar las estadísticas referentes a la carterización en sus diferentes segmentos		
Secuencia	Paso	Acción	
	1	El modificador ingresa a la sección de "Estadísticas"	
	2	Sistema devuelve la ventana donde se encuentran almacenados	
Postcondición	El modificador visualiza las estadísticas		
Excepciones	Paso	Acción	
	N/A	N/A	
		E.1	N/A

Comentarios	Modificador no tienen ninguna restricción para la visualización de la sección de "Estadísticas"
--------------------	---

Fuente: Creación propia

Caso de uso la subida de documentos a la plataforma para realizar la carterización por parte del modificador:

Tabla 22 Caso de uso subida documentos de carterización Modificador

SUBIDA DE DOCUMENTOS DE CARTERIZACIÓN				
Precondiciones	El modificador debe estar logueado en la aplicación			
Descripción	El modificador puede subir los insumos, en este caso los archivos, para que la aplicación pueda funcionar			
Secuencia	Paso		Acción	
	1		El modificador ingresa a la sección de "Carterización"	
	2		Sistema devuelve la ventana donde se encuentran almacenados	
	3		Modificador navega por la aplicación hasta llegar al segmento y herramienta deseada	
	4		Modificador selecciona la opción "archivos"	
	5		Sistema solicita ingreso del archivo	
	6		Sistema confirma la carga de los archivos	
Postcondición	El sistema ahora cuenta con los insumos para generar la carterización			
Excepciones	Paso		Acción	
	6		El sistema arroja error en la carga de archivo	
			E.1	Error en el formato del archivo de carga
			E.2	Error en el tipo de archivo
Comentarios	Modificador es el único tipo de usuario que tiene acceso a esta sección El tipo de archivo es .TXT Es necesario dos archivos, uno que contiene a los clientes y el otro a los analistas			

Fuente: Creación propia

Caso de uso para conocer las reglas actuales de la carterización por parte del modificador:

Tabla 23 Caso de uso Consultar reglas de carterización Modificador

CONSULTAR REGLAS DE CARTERIZACIÓN			
Precondiciones	El modificador debe estar logueado en la aplicación		
Descripción	El modificador puede conocer cuáles son las reglas que están configuradas actualmente para la carterización		
Secuencia	Paso	Acción	
	1	El modificador ingresa a la sección de "Carterización"	
	2	Sistema devuelve la ventana donde se encuentran almacenados	
	3	Modificador navega por la aplicación hasta llegar al segmento y herramienta deseada	
	4	Modificador selecciona la opción "Reglas"	
	5	Sistema muestra una tabla con las reglas creadas, nombre, fecha de modificación, usuario que realizo la última modificación, descripción de la regla, opción de modificar, opción de eliminar	
Postcondición	El modificador ahora conoce las reglas actuales		
Excepciones	Paso	Acción	
	N/A	N/A	
		E.1	N/A
Comentarios	Modificador es el único tipo de usuario que tiene acceso a esta sección		

Caso de uso la creación de las reglas para la carterización por parte del modificador:

Tabla 24 Caso de uso crear reglas de carterización Modificador

CREAR REGLAS DE CARTERIZACIÓN			
Precondiciones	El modificador debe estar logueado en la aplicación		
Descripción	El modificador puede crear nuevas reglas para la carterización		
Secuencia	Paso	Acción	
	1	El modificador ingresa a la sección de "Carterización"	
	2	Sistema devuelve la ventana donde se encuentran almacenados	

	3	Modificador navega por la aplicación hasta llegar al segmento y herramienta deseada		
	4	Modificador selecciona la opción "Reglas"		
	5	Sistema muestra una tabla con las reglas creadas adicional muestra una la opción de crear regla		
	6	Modificador selecciona la opción " Crear regla"		
	7	Sistema arroja ventana donde el modificador puede especificar el tipo de regla		
	8	Sistema confirma la creación		
	9	Sistema actualiza la tabla de la opción "Reglas"		
Postcondición		La nueva regla ha sido creada y se aplica los archivos a la hora de carterizar		
Excepciones	Paso	Acción		
	8	La regla no pudo ser creada		
		E.1	Se debe verificar que la regla este creada según los parámetros de la aplicación lo define	
		E.2	La regla puede estar duplicada	
Comentarios		Modificador es el único tipo de usuario que tiene acceso a esta sección		

Fuente: Creación propia

Caso de uso cambiar las reglas de carterización actuales por parte del modificador:

Tabla 25 Caso de uso cambiar reglas de carterización Modificador

CAMBIAR REGLAS DE CARTERIZACIÓN			
Precondiciones	El modificador debe estar logueado en la aplicación		
Descripción	El modificador puede modificar las reglas que desee según sea necesario		
Secuencia	Paso	Acción	

	1	El modificador ingresa a la sección de "Carterización"		
	2	Sistema devuelve la ventana donde se encuentran almacenados		
	3	Modificador navega por la aplicación hasta llegar al segmento y herramienta deseada		
	4	Modificador selecciona la opción "Reglas"		
	5	Sistema muestra una tabla con las reglas creadas, fecha de modificación, usuario que realizo la última modificación, opción de modificar, opción de eliminar		
	6	Modificador selecciona la regla que desea modificar y obtura la opción "modificar"		
	7	Sistema arroja ventana donde el modificador puede especificar el tipo de regla		
	8	Sistema confirma la modificación		
	9	Sistema actualiza la tabla de la opción "Reglas"		
Postcondición		El modificador modifiko las reglas previamente creadas		
Excepciones	Paso	Acción		
	8	La regla no pudo ser creada		
		E.1	Se debe verificar que la regla este creada según los parámetros de la aplicación lo define	
		E.2	La regla puede estar duplicada	
Comentarios		Modificador es el único tipo de usuario que tiene acceso a esta sección		

Fuente: Creación propia

Caso de uso eliminar las reglas de carterización por parte del modificador:

Tabla 26 Caso de uso eliminar reglas de carterización Modificador

ELIMINAR REGLAS DE CARTERIZACIÓN			
Precondiciones	El modificador debe estar logueado en la aplicación		
Descripción	El modificador puede modificar las reglas que desee según sea necesario		
Secuencia	Paso	Acción	
	1	El modificador ingresa a la sección de "Carterización"	
	2	Sistema devuelve la ventana donde se encuentran almacenados	
	3	Modificador navega por la aplicación hasta llegar al segmento y herramienta deseada	
	4	Modificador selecciona la opción "Reglas"	

	5	Sistema muestra una tabla con las reglas creadas, fecha de modificación, usuario que realizo la última modificación, opción de modificar, opción de eliminar	
	6	Modificador selecciona la regla que desea modificar y obtura la opción "Eliminar"	
	7	Sistema confirma la acción	
	8	Sistema actualiza la tabla de la opción "Reglas"	
Postcondición	El modificador elimino las reglas correctamente		
Excepciones	Paso	Acción	
	N/A	N/A	
		E.1	N/A
Comentarios	Modificador es el único tipo de usuario que tiene acceso a esta sección		

Fuente: Creación propia

Caso de uso para generar la carterización por parte del modificador:

Tabla 27 Caso de uso generador de carterización Modificador

GENERADOR DE CARTERIZACIÓN		
Precondiciones	El modificador debe estar logueado en la aplicación	
	Los archivos deben estar subidos correctamente	
	Las reglas deben estar establecidas	
Descripción	El modificador puede generar las carterización teniendo como base los insumos proporcionados a través de los archivos y las reglas establecidas	
Secuencia	Paso	Acción
	1	El modificador ingresa a la sección de "Carterización"
	2	Sistema devuelve la ventana donde se encuentran almacenados
	3	Modificador navega por la aplicación hasta llegar al segmento y herramienta deseada
	4	Sistema valida que estén cargados los archivos y las reglas estén establecidas
	5	Modificador selecciona la opción "Generar"
	6	Sistema confirma la acción

	7	Sistema actualiza las tablas para la consulta de la carterización		
Postcondición	El modificador genero la carterización			
Excepciones	Paso	Acción		
	4	Sistema no encuentra los insumos cargados		
		E.1	Se debe verificar si ambos insumos están cargados	
		E.2	Se debe verificar si las reglas están establecidas	
Comentarios	Modificador es el único tipo de usuario que tiene acceso a esta sección			

Fuente: Creación propia

Caso de uso el logout por parte del modificador:

Tabla 28 Caso de uso Logout Modificador

LOGOUT			
Precondiciones	El modificador debe estar logueado en la aplicación		
Descripción	El modificador desea salir del aplicativo		
Secuencia	Paso	Acción	
	1	Modificador acciona la opción de "Cerrar sesión"	
	2	Sistema dirige al usuario a la sección de login	
Postcondición	El usuario ha salido de la aplicación correctamente		
Excepciones	Paso	Acción	
	N/A	N/A	
		E.1	N/A
Comentarios	Para volver a acceder a las funciones es necesario volver a loguearse		

Fuente: Creación propia

3.2 ETAPA 2 DISEÑO

Esta etapa permitió realizar un diseño general de la aplicación web basado en las exigencias que fueron definidas en la etapa anterior. A partir de ahora se entra en consideración aspectos más técnicos como el diseño de la base de datos, el modelo de navegación o el

diagrama de despliegue, esta información es especialmente útil a la hora del desarrollo de la aplicación.

