

PROYECTO
SGAL
SISTEMA DE GESTIÓN DE ALMACENES

JUAN DAVID DUQUE

Asesor:

MYRIAM CASTRO CÁRDENAS

UNIVERSIDAD EAFIT
ESCUELA DE INGENIERÍAS
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS
MEDELLÍN, COLOMBIA

2016

Tabla de contenido

Tabla de contenido	2
Listado de tablas	5
Listado de ilustraciones	7
Listado de ecuaciones	9
Listado de anexos	10
1 Introducción	11
2 Objetivos	14
2.1 Objetivo general	14
2.2 Objetivos específicos.....	14
3 Almacenes intervenidos	15
4 Metodología Aplicada.....	16
4.1 Análisis de requisitos:	19
4.2 Diseño.....	27
4.3 Implementación.....	34
4.4 Pruebas del sistema	37
5 Documentos de ingeniería de software	40
5.1 Documento de requisitos	40
5.1.1 Objetivos del sistema	40
5.1.2 Detalle de las necesidades.....	41
5.1.3 Análisis de los requisitos	47
5.1.4 Actores	67
5.1.5 Requisitos Funcionales	68
5.1.6 Requisitos No Funcionales	82

5.2	Documento de Casos de uso.....	84
5.2.1	Administración.....	84
5.2.2	Cuentas por cobrar	92
5.2.3	Facturación.....	99
5.2.4	Inventario	107
5.2.5	Cotizaciones.....	112
5.2.6	Cuentas x pagar.....	115
5.2.7	Reportes	118
5.3	Prototipos no funcionales.....	127
5.3.1	Administración de clientes.....	127
5.3.2	Administración de clientes.....	128
5.3.3	Administración de crédito.....	129
5.3.4	Administración de ítems del inventario	130
5.3.5	Administración de facturas	131
5.3.6	Administración de cotizaciones	132
5.4	Documento de arquitectura de la solución	132
5.4.1	Definiciones	133
5.4.2	Acrónimos.....	134
5.4.3	Arquitectura Base.....	134
5.4.4	Representación arquitectónica	135
5.4.5	Restricciones especiales.....	137
5.4.6	Herramientas de desarrollo	137
5.4.7	Tecnología empleada en el desarrollo.....	138
5.4.8	Interoperabilidad	138
5.4.9	Necesidades de parte del cliente	138
5.4.10	Otras decisiones arquitectónicas	139
5.4.11	Vista de despliegue	139
5.4.12	Nodos y sistemas.....	140
5.4.13	Identificación y descripción de modelos y módulos.....	142
5.4.14	Vista lógica	146
5.4.15	Orden de implementación.....	150
5.4.16	Vista de datos.....	151
5.4.17	Calidad	152
5.5	Documento de Diseño detallado de la solución	156

5.5.1	Diagrama de datos.....	157
5.5.2	Diseño de módulos/subsistemas	157
5.6	Soluciones Similares a las necesidades	186
5.6.1	World office	186
5.6.2	Affinity.....	187
6	Conclusiones	188
7	Bibliografía	189
8	Anexos	194

Listado de tablas

Tabla 1. Actores	67
Tabla 2. Administrar clientes	85
Tabla 3. Administrar parámetros.	86
Tabla 4. Administrar perfiles.	87
Tabla 5. Administrar ítems.	88
Tabla 6. Administrar servicios.....	90
Tabla 7. Administrar codeudores	92
Tabla 8. Administrar créditos.	94
Tabla 9. Administrar cuotas.....	95
Tabla 10. Aplicar interés por mora.	96
Tabla 11. Recordar pagos de cuotas.	97
Tabla 12. Informar cuotas atrasadas a clientes.	98
Tabla 13. Administrar facturas.	99
Tabla 14. Controlar rangos de facturación.....	100
Tabla 15. Imprimir detalle de factura.	102
Tabla 16. Aplicar descuentos en ítems de facturas.	102
Tabla 17. Separar IVA del valor de las facturas.	103
Tabla 18. Administar servicios adicionales.	104
Tabla 19. Agregar servicios a las facturas.	105
Tabla 20. Generar factura desde una cotización.	105
Tabla 21. Generar código único.....	107