Como se comentó anteriormente, para la gestión del proyecto hizo uso de SCRUM, dado esto es importante tener en cuenta diferentes tipos de terminología que son propias de la metodología como el product Backlog y el sprint Backlog.

3.2.1 PRODUCT BACKLOG

El product backlog se caracteriza por ser un listado donde se evidencian los requisitos que se van a cumplir a la hora del diseño del software. En el backlog se jerarquiza los requisitos dependiendo de su importancia, la cual se calcula al multiplicar la urgencia por el impacto.

Este backlog no es fijo durante el proyecto, sino que puede actualizarse según se presente la necesidad, gracias a las reuniones que se hagan.

El formato que se maneja es el siguiente:

Ilustración 36 Formato Product Backlog

PRODUCT BACKLOG					
ID	ACTIVIDAD	URGENCIA	IMPACTO	IMPORTANCIA	TIEMPO ESTIMADO

Fuente: Creación propia

Donde:

- ✓ ID: Identificador único de la historia
- ✓ Historia de usuarios: Nombre de la historia
- ✓ Urgencia: Se maneja en el rango de 1 como menor urgencia hasta 5 como mayor urgencia

- ✓ Impacto: Se maneja en el rango de 1 como menor impacto hasta 5 como mayor impacto
- ✓ Importancia: Se obtiene con el producto de la urgencia por el impacto
- ✓ Tiempo estimado: Estimado de duración de la tarea en días.

El siguiente Product Backlog que se presenta es el que se utilizó para el desarrollo del actual trabajo, en donde se cubre hasta la etapa de diseño con base a la metodología de cascada.

Tabla 29 Product Backlog

PRODUCT BACKLOG					
ID	ACTIVIDAD	URGENCIA	IMPACTO	IMPORTANCIA	TIEMPO ESTIMADO
1	Requerimientos	5	5	25	15
2	Requisitos funcionales y no funcionales	4	5	20	10
3	Modelo de navegación (Arquitectura de información)	4	3	12	5
4	UI/Mockups	5	5	25	18
5	Consideraciones de accesibilidad y usabilidad	2	3	6	5
6	Modelo de base de datos	5	4	20	12
7	Modelo de componentes y despliegue	3	3	9	5

Fuente: Creación propia

3.2.2 SPRINT BACKLOG

Otro de los productos del uso de SCRUM es el sprint backlog, los cuales son una serie de historias de usuario que se van a solucionar durante un tiempo determinado, estas historias son las que se encuentran en el Product Backlog. A su vez, este sprint es dividido en tareas más

simples que facilitan su desarrollo y cumplimiento. Según las buenas prácticas de scrum, su duración no debe ser mayor a las 4 semanas, si se excede este tiempo, es mejor programar la historia para el siguiente sprint.

A continuación, se asignan las historias que se encuentran en el product backlog a un determinado sprint.

Tabla 30 Sprint Backlog

SPRINT BACKLOG		
ID	HISTORIA DE USUARIO	SPRINT PROGRAMADO
1	Requerimientos	1
2	Requisitos funcionales y no funcionales	1
4	UI/Mockups	2
6	Modelo de base de datos	2
3	Modelo de navegación (Arquitectura de información)	3
5	Consideraciones de accesibilidad y usabilidad	3
7	Modelo de componentes y despliegue	3

Fuente: Creación propia

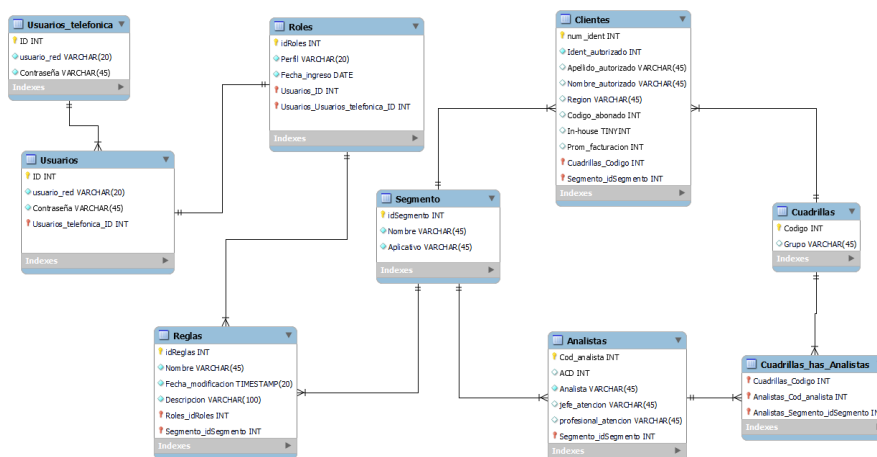
3.2.3 DISEÑO DE BASE DE DATOS

Teniendo en cuenta que se realizó el diseño de un aplicativo web, se convierte en indispensable no hacer referencia a la base de datos que se va a utilizar. Como se mencionó anteriormente, en este apartado de la aplicación se almacena toda la información relevante de la misma, es decir, acá se guardan los datos de la aplicación, por ejemplo, cuáles son los usuarios

que pueden entrar, cuáles son los clientes que hay, cuáles son las reglas, en fin, información que al final es el objetivo principal de la aplicación.

A partir de ello se realizó la elaboración de un modelo de la base de datos, este es el modelo relacional, el cual se caracteriza por ser un modelo que muestra la relación entre entidades de manera más correcta, ya que incluye las llaves primarias de cada entidad y las llaves foráneas que permite relacionar una entidad con la otra. El diagrama muestra como la base de datos que se diseña es de tipo relacional. Este diagrama se construyó gracias a la ayuda de MYSQL Workbench.

Ilustración 37 Modelo relacional



Fuente: Creación propia

Otra de las opciones que existe en MYSQL Workbench es la posibilidad de generar un diccionario de la base de datos, lo cual es una lista que muestra de manera organizada los nombres de las tablas y los nombres, características y definiciones de cada uno de los atributos de dichas tablas. Tener acceso a esta información tiene un papel crucial ya que ayuda a entender de mejor manera cual es la función de determinada tabla y cuál es su contenido.

Las tablas a continuación presentan un formato que incluye el nombre de la columna, el tipo de dato y las características. Las más importantes son: PK: Primary Key, NN: Not Null, UQ: Unique y AI: Auto Increment

A continuación, se presenta el diccionario de la tabla analistas:

Ilustración 38 BD Analistas

Analistas										
Column name	Data Type	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
Cod_analista	INT	✓	✓							
ACD	INT									
Analista	VARCHAR(45)		✓							
jefe_atencion	VARCHAR(45)									
profesional_atencion	VARCHAR(45)									
Segmento_idSegmento	INT	✓	✓							

FUENTE: Creación propia

Se presenta el diccionario de la tabla clientes:

Ilustración 39 BD Clientes

Clientes										
Column name	Data Type	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
num_ident	INT	✓	✓							
Ident_authorized	INT		✓							
Apellido_authorized	VARCHAR(45)									
Nombre_authorized	VARCHAR(45)									
Region	VARCHAR(45)									
Codigo_abonado	INT									
In-house	TINYINT									
Prom_facturacion	INT									
Cuadrillas_Codigo	INT	✓	✓							
Segmento_idSegmento	INT	✓	✓							

FUENTE: Creación propia

Se presenta el diccionario de la tabla cuadrillas:

Ilustración 40 BD Cuadrillas

Cuadrillas										
Column name	Data Type	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
Codigo	INT	✓	✓							
Grupo	VARCHAR(45)									

FUENTE: Creación propia

Se presenta el diccionario de la tabla de rompimiento cuadrillas-analistas:

Ilustración 41 BD tabla rompimiento cuadrillas-analistas

Cuadrillas_has_Analistas											
Column name	Data Type	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment	
Cuadrillas_Codigo	INT	✓	✓								
Analistas_Cod_analista	INT	✓	✓								
Analistas_Segmento_idSegmento	INT	✓	✓								

FUENTE: Creación propia

Se presenta el diccionario de la tabla reglas:

Ilustración 42 BD Reglas

Reglas											
Column name	Data Type	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment	
idReglas	INT	✓	✓					✓			
Nombre	VARCHAR(45)		✓								
Fecha_modificacion	TIMESTAMP(20)		✓								
Descripcion	VARCHAR(100)		✓								
Roles_idRoles	INT	✓	✓								
Segmento_idSegmento	INT	✓	✓								

FUENTE: Creación propia

Se presenta el diccionario de la tabla roles:

Ilustración 43 BD Roles

Roles											
Column name	Data Type	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment	
idRoles	INT	✓	✓					✓			
Perfil	VARCHAR(20)		✓								
Fecha_ingreso	DATE		✓								
Usuarios_ID	INT	✓	✓								
Usuarios_Usuarios_telefonica_ID	INT	✓	✓								

FUENTE: Creación propia

Se presenta el diccionario de la tabla segmento:

Ilustración 44 BD Segmento

Segmento											
Column name	Data Type	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment	
idSegmento	INT	✓	✓								
Nombre	VARCHAR(45)		✓								
Aplicativo	VARCHAR(45)		✓								

FUENTE: Creación propia

Se presenta el diccionario de la tabla usuarios:

Ilustración 45 BD usuarios

Usuarios											
Column name	Data Type	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment	
ID	INT	✓	✓					✓			
usuario_red	VARCHAR(20)		✓								
Contraseña	VARCHAR(45)		✓								
Usuarios_telefonica_ID	INT	✓	✓								

FUENTE: Creación propia

Se presenta el diccionario de la tabla usuarios telefónica:

Ilustración 46 BD usuarios Telefónica

Usuarios_telefonica										
Column name	Data Type	PK	NN	UQ	BIN	UN	ZF	AI	Default	Comment
ID	INT	✓	✓							
usuario_red	VARCHAR(20)		✓							
Contraseña	VARCHAR(45)		✓							

FUENTE: Creación propia

Para la puesta en marcha de la aplicación se recomienda utilizar gestores de bases de datos como Oracle, o SQL Server, debido a que están diseñadas para almacenar y gestionar grandes cantidades de datos, como sería el caso para la aplicación propuesta, sin embargo, la principal dificultad que tiene es el hecho de que son aplicaciones de pago.

3.2.4 VISTAS CONCEPTUALES DEL APLICATIVO

Para tener una mayor claridad de cómo se puede visualizar el aplicativo web, se hace un modelado de las vistas, para esto se hizo uso de los mockups que dan un concepto realista de cómo se vería la aplicación y que, aparte de mostrarle al cliente, también se convierte en una ayuda para el desarrollador para que tenga una idea mucho más clara de la aplicación para su posterior desarrollo.

Inicialmente, se debe informar que hay 3 perfiles dentro de la aplicación, a partir de eso todos los perfiles tienen el mismo modelo para el login de la aplicación.

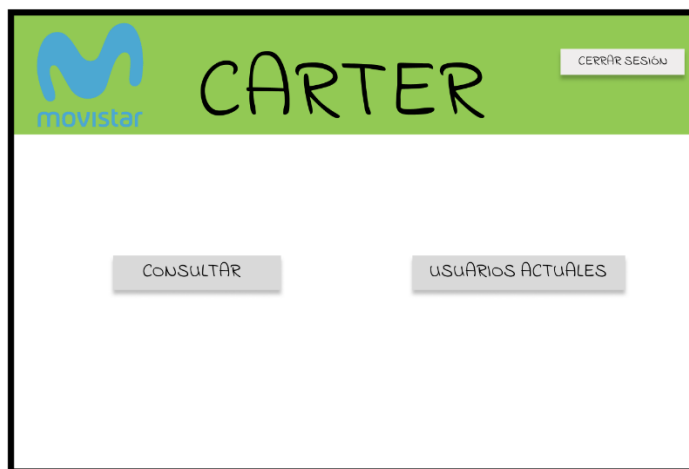
Ilustración 47 Login de la aplicación

Fuente: Creación propia

3.2.4.1 PERFIL UNO: ADMINISTRADOR

Para el caso del administrador, está el siguiente menú principal, en el cual se puede consultar la carterización o se pueden ver cuales usuarios tienen acceso a la aplicación y su respectivo perfil.

Ilustración 48 Página principal del administrador



Fuente: Creación propia

Al ingresar a la sección de “usuarios actuales” se encuentra la siguiente tabla, en la cual se observa el usuario, perfil, fecha de ultimo ingreso, y las opciones de modificar y eliminar en cada usuario.

Ilustración 49 Usuarios actuales

USUARIO	PERFIL	FECHA ULTIMO INGRESO	MODIFICAR	ELIMINAR
AJSOTELOMO	ADMINISTRADOR	04/08/2023 12:53 PM	MODIFICAR	ELIMINAR
JJMARTINEZQU	CONSULTOR	02/07/2023 2:21 PM	MODIFICAR	ELIMINAR

Fuente: Creación propia

Al ingresar en la opción “Modificar” se despliega el siguiente cuadro el cual permite cambiar el perfil del usuario seleccionado y posteriormente solicita confirmación en un cuadro de tipo pop-up.

Ilustración 50 Modificar usuario



The screenshot shows a web interface with a green header. On the left is the Movistar logo. In the center, the word 'CARTER' is written in a large, black, hand-drawn font. On the right is a button labeled 'CERRAR SESIÓN'. Below the header, there are three main sections: 'Modificar usuario' on the left, 'usuario DE RED' in the center, and 'REGRESAR' on the right. Under 'usuario DE RED' is a button labeled 'AJSOTELOMO'. Below that is a section labeled 'PERFIL' with a list of roles: 'ADMINISTRADOR', 'CONSULTADOR', and 'MODIFICADOR'. At the bottom center is a button labeled 'ACEPTAR'.

Fuente: Creación propia

Al darle aceptar, parecer el siguiente cuadro de confirmación:

Ilustración 51 Confirmar modificar usuario



The screenshot shows a confirmation dialog box with a white background and a black border. At the top center is the Movistar logo. Below it, the text reads: '¿DESEA MODIFICAR AL usuario AJSOTELOMO AL PERFIL DE ____?'. At the bottom, there are two buttons: 'CONFIRMAR' and 'CANCELAR'.

Fuente: Creación propia

Para eliminar un usuario, se debe regresar al menú anterior y seleccionar el usuario.

Al seleccionar en la opción “eliminar” se pide una confirmación de la acción.

Ilustración 52 Eliminar usuario



Fuente: Creación propia}

Posteriormente, se arroja un mensaje confirmando la acción.

Ilustración 53 Confirmación usuario eliminado



Fuente: Creación propia

Otra de las opciones con las que cuenta el administrador es la opción de “agregar usuario” el cual se encuentra en la pestaña de consulta de usuarios, al usar esta opción se despliega un formulario donde se debe ingresar el usuario de red y el perfil dentro de la aplicación, posterior a esto se debe confirmar.

Ilustración 54 Agregar usuario

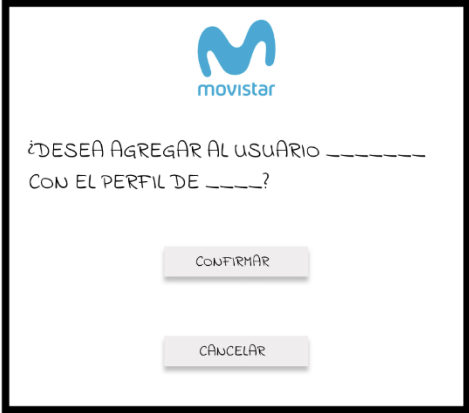


The screenshot shows the 'CARTER' application interface. At the top, there is a green header with the Movistar logo on the left, the word 'CARTER' in large, stylized letters in the center, and a 'CERRAR SESIÓN' button on the right. Below the header, there is a '+ AGREGAR USUARIO' button on the left. To the right of this button, there is a 'USUARIO DE RED' label above a text input field, and a 'REGRESAR' button. Below the 'USUARIO DE RED' field, there is a 'PERFIL' label above a dropdown menu. The dropdown menu is open, showing three options: 'ADMINISTRADOR', 'CONSULTADOR', and 'MODIFICADOR'. At the bottom of the form, there is an 'ACEPTAR' button.

Fuente: Creación propia

Se genera un mensaje de confirmación.

Ilustración 55 Confirmar agregar usuario



The screenshot shows a confirmation dialog box. At the top, there is the Movistar logo. Below the logo, the text reads: '¿DESEA AGREGAR AL USUARIO _____ CON EL PERFIL DE _____?'. At the bottom of the dialog, there are two buttons: 'CONFIRMAR' and 'CANCELAR'.

Fuente: Creación propia

Al regresar a la página principal, se observa la opción de “Consultar” al obturarlo se despliegan otras dos opciones, “Carterización” y “Estadísticas”.