Tabla 22. Consultar stock de artículos.....	108
Tabla 23. Registrar lotes de entrada.....	109
Tabla 24. Registrar salida de artículo.	110
Tabla 25. Descontar ítems de inventario.....	111
Tabla 26. Generar cotización.	112
Tabla 27. Imprimir cotización.....	113
Tabla 28. Exportar cotización.	114
Tabla 29. Administrar facturas por pagar.	115
Tabla 30. Alertar vencimiento de facturas por pagar.....	117
Tabla 31. Listar lotes de entrada en un rango de fechas.	118
Tabla 32. Listar clientes con cuotas atrasadas.	120
Tabla 33. Exportar listado de clientes con cuotas atrasadas.	121
Tabla 34. Listar crédito de cliente.....	122
Tabla 35. Listar historial de pagos de clientes.....	123
Tabla 36. Exportar crédito del cliente.....	124
Tabla 37. Listar cuentas por pagar.....	125
Tabla 38. Exportar listado de cuentas por pagar.....	126
Tabla 39. Definiciones.....	133
Tabla 40. Acrónimo.	134
Tabla 41. Vista lógica.	150
Tabla 42. Tabla para recorrer datos de intereses por mora.	182
Tabla 43. Tabla para recorrido de datos de recordatorio de pago de cuotas.....	184

Listado de ilustraciones

Ilustración 1 Prototipo no funcional - Administración de clientes	127
Ilustración 2 Prototipo no funcional - Administración de clientes	128
Ilustración 3 Prototipo no funcional - Administración de crédito	129
Ilustración 4 Prototipo no funcional - Administración de ítems del inventario.....	130
Ilustración 5 Prototipo no funcional - Administración de facturas.....	131
Ilustración 6 Prototipo no funcional - Administración de cotizaciones.....	132
Ilustración 7. Arquitectura base.	135
Ilustración 8. Vista de despliegue.	140
Ilustración 9. Vista de implementación.	148
Ilustración 10. Diagrama de datos.	157
Ilustración 11. Administrar parámetros.	160
Ilustración 12. Administrar parámetros funcionales.	165
Ilustración 13. Administrar consultas dinámicas.	167
Ilustración 14. Administrar notificaciones.....	168
Ilustración 15. Administrar perfiles por aplicación.	170
Ilustración 16. Query base "Estado crédito en curso".....	172
Ilustración 17. Proceso para valor de cancelación.	174

Listado de diagramas

Diagrama 1 Diagrama en cascada.....	17
Diagrama 2 Casos de uso – Administración.....	84
Diagrama 3 Diagrama casos de uso - Cuentas por cobrar	92
Diagrama 4 Diagrama casos de uso - Facturación.....	99
Diagrama 5 Diagrama de caso de uso - Inventario.....	107
Diagrama 6 Diagrama de casos de uso - Cotiozaciones	112
Diagrama 7 Diagrama de casos de uso - Cuentas por pagar.....	115
Diagrama 8 Diagrama de casos de uso - Reportes.....	118

Listado de ecuaciones

Ecuación 1. Calcular cuota inicial de créditos.....	72
Ecuación 2. Intereses.	182

Listado de anexos

Anexo 1 Código fuente	194
Anexo 2 Instaladores de aplicación	194
Anexo 3 Base datos.....	194

1 Introducción

Este proyecto de grado, ha nacido luego de evaluar los procesos de ventas, compras y manejo de inventario de varios almacenes, cuya finalidad es la distribución de motos, estos almacenes poseen procesos similares y por ende, tienen falencias similares.

Aplicando los conocimientos de Ingeniería de Software, se logra comprender cómo se puede mejorar un proceso a través de la sistematización, para lograr un mejor desempeño, utilizando tecnologías informáticas que ofrecen diversas soluciones exitosas para el manejo de los flujos de información.

La ingeniería de Software, es una disciplina de la ingeniería que se enfoca en todos los aspectos de la producción de Software, a diferencia de las ciencias de la computación, la Ingeniería de Software se especializa en las prácticas de desarrollo de un “buen software”, lo cual se puede lograr si el Software cumple con las especificaciones requeridas, de manera confiable, usable, mantenible y con un buen rendimiento (Sommerville, 2014).

Para este proyecto, se planea utilizar una tecnología que permita el manejo de datos a través de internet, sin necesidad de tener equipos propios con administraciones complicadas, que además por un bajo costo mensual, potencie el flujo del negocio a la red global, buscando de esta manera que exista acceso a la información de manera en tiempo real y globalizada.

Actualmente, en los procesos de los almacenes existen limitantes que generan la necesidad de un sistema de información. Las principales limitantes que se encontraron en los procesos actuales son:

- No existen copias de la información para recuperar la pérdida o corrupción de datos.
- No existe la posibilidad de visualizar en tiempo real la información, la generación de informes
- No hay información de balance sistematizada, debe realizarse a partir de las facturas físicas.

Por ello, después de un análisis de la situación actual, se ha decidido plantear el inicio de una solución sistematizada que modernice los procesos actuales y que automatice la mayoría de procedimientos que se llevan a cabo en los almacenes, ya que la manera en que se realizan en este momento representa un riesgo y una desventaja a la hora de analizar el comportamiento de las ventas. Con los procesos modernizados y sistematizados, se espera tener la información disponible para crear acciones que representen una ventaja competitiva en el mercado.