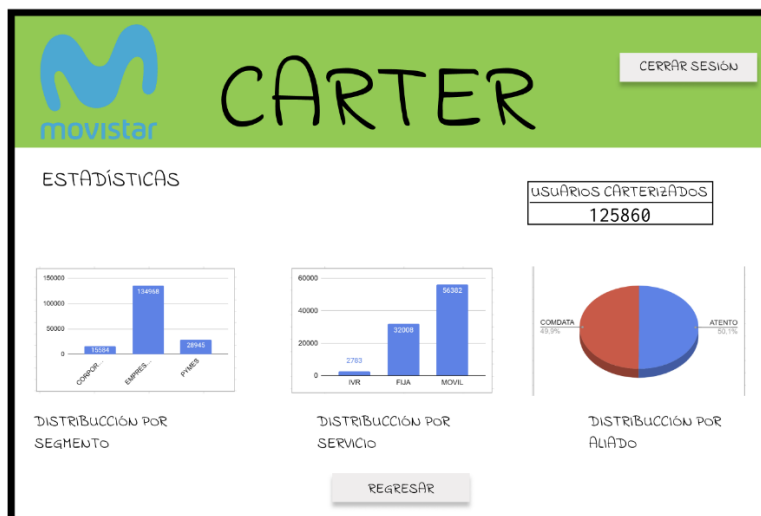
Ilustración 56 Opción consultar Admin



Fuente: Creación propia

La opción de “Estadísticas” da una ayuda visual de cuantos usuarios hay carterizados, como se reparten por segmento o incluso como están repartidos los analistas según el aliado al cual pertenece.

Ilustración 57 Estadística



Fuente: Creación propia

La otra opción, la de “caracterización” permite conocer cuales como se va a atender a los clientes, para ello se divide en segmentos, servicios y aplicativos, cuando un usuario quiera conocer esto, debe navegar por la aplicación según lo que quiera consultar.

Ilustración 58 Segmento a consultar

The screenshot shows the 'CARTER' application interface. At the top, there is a green header with the Movistar logo on the left and the word 'CARTER' in large, handwritten-style letters in the center. A 'CERRAR SESIÓN' button is located in the top right corner. Below the header, the text 'SELECCIONE EL SEGMENTO QUE DESEA CONSULTAR' is displayed. Underneath this text, there are three buttons labeled 'CORPORACIONES', 'EMPRESAS', and 'PYMES'. At the bottom center, there is a 'REGRESAR' button.

Fuente: Creación propia

Posteriormente se debe seleccionar el servicio.

Ilustración 59 Servicio a consultar

The screenshot shows the 'CARTER' application interface after selecting a segment. The header is identical to the previous screen, with the Movistar logo, 'CARTER' text, and 'CERRAR SESIÓN' button. Below the header, the text '[SEGMENTO]' is displayed in a box. The main text 'SELECCIONE EL SERVICIO QUE DESEA CONSULTAR' is shown. Underneath, there are three buttons labeled 'IVR', 'MOVIL', and 'FIJA'. At the bottom center, there is a 'REGRESAR' button.

Fuente: Creación propia

Para la sección de aplicativos, estos van a variar dependiendo del servicio.

Para el servicio de móvil es el siguiente:

Ilustración 60 Aplicativo móvil



Fuente: Creación propia

Y para el servicio de fija se encuentra este:

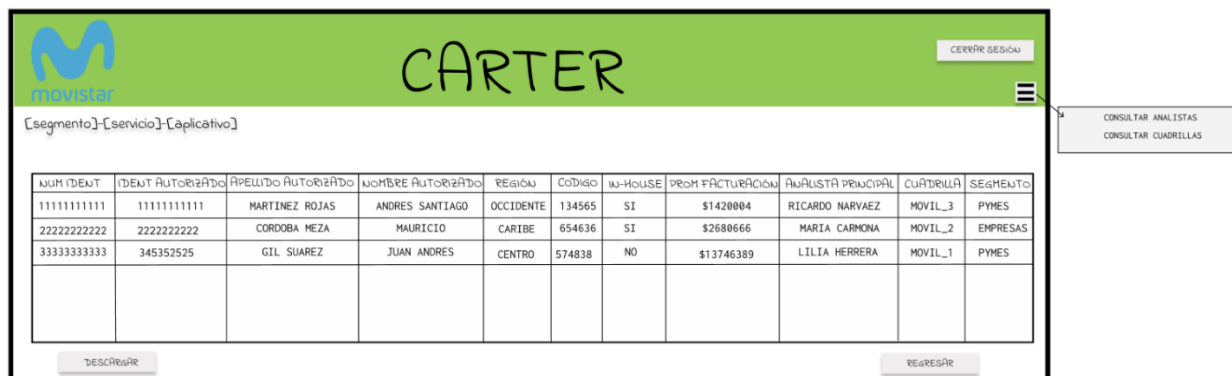
Ilustración 61 Aplicativo fija



Fuente: Creación propia

Así finalmente aparece la sección donde se encuentra la carterización y que cuenta con los siguientes atributos: Numero identificación, Identificación autorizado, apellido autorizado, nombre autorizado, región, código, in-house, promedio de facturación, cuadrilla y segmento.

Ilustración 62 Carterización actual



The screenshot shows a web application interface for 'CARTER'. At the top, there is a green header with the Movistar logo and the word 'CARTER' in large, handwritten-style letters. Below the header, there is a search bar with the placeholder text '[segmento]-[servicio]-[aplicativo]'. To the right of the search bar, there are two buttons: 'CERRAR SESIÓN' and a menu icon. Below the search bar, there is a table with the following columns: NUM IDENT, IDENT AUTORIZADO, APELLIDO AUTORIZADO, NOMBRE AUTORIZADO, REGIÓN, CODIGO, IN-HOUSE, PROM FACTURACIÓN, AJUSTE PRINCIPAL, CUADRILLA, and SEGMENTO. The table contains three rows of data. At the bottom of the table, there are two buttons: 'DESCARGAR' and 'REGRESAR'. To the right of the table, there is a sidebar with two buttons: 'CONSULTAR ANALISTAS' and 'CONSULTAR CUADRELLAS'.

NUM IDENT	IDENT AUTORIZADO	APELLIDO AUTORIZADO	NOMBRE AUTORIZADO	REGIÓN	CODIGO	IN-HOUSE	PROM FACTURACIÓN	AJUSTE PRINCIPAL	CUADRILLA	SEGMENTO
11111111111	11111111111	MARTINEZ ROJAS	ANDRES SANTIAGO	OCCIDENTE	134565	SI	\$1420004	RICARDO NARVAEZ	MOVIL_3	PYMES
22222222222	22222222222	CORDOBA MEZA	MAURICIO	CARIBE	654636	SI	\$2680666	MARIA CARMONA	MOVIL_2	EMPRESAS
33333333333	345352525	GIL SUAREZ	JUAN ANDRES	CENTRO	574838	NO	\$13746389	LILIA HERRERA	MOVIL_1	PYMES

Fuente: Creación propia

Aquí existe la opción de descargar el archivo que se está visualizando y pueda ser utilizado en otras aplicaciones como Excel.

Ilustración 63 Descargar carterización



The screenshot shows a confirmation dialog box with the Movistar logo at the top. The text inside the dialog box asks '¿DESEA DESCARGAR LA CARTERIZACIÓN?'. Below the text, there are two buttons: 'ACEPTAR' and 'CANCELAR'.

Fuente: Creación propia

También está la opción de “consultar analistas” y “consultar cuadrillas”, la primera cuenta con atributos como: Código analista, ACD, analista, jefe de atención, profesional atención, segmento, cuadrilla principal. De igual manera se puede descargar el archivo.

Ilustración 64 Consultar analistas

COD ANALISTA	ACD	ANALISTA	JEFE ATENCIÓN	PROFESIONAL ATENCIÓN	SEGMENTO	CUADRILLA PRINCIPAL
00001	42563	RICARDO NARVAEZ	ANDRES MARTINEZ	ALEX CARDONA	CORPORACIONES	FIJA_2
00002	86745	MARIA CARMONA	ALEX CARDONA	MAURICIO CORDOBA	PYMES	MOVIL_3
00003	64489	LILIA HERRERA	ALEX CARDONA	JUAN ROJAS	PYMES	MOVIL_1

Fuente: Creación propia

La otra opción que existe es “consultar cuadrillas” y en ella se observan atributos como código de cuadrilla, grupo, analista 1, analista 2 y analista 3 ya que se maneja un modelo de atención con respaldo. Igualmente se puede descargar el archivo.

Ilustración 65 Consultar cuadrillas

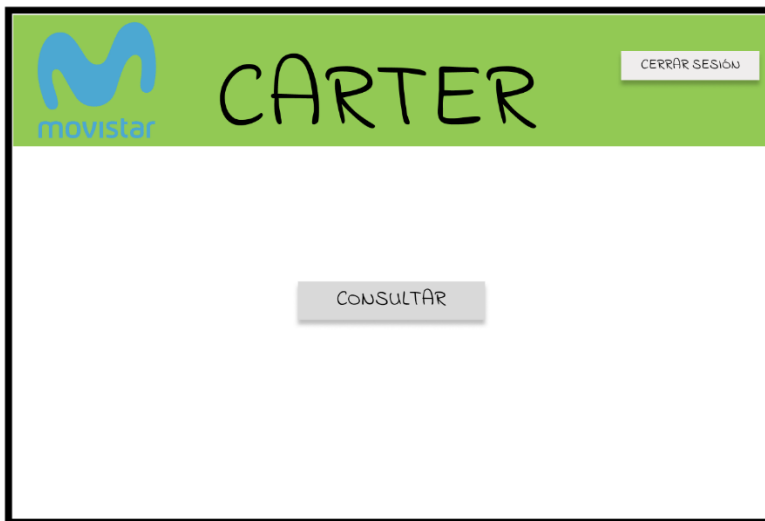
CODIGO	GRUPO	ANALISTA 1	ANALISTA 2	ANALISTA 3
00001	MOVIL_1	RICARDO NARVAEZ	MARIA CARMONA	LILIA HERRERA
00002	FIJA_1	MARIA CARMONA	RICARDO NARVAEZ	LILIA HERRERA
00003	CONVERGENTE_1	LILIA HERRERA	MARIA CARMONA	RICARDO NARVAEZ

Fuente: Creación propia

3.2.4.2 PERFIL DOS: CONSULTADOR

El siguiente perfil que existe es el de consultador, el cual únicamente tiene derecho a consultar sin realizar modificaciones, por ello el perfil principal de él se encuentra de esta manera.

Ilustración 66 Página principal Consultador y Modificador



Fuente: Creación propia

A partir de allí, si cuenta con las opciones de navegar para encontrar tanto la carterización como las estadísticas que fueron mostradas anteriormente.

3.2.4.3 PERFIL TRES: MODIFICADOR

El ultimo perfil, es el del modificador, la sección de login y la página principal es igual a la anterior del consultador, sin embargo, a partir de este momento presenta algunas diferencias, al acceder al botón de “consultar” se despliega el siguiente cuadro donde confirma si desea consultar o empezar a modificar la carterización.

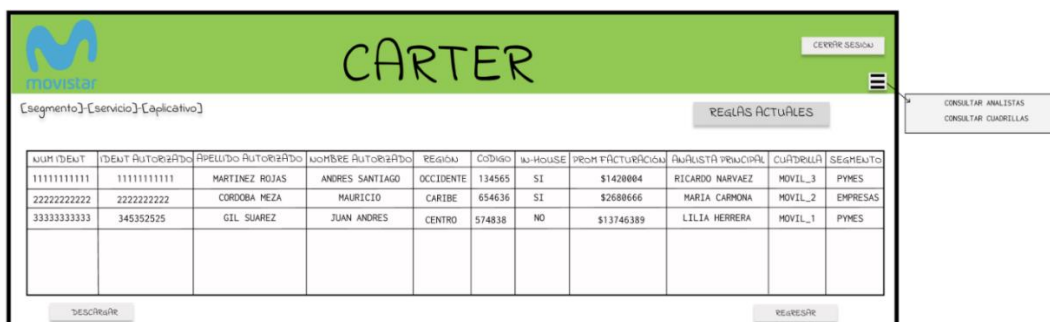
Ilustración 67 Página consultar del Modificador



Fuente: Creación propia

La navegación a través del botón de consultar es igual a las vistas anteriormente, el cambio que se observa es ahora como tal en el archivo donde está la carterización, ya que ahora hay un nuevo elemento, y es el de “reglas actuales”

Ilustración 68 Página carterización Modificador



Fuente: Creación propia

Al dar clic a esta opción se despliega la siguiente tabla en la cual están los atributos de id, nombre, fecha de modificación, descripción, usuario, segmento y las opciones de modificar o eliminar para cada regla. Esta tabla depende del segmento y aplicativo que se está consultando ya que cada una puede tener sus propias reglas. En esta opción también se puede crear nuevas reglas.

Ilustración 69 Consulta de reglas

ID REGLAS	NOMBRE	FECHA MODIFICACIÓN	DESCRIPCIÓN	USUARIO	SEGMENTO	MODIFICAR	ELIMINAR
0001	Prom_facturacion mayor a \$3.000.000	03/07/2023	Si el promedio de facturación de un cliente es mayor a \$3.000.000 será atendido por las cuadrillas 1,2 o 3	AJSOTELOMO	PYMES	MODIFICAR	ELIMINAR
0002	Cliente con In-house	19/08/2023	Si el cliente tiene in-house será atendido por las cuadrillas 7 u 8	AJSOTELOMO	PYMES	MODIFICAR	ELIMINAR

Fuente: Creación propia

Para la opción de “agregar nueva regla” se tiene un formato preestablecido que ayude a la creación de la regla, al crearla, el sistema evaluará si está correctamente diseñada y lo confirmará.

Ilustración 70 Nueva regla

+ NUEVA REGLA

NOMBRE

SI CLIENTE ATRIBUTO (CLIENTE) OPERADOR A

ENTONCES SERÁ ATENDIDO POR CUADRILLA O ANALISTA

DESCRIPCIÓN GENERAL DE LA REGLA

ACEPTAR

Fuente: Creación propia

Se hace la confirmación de que la regla fue creada correctamente.

Ilustración 71 Confirmación regla creada



Fuente: Creación propia

Ahora bien, la opción de cambiar una regla determinada es muy similar a la creación con la diferencia de que trae pre escrito la información actual de la regla, de igual manera al modificarla, el sistema realiza una validación y confirma el cambio.

Ilustración 72 Modificar regla

A screenshot of a web form titled "MODIFICAR REGLA" within a system called "CARTER". The form has a green header bar with the "CARTER" logo and a "CERRAR SESIÓN" button. The form fields include: "NOMBRE" with the value "Prom_facturacion mayor a \$3.000.000"; a section with a green "SI" button, "CUESTA" label, "ATRIBUTO (CUESTA)" dropdown set to "Prom_facturacion", "OPERADOR" dropdown set to "mayor", and a value field set to "3.000.000"; a section with a pink "ENTONCES" button, "SERÁ ATENDIDO POR" label, "CUADRILLA" dropdown, a radio button, and "AJUSTAR" dropdown; and a "DESCRIPCIÓN GENERAL DE LA REGLA" text area. An "ACEPTAR" button is at the bottom.

Fuente: Creación propia

Se hace la confirmación de que la regla fue modificada correctamente.

Ilustración 73 Confirmación regla modificada

Fuente: Creación propia

La opción de eliminar regla genera un cuadro de tipo pop-up donde solicita confirmación para eliminar la regla seleccionada.

Ilustración 74 Eliminar regla

Fuente: Creación propia

Se hace la confirmación de que la regla fue eliminada correctamente.

Ilustración 75 Confirmación regla eliminada



Fuente: Creación propia

Finalmente, está con la opción de logout, para ello se debe obturar en la opción de “cerrar sesión”, la cual aparece en la mayoría de las vistas y a partir de ello se presenta un ventana pop-up para confirmar la selección y redirigir a la sección de login.

Ilustración 76 Cerrar sesión



Fuente: Creación propia

3.2.5 DIAGRAMA DE DESPLIEGUE

El diagrama de despliegue UML es un diseño en el cual se relaciona el software que se utiliza con el sistema físico que se necesita para que funcione correctamente, en otras palabras, relaciona el software con las necesidades físicas para su correcto uso.

Hacer este diagrama trae diferentes beneficios, por ejemplo, determina el sistema operativo, las interfaces y protocolos para la comunicación del sistema, al hacer esto se puede determinar qué tan robusto debe ser el sistema para así minimizar la posibilidad de fallas [30].

Un diagrama de despliegue está compuesto principalmente por cuatro símbolos:

✓ Artefactos:

Es un objeto tangible que se conecta a la programación del software, los artefactos se utilizan en los nodos.