La solución que se planteará a continuación, además de minimizar las falencias propias de los procesos intervenidos, busca flexibilidad para ajustarse a procesos similares de otros almacenes, de esta manera no solo aplicará para los almacenes intervenidos, sino que por medio de configuraciones al sistema, se puedan intervenir procesos similares de otros almacenes que presenten falencias similares, permitiendo la comercialización del producto a un precio asequible en el mercado de las pymes.

Para el desarrollo de este producto, se ha realizado un proceso de ingeniería de Software, aplicando la metodología tradicional de desarrollo, en la que se incluyen las fases de análisis de requisitos, diseño de la solución, implementación y pruebas del sistema.

2 Objetivos

2.1 Objetivo general

Aplicar los conceptos de Ingeniería de Software, para realizar un proyecto que produzca un sistema de información enfocado en el proceso de ventas, compras y manejo de inventarios en almacenes de ventas de motos.

2.2 Objetivos específicos

Aplicar la teoría de Ingeniería de Software, para determinar cuál de los posibles ciclos de vida de los proyectos, puede regir el desarrollo de este proyecto.

Especificar las actividades, hitos, entradas y salidas de cada una de las fases del ciclo de vida seleccionado para el desarrollo del proyecto.

Especificar certificaciones técnicas deseables en los integrantes de los equipos que ejecutan cada fase del ciclo de vida seleccionado para el desarrollo del proyecto.

3 Almacenes intervenidos

Los almacenes que se intervinieron, para realizar el proceso de ingeniería de Software y construir un sistema de información que apoye los procesos, son los siguientes:

- **Yamaha Alto del Rosario**
 - Actividad comercial: Venta de motos, repuestos y accesorios, servicio de taller y reparación de motos, con exclusividad para la marca Yamaha.
 - Antigüedad: Aproximadamente 30 años.
 - Número de empleados: 4
 - Ubicación: Honda – Tolima

- **Santa Fe Motor**
 - Actividad comercial: Venta de motos, repuestos y accesorios, con exclusividad para la marca Auteco
 - Antigüedad: Aproximadamente 15 años
 - Número de empleados: 3
 - Ubicación: Santa Fe de Antioquia - Antioquia

4 Metodología Aplicada

La Ingeniería de Software define ciclos de vida para el desarrollo de sistemas, cada ciclo define las actividades y las tareas involucradas en el desarrollo, la exploración y el mantenimiento del producto de Software (ISO 12207-1, 2013), escoger el ciclo de vida depende de las necesidades y propiedades únicas de cada proyecto, los ciclos más conocidos son:

- **Modelo en cascada:**

Este es un modelo riguroso, que se define por fases, y cada fase debe estar cerrada para continuar con la siguiente fase, las fases que intervienen en el modelo son: análisis de requisitos, diseño, implementación, pruebas y mantenimiento.

- **Modelo de prototipos**

Este es un modelo de construcción rápida, en donde se construyen funcionalidades donde el usuario y el desarrollador estén de acuerdo, para minimizar el riesgo en el desarrollo

- **Modelo en espiral**

Este es un modelo en el que se descomponen los módulos o propiedades del Sistema y se entregan al cliente en etapas que comprenden el ciclo en cascada, cada iteración se ejecuta de nuevo todo el modelo, y al final se obtiene un producto integrado y validado por el cliente en cada iteración.

- **Modelo RAD:**

Es un modelo de aplicación rápida del desarrollo, en el que se usa un ciclo corto de desarrollo que se basa en el entendimiento del requisito, la implementación y las pruebas.

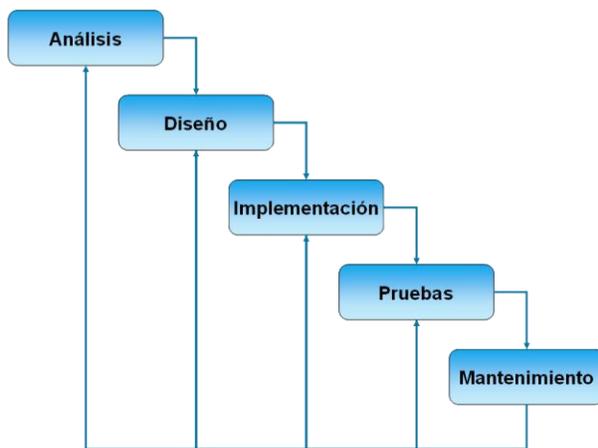
- **Otros modelos:**

Otros modelos que existen son: Modelo de desarrollo por etapas, Modelo incremental o estructurado, Modelo orientado a objetos, Modelo de desarrollo concurrente (Wikipedia, 2016).