✓ Nodos:

Es un objeto físico el cual permite la implementación de uno o más artefactos y que puede cambiar su valor, los nodos permiten la implementación del código junto a la transmisión de diferentes estructuras del sistema

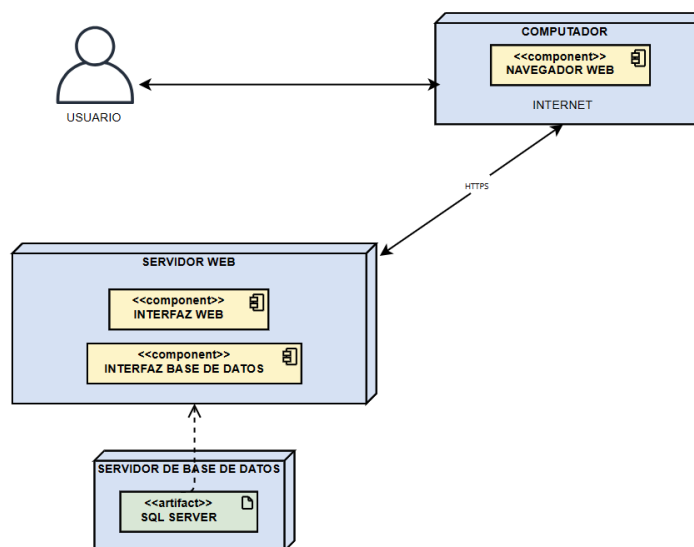
✓ Asociación

Se representa a través de una línea y estipula la comunicación entre los nodos.

✓ Estereotipo:

Es un dispositivo ubicado dentro del nodo el cual estipula el nombre de este y se representa con el nombre entre corchetes con flechas dobles.

Ilustración 77 Diagrama de despliegue

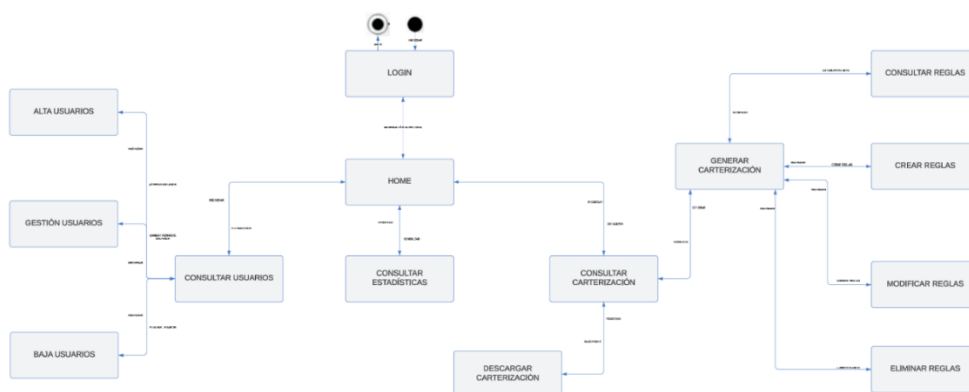


Fuente: Creación propia

3.2.6 MODELO DE NAVEGACIÓN

Un modelo de navegación es una herramienta que permite conocer gráficamente el flujo que hay en la navegación dentro de la aplicación, es de gran ayuda porque da una visión global de cómo se debe utilizar la aplicación para llegar al resultado deseado.

Ilustración 78 Modelo de navegación



Fuente: Creación propia

3.2.7 CONSIDERACIONES DE ACCESIBILIDAD Y USABILIDAD

Cuando se está haciendo el desarrollo de un software es importante tener en cuenta que hay diferentes tipos de usuarios que van a hacer uso de él, algún con diferentes tipos de discapacidades que también tienen el derecho de utilizar la aplicación [33].

Basados en esto surge dos conceptos: Usabilidad y Accesibilidad.

El primero hace referencia a la capacidad de un software de ser comprendido y amigable por un usuario, que sea intuitivo para el utilizarlo. Esto está definido en el estándar ISO 9126, y justamente busca que el software se ajuste al usuario y no que suceda, al contrario.

El segundo concepto, el de accesibilidad, busca que el diseño de una aplicación web pueda compensar las discapacidades de algunos usuarios, por ejemplo, si hay un usuario con discapacidad visual que quiera utilizar la herramienta, entonces el aplicativo puede adaptarse y brindar una solución.

En la siguiente tabla están algunas de las métricas de calidad más importantes.

Tabla 31 Accesibilidad y usabilidad

CONCEPTO	DESCRIPCIÓN	SOLUCIÓN
Accesibilidad	Usuarios con ceguera	Software debe leer el contenido en voz alta y/o incorporar funciones para la traducción del contenido a braille
Accesibilidad	Usuarios con visión débil	Software debe tener funciones de ampliación
Accesibilidad	Usuarios con deficiencia auditiva	Para el contenido auditivo, el software debe contener subtítulos legibles o transcripciones
Accesibilidad	Usuarios con deficiencia motora, o incapacidad para hacer uso de mouse	La primera solución es el desarrollo del software de manera robusta el cual tenga atajos de teclado, otra solución potencial es el uso de interfaces de voz
Accesibilidad	Usuarios con epilepsia fotosensibles	Software debe contar con la posibilidad de detener las animaciones que tenga la aplicación y los diferentes tipos de transiciones fuertes para así disminuir la posibilidad que sufran un ataque epiléptico

Usabilidad	Estructura	La estructura de la página debe ser de tipo jerárquica donde esté bien organizados los contenidos
Usabilidad	Diseño limpio	Utilizar tipografía, imágenes, recursos o gráficos claro que ayuden al usuario a encontrar rápidamente lo que desea
Usabilidad	El usuario tiene el control	El menú siempre debe estar accesible y visible, así se reduce la probabilidad que el usuario se sienta perdido en el uso de la aplicación.
Usabilidad	Simplificar y sintetizar	Se debe intentar tener lo necesario en la aplicación, si algo realmente no aporta al usuario lo mejor es quitarlo

Fuente: Creación propia

3.3 ANALISIS Y VALIDACIÓN DE RESULTADOS

Como se mencionó a lo largo del documento, el objetivo principal fue el diseño de un software (aplicativo web) que pueda disminuir el tiempo que se utiliza al realizar esta tarea, dado esto se utilizó la metodología de cascada mejorado para así llegar al objetivo, esta metodología está compuesta por 5 fases principales que ayuda a desarrollar una aplicación por completo incluso teniendo en cuenta el posterior mantenimiento que se deba dar. Sin embargo, como se enfocaba al diseño de la aplicación se hizo uso de las dos primeras fases, la fase de análisis y la fase de diseño. También se hizo uso de la metodología ágil de SCRUM para la gestión del proyecto haciendo uso de buenas prácticas, para esto se construyó un product backlog el cual está constituido por los requisitos que se debe cumplir para lograr el diseño del software, a partir del backlog también se construyó los sprints que permiten dividir las tareas que se deben cumplir.

Durante la fase de análisis, se realizó el levantamiento de requerimientos para así conocer que es lo que realmente necesita la aplicación, teniendo esto se pudo plasmar de manera más clara en las historias de usuario y así determinar cuáles son los requisitos funcionales y no funcionales que llevaría la aplicación, finalmente se procedió a detallar los casos

de uso de cada uno de los actores de la aplicación para así determinar cuáles son sus funciones y a que pueden tener acceso.

Después se inició con la fase de diseño, acá se cuenta con elementos más técnicos que surgen a partir de la información que se recolecto en la etapa anterior, por ejemplo, se realizó el diseño de la base de datos, acá se utilizó un modelo relacional el cual muestra de manera gráfica la distribución de las tablas en conjunto con sus relaciones, también se realizó las vistas del aplicación o los mockups, esta es una herramienta donde se diseña como se visualizaría la aplicación, después de realizo el diagrama de despliegue y el modelo de navegación, de igual manera se tuvieron en cuenta ciertas consideraciones para la accesibilidad y la usabilidad del aplicativo si este se llegara a desarrollar y poner en funcionamiento.

Como paso final, es importante dar a conocer los resultados obtenidos del diseño de una aplicación web que permita reducir el tiempo que conlleva la realización del proceso de carterización. Para esto se utilizó el método de entrevistas, el proceso se realizó de manera virtual donde se validó con dos personas para conocer su opinión acerca de la propuesta presentada. Para esta validación se utilizó la ayuda de Edwin Garavito y Maicol Mendieta, los cuales son personas que actualmente manejan el proceso dentro de la empresa. Primero se compartió el actual documento y se explicó en que se basa la propuesta presentada, seguido de eso se entregó un formato que consta de 6 preguntas que miden la satisfacción con respecto a los resultados presentados y una sección final de comentarios. Estas entrevistas se encuentran en el apéndice B.

Las entrevistas realizadas están construidas de manera cuantitativa ya que se pidió una calificación de 1 a 10 en cada pregunta, en promedio en la primera entrevista se obtuvo una calificación de 8.2, lo cual indica que la satisfacción es buena pero que se puede mejorar, para ello se solicitó una recomendación o comentario final.

La segunda entrevista cuenta con el mismo formato y en esta ocasión se obtuvo una calificación de 7.8 y de nuevo fue otorgada una retroalimentación.

4 CONCLUSIONES Y RECOMENDACIONES.

4.1 CONCLUSIONES

- ✓ La implementación de la metodología ágil para la gestión del proyecto en conjunto a la metodología tradicional fue de gran ayuda ya que brinda una alta rapidez en la elaboración del proyecto, así mismo permite dejar el trabajo en un punto de continuación que facilite la implementación de las demás fases de la metodología de ser necesario.
- ✓ EL diseño presentado cumple con los requerimientos especificados, así mismo, se proyecta como una solución para la automatización del proceso y mejorar los tiempos de ejecución del proceso, sin embargo, esto se puede conocer con seguridad si se llegara a construir e implementar.
- ✓ El proyecto aún tiene margen de crecimiento en la parte de diseño ya que se puede agregar mayor cantidad de funciones para otorgar más robustez y también en su desarrollo de ser aprobado.
- ✓ Un desafío importante que surgió fue el diseño de la base de datos, porque es clave tener una correcta base de datos que soporte toda la aplicación, esto implica que se debe analizar bien cuales son las tablas necesarias y como se van a relacionar para evitar duplicar la información de manera innecesaria.
- ✓ El diseño que se presento tuvo buena aceptación por parte de las personas que validaron el trabajo, dado esto tiene buenas probabilidades de que se continúe con el proyecto.

4.2 RECOMENDACIONES

Las siguientes recomendaciones están enfocadas para el proyecto actual y su implementación en caso de ser aprobado.

- ✓ Para la fase de implementación, es necesario construir el front-end que permita una interacción amigable con el usuario, para esto, se recomienda usar tecnologías como HTML, CSS y JavaScript, ya que, con diferencia, son las herramientas más populares y completas para el desarrollo del front-end.
- ✓ Así mismo, se recomienda el uso de frameworks como Bootstrap para la implementación de esta aplicación, el cual ayuda a minimizar el tiempo de elaboración y esta alineado con las buenas prácticas para la creación del front-end.
- ✓ Para la parte de back-end, se recomienda el uso de lenguajes como Python, el cual como se describió, es uno de los lenguajes de programación más utilizados y que es multipropósito, adicional a esto se recomienda el uso de frameworks como Django que permite el desarrollo de la aplicación web de una manera mucho más eficiente y que permite reducir el tiempo, además de traer incorporadas funciones comunes como la autenticación de usuarios.
- ✓ Como se mencionó anteriormente, para las bases de datos se recomienda de tipo relacional, con esto en mente, y teniendo en cuenta que el volumen de datos puede ser bastante alto, se recomienda usar gestores de bases de datos enfocados en Big Data como lo son Oracle, SQL Server o DB2 de IBM.
- ✓ Al ser una aplicación web, es necesario que haga uso de un servicio en la nube, para esto se recomienda hacer uso de AWS, el cual es el más popular y que ofrece diferentes servicios como EC2, el cual son servidores virtuales y que permite

almacenar la aplicación, también esta RDS el cual almacena la base de datos y Route 53 que funciona como DNS para asignar un dominio a nuestra página web.

- ✓ Otra herramienta importante es el editor de código, para esto se sugiere el uso de visual studio code, el cual tiene grandes ventajas y popularidad alrededor del mundo.

5 REFERENCIAS.

- 1] «Forbes,» [En línea]. Available: <https://www.forbes.com/lists/global2000/?sh=6191ac595ac0#tab:overall>.
- 2] «Telefonica,» [En línea]. Available: <https://www.telefonica.co/estructura-del-grupo>.
- 3] «Mckinsey,» [En línea]. Available: https://www.mckinsey.com/capabilities/operations/our-insights/the-automation-imperative#.
- 4] «Accenture,» [En línea]. Available: <https://www.accenture.com/lu-en/services/intelligent-automation-index>.
- 5] «Función Pública,» [En línea]. Available: <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=13960>.
- 6] «Función publica,» [En línea]. Available: <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981>.
- 7] «Fintechile,» [En línea]. Available: [https://www.fintechile.org/noticias/el-impacto-ambiental-de-la-digitalizacion#:~:text=Un%20an%C3%A1lisis%20de%20varios%20estudios,CO2%20de%202%2C5%20%25\)..](https://www.fintechile.org/noticias/el-impacto-ambiental-de-la-digitalizacion#:~:text=Un%20an%C3%A1lisis%20de%20varios%20estudios,CO2%20de%202%2C5%20%25)..)
- 8] «Tipos de investigación,» [En línea]. Available: <https://www.uv.mx/apps/bdh/investigacion/unidad1/investigacion-tipos.html>.
- 9] R. S. Pressman, Ingeniería del Software - Un Enfoque Practico, 7ma Edicion. ed., McGraw-Hill Companies.
- 10] «Repositorio USTA,» [En línea]. Available: <https://repository.usta.edu.co/bitstream/handle/11634/33594/2021andresbello.pdf?sequence=1&isAllowed=y>.

- «Repositorio USTA,» [En línea]. Available:
11] <https://repository.usta.edu.co/bitstream/handle/11634/21465/PlataRincon2020InformeFinal.pdf?sequence=3&isAllowed=y>.
- «Repositorio USTA,» [En línea]. Available:
12] <https://repository.usta.edu.co/bitstream/handle/11634/37671/2021carlostorres.pdf?sequence=1&isAllowed=y>.
- «Repositorio USTA,» [En línea]. Available:
13] <https://repository.usta.edu.co/bitstream/handle/11634/48259/2022BautistaRub%c3%a9n.pdf?sequence=1&isAllowed=y>.
- «ITCHA,» [En línea]. Available: <https://www.itcha.edu.sv/publicaciones/ITCHA/1167-2020-12-01/1167-ARTICULO---NUEVAS-TENDENCIAS-EN-DESARROLLO-WEB.pdf>.
14]
- «Programación de aplicaciones web,» [En línea]. Available:
15] https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjUt4KAyLT_AhXCnGoFHbNKCdcQFnoECAsQAQ&url=https%3A%2F%2Fsegiolujanmora.es%2Fverpdf%2F42&usg=AOvVaw0Q423lgLqFZM9bh7ayUszH.
- «riunet,» [En línea]. Available:
16] <https://riunet.upv.es/bitstream/handle/10251/75120/NIETO%20-%20Desarrollo%20de%20una%20aplicaci%C3%B3n%20web%2C%20con%20Front-end%20y%20Back-end%20para%20compraventa%20de%20segunda%20mano.pdf?sequence=1>.
- «autentia,» [En línea]. Available: https://www.autentia.com/wp-content/uploads/libros/Front_GuiaCompleta-Autentia.pdf.
17]
- «Desarrollo web,» [En línea]. Available: <https://blog.hubspot.es/website/framework-desarrollo-web>.
18]
- «Repositorio USTA,» [En línea]. Available:
19] <https://repositorio.une.edu.pe/bitstream/handle/20.500.14039/6052/MONOGRAF%C3%8DA%20-%20GALA%20YAULAPIN%20DOLLYBETH%20-%20FATEC.pdf?sequence=1&isAllowed=y>.
- «Aprende Python,» [En línea]. Available:
20] https://aprendepython.es/_downloads/907b5202c1466977a8d6bd3a2641453f/aprendepython.pdf.
- «tiobe,» [En línea]. Available: <https://www.tiobe.com/tiobe-index/>.
21]
- «dataprix,» [En línea]. Available: <https://www.dataprix.com/es/mineria-datos-aplicada-encuesta-permanente-hogares/24-origenes-y-antecedentes-bases-datos>.
22]
- «Acens,» [En línea]. Available: <https://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>.
23]

- «Agencia digital,» [En línea]. Available:
24] <https://www.occamagenciadigital.com/blog/las-mejores-metodologias-para-un-correcto-desarrollo-de-software>.
- «Open access,» [En línea]. Available:
25] https://openaccess.uoc.edu/bitstream/10609/9121/1/Intro_UML.pdf.
- «domestika,» [En línea]. Available: <https://www.domestika.org/es/blog/8390-como-crear-un-mockup-en-photoshop-3-tutoriales-gratis>.
26]
- «pixartprinting,» [En línea]. Available: <https://www.pixartprinting.es/blog/figma-que-es/>.
27]
- «Junta de Andalucía,» [En línea]. Available:
28] <https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/416>.
- «Biblioteca Digital,» [En línea]. Available:
29] <https://bibliotecadigital.usb.edu.co/server/api/core/bitstreams/7fb737ac-4e2e-4310-a2dd-f5d4c5ece874/content>.
- «Edrawsoft,» [En línea]. Available:
30] <https://www.edrawsoft.com/es/article/deployment-diagram-explained.html>.
- «Respositorio UNAB,» [En línea]. Available:
31] https://repository.unab.edu.co/bitstream/handle/20.500.12749/14408/2021_Tesis_Pedro_Jota_Regueros.pdf?sequence=1.
- «Ionos,» [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/diagrama-de-casos-de-uso/>.
32]
- «Genbeta,» [En línea]. Available: <https://www.genbeta.com/a-fondo/usabilidad-accesibilidad-web-que-que-se-diferencian-estos-conceptos-que-nos-facilitan-vida-como-usuarios>.
33]
- «Repositorio USTA,» [En línea]. Available:
34] <https://repository.usta.edu.co/bitstream/handle/11634/31494/2021sofiacarrillo.pdf?sequence=6&isAllowed=y>.
- «Repositorio UTP,» [En línea]. Available:
35] <https://repositorio.utp.edu.co/server/api/core/bitstreams/589d6902-16f7-4bc6-abeb-1acb1bbb0053/content>.
- «Escuela politecnica nacional,» [En línea]. Available:
36] <https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjRmeLJo92AAxWQQzABHd6rAHoQFnoECAkQAQ&url=https%3A%2F%2Fbibdigital.epn.edu.ec%2Fbitstream%2F15000%2F9129%2F3%2FCD-6083.pdf&usq=AOvVaw2sYL1NI6XKavWxsVFeKbO4&opi=89978449>.
- «udlap,» [En línea]. Available:
37] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf.

6 APENDICES.

6.1 Apéndice A

Entrevista a Maicol Mendieta Aguilar, profesional del aliado Atento encargado actual del proceso de carterización

Andrey: Buenas tardes, Maicol, la siguiente entrevista la realizo con el fin de recolectar información para conocer acerca del proceso de carterización y encontrar alguna manera para mejorar el proceso.

Maicol: Buenas tardes, Andrey, por supuesto no hay problema.

Andrey: ¿A que hace referencia el proceso de carterización?

Maicol: Bueno, podríamos definir que la carterización es la asignación de un analista a un cliente siguiendo una serie de reglas, aunque esta es una definición muy muy superficial. Es más, podría decir que un mejor concepto sería el de darle a un cliente, de acuerdo con el segmento al que pertenece, un modelo de atención. Si, esa definición siento que se acerca más a la realidad.

Andrey: ¿Y a qué tipo de reglas haces referencia?

Maicol: Cuando hablo de reglas, es evaluar, por ejemplo, en que segmento está el cliente. Dentro de Telefónica se manejan cierta clasificación de clientes, estas empresas fijas, empresas móviles, pymes fija y pymes móvil, y dentro de cada clasificación hay segmentos, por ejemplo, en empresas móvil contamos con cinco segmentos de atención: Corporaciones, empresas, empresas clúster 4, pymes C y pymes vertical. Cuando un cliente nuevo se incorpora a Telefónica y pertenece a empresas móvil, se asigna a uno de estos segmentos, entonces esto sería lo primero que tendría que evaluar, también podríamos evaluar si el cliente tiene in-house o no. Te voy a dar un ejemplo, para todos los segmentos, excepto corporaciones, también para todas las regionales menos noroccidente, de todos los jefes comerciales con In-house, se asigna a Jair Villamil como jefe de atención y German Hernández como profesional de atención. Entonces se deben tener en cuenta diferentes reglas para determinar el analista.

Andrey: Ya veo, entiendo que hay que tener en cuenta muchas condiciones antes de hacer la asignación, ¿Ha realizado la cuenta de cuantas reglas debe tener en cuenta?

Maicol: Pues no tengo el dato exacto, pero pensaría que aproximadamente 60 reglas o condiciones son las que debo tener en cuenta

Andrey: ¿y este proceso lo ejecuta solo? ¿No hay nadie más?

Maicol: Pues principalmente yo soy el que la realizo, aunque no estoy encargado de todo, por ejemplo, Edwin Garavito, maneja la asignación en el segmento de empresas clúster 4 dentro de la móvil y así hay otras personas que manejan la asignación, pero de manera más pequeña o en un segmento en concreto.

Andrey: Tengo entendido que los analistas encargados de contactarse con los clientes pertenecen a empresas aliadas de Telefónica, ¿es verdad?

Maicol: Efectivamente, hay dos empresas aliadas llamadas Atento y Comdata, de hecho, actualmente trabajo para Atento, y esto es importante a la hora de realizar con la carterización, porque hay que repartir el 50% de las transaccionalidades entre ambos aliados.

Andrey: ¿Y cada cuanto se debe realizar este proceso?

Maicol: Pues en teoría se debe realizar cada semana, sin embargo, prácticamente todos los días salen nuevos requerimientos de actualización tal contacto y cambiar el contacto autorizado, o arreglar la carterización porque quedo mal, y así, diversos errores que pueden salir.

Andrey: ¿Y en promedio cuánto tiempo lleva realizar esta asignación?

Maicol: La actualización semanal lleva unos dos días porque hay que actualizar toda la base a parte del tiempo que lleva arreglar los errores que se presentan.

Andrey: Bueno Maicol, le agradezco por el tiempo y por brindarme esta información, feliz tarde.

6.2 Apéndice B

Entrevista de Opinión sobre el Diseño de Software “Carter”

Información del Entrevistado:

- Nombre: Edwin Garavito
- Cargo: Profesional de atención clientes TOP
- Compañía u Organización: Telefónica

- Fecha de la Entrevista: 16/08/2023

Introducción:

- Con esta entrevista, se busca medir el nivel de satisfacción actual con el diseño presentando para la automatización del proceso de Carterización que se maneja en Telefónica, se agradece responde con sinceridad ya que la retroalimentación permite conocer cuáles son los puntos por mejorar de cara a su implementación.

Preguntas:

1. En una escala del 1 al 10, ¿Considera que el levantamiento de requerimientos abarca en su totalidad la especificación del proceso?

RTA: **8**

2. En una escala del 1 al 10, ¿Considera que los requisitos tanto funcionales (obligatorios para la aplicación) como no funcionales (le dan calidad a la aplicación) son los correctos?

RTA: **9**

3. En una escala del 1 al 10, ¿Considera que las funciones determinadas para cada actor en el sistema son las correctas?

RTA: **9**

4. En una escala del 1 al 10, ¿Qué tan intuitivo en términos de navegación (teniendo en cuenta los Mockups) considera que es la aplicación?

RTA: **7**

5. En una escala del 1 al 10, ¿Qué tan probable es la incorporación de esta aplicación para la automatización del proceso?

RTA: **8**

6. ¿Cuáles son sus sensaciones generales del diseño presentado?

RTA: De manera general, la aplicación cumple con lo que promete, pero creo que falta implementar más servicios que se manejan en la empresa que, aunque no son tan utilizados igual si se pudieran tener en cuenta. También creo que incorporarlo sería un gran beneficio porque este proceso se utiliza mucho pero también gasta mucho tiempo

De antemano, se agradece la respuestas dadas, son altamente valoradas y permiten validar el diseño presentado.

Entrevista de Opinión sobre el Diseño de Software “Carter”

Información del Entrevistado:

- Nombre: Maicol Mendieta
- Cargo: Profesional de atención clientes Corporativo
- Compañía u Organización: Atento, aliado Telefónica
- Fecha de la Entrevista: 16/08/2023

Introducción:

- Con esta entrevista, se busca medir el nivel de satisfacción actual con el diseño presentando para la automatización del proceso de Carterización que se maneja en Telefónica, se agradece responde con sinceridad ya que la retroalimentación permite conocer cuáles son los puntos por mejorar de cara a su implementación.

Preguntas:

1. En una escala del 1 al 10, ¿Considera que el levantamiento de requerimientos abarca en su totalidad la especificación del proceso?

RTA: **8**

2. En una escala del 1 al 10, ¿Considera que los requisitos tanto funcionales (obligatorios para la aplicación) como no funcionales (le dan calidad a la aplicación) son los correctos?

RTA: **7**

3. En una escala del 1 al 10, ¿Considera que las funciones determinadas para cada actor en el sistema son las correctas?

RTA: **9**

4. En una escala del 1 al 10, ¿Qué tan intuitivo en términos de navegación (teniendo en cuenta los Mockups) considera que es la aplicación?

RTA: **6**

5. En una escala del 1 al 10, ¿Qué tan probable es la incorporación de esta aplicación para la automatización del proceso?

RTA: **9**

6. ¿Cuáles son sus sensaciones generales del diseño presentado?

RTA: Desde mi punto de vista, la aplicación sería de gran ayuda para todo el proceso de carterización porque es un proceso de mucho cuidado y que lleva mucho tiempo. Me gustaría que se mejorara el diseño de la aplicación, no siento que sea muy intuitiva y tiene muchos pasos.

De antemano, se agradece las respuestas dadas, son altamente valoradas y permiten validar el diseño presentado.