Este proyecto está concebido utilizando una metodología de software tradicional o cascada, esta comprende las siguientes grandes fases:

El ciclo de vida del proyecto está dado por el modelo en cascada, este establece que cada fase debe estar cerrada e inspeccionada antes de continuar con la siguiente, y en caso de encontrar algún error, se deben reabrir las fases anteriores involucradas para la corrección, desde fases de diseño, implementación y pruebas, se hizo una excepción a la metodología y se continúan con las siguientes fases cuando se cierra un módulo completo.

Diagrama 1 Diagrama en cascada



Fuente: Tomado de Metodología en Cascada (Solorio, 2013).

Cada fase tiene programadas actividades e hitos, que todas son controladas desde un cronograma que regula el costo y el tiempo invertido en el proyecto, al aplicar toda la metodología, se puede obtener un sistema robusto y de fácil mantenimiento, que cumpla todas las necesidades del usuario.

Una de las definiciones técnicas de ingeniería de software que más se ajusta a este proyecto por su alcance es: “La ingeniería del software es el proceso formal de desarrollo de software en el que las necesidades del usuario se traducen en requerimientos, estos se transforman en diseño que se implementa en código que se prueba, documenta y se certifica para su uso operativo”. (Fernández, 2013)

Con la aplicación de este modelo en el proyecto, se minimiza el riesgo de encontrar bugs en las fases adelantadas, los cuales representan un alto costo cuando son encontrados (Sánchez Rios, 2016) (Ortiz, 2016), este costo está investigado y documentado en varias empresas, la siguiente tabla muestra cuantas veces más cuesta encontrar un error en una etapa posterior por empresa:

- IBM - Codificación: 1.5; Pruebas: 60; Producción: 100
- Boehm - Diseño: 1; Desarrollo: 15 a 40; Pruebas: 30 a 70; Producción: 40 a 1000
- Remus - Diseño: 1, Desarrollo: 20, Pruebas: 82
- Ackerman - Pruebas: 2 - 10 veces por inspección
- Russell - Inspección: 1, Pruebas: 2 a 4, Producción: 33

- PSP Research – Hasta Pruebas, toma 12 veces más encontrar y corregir un error que en revisiones de código. (Maxim, 2014)

4.1 Análisis de requisitos:

La fase de análisis de requisitos tiene como objetivo identificar todas las necesidades del proyecto, y documentar estas necesidades.

La persona encargada de licitar los requisitos con el cliente debe conocer cabalmente el negocio correspondiente al proyecto, poder hablar con propiedad con los usuarios finales bajo los mismos términos de negocio, obtener información del ambiente del proyecto, detectar incongruencias que puedan causar conflictos entre requisitos, aportar sugerencias que pueden mejorar la experiencia de usuario en el proyecto simplificando los procesos, identificar las necesidades estratégicas y ahondar hasta conocer el negocio y su funcionamiento de forma general y detallada.

En el reporte Standish Group de 2015 (Hastie & Wojewoda, 2015), se hace un seguimiento de proyectos de Software a nivel mundial y se obtienen diferentes análisis y resultados de estos, se menciona que una de las fallas que más afecta el ciclo de vida de un proyecto es la

inexperiencia de las personas encargadas de la elicitación, es por esto, que un proyecto debe tener un personal idóneo según su tamaño y complejidad de este, buscando disminuir los errores inyectados en esta fase, ya que es la primera etapa y constituye toda la base del sistema. Una de las certificaciones más recomendadas a nivel mundial para el equipo de trabajo que participe en esta fase es la CPRE de IREB (International requirements engineering boards, 2016), que certifica a la persona de por vida como profesional elicitor de requisitos.

Las salidas que produce esta fase son las siguientes:

- a. Documento de especificación de requisitos.
- b. Documento de casos de uso
- c. Prototipos no funcionales (Bosquejos) del sistema

a. Documento de especificación de requisitos

El documento de especificación de requisitos, como su nombre lo indica, contiene todos los requisitos funcionales y no funcionales que puede llegar a encontrar la persona elicitor, adicionalmente, como buena práctica, se recomienda identificar el proceso de la organización que intervendrá el sistema, esto con el objetivo de conocer el entorno, las acciones, los involucrados y las relaciones que se deben tener en cuenta en esta fase y en las siguientes.

Los requisitos funcionales se refieren a los servicios que proveerá el Software, la manera cómo reaccionará ante ciertas entradas y en algunos casos particulares, a cosas específicas que el

sistema no debe hacer. Para obtener estas definiciones y servicios, la persona que elicitó los requisitos puede utilizar alguna de las siguientes técnicas:

- **Entrevistas**

Con esta técnica, la persona debe licitar los requisitos por medio de entrevistas guiadas, enfocando las preguntas realizadas a 3 áreas principalmente:

- I. **Usuario:** para formar una idea del tipo de información que se generará en esta área, la persona elicitadora debe responder las siguientes preguntas: ¿Quién es el cliente?, ¿Quién es el usuario?, ¿Son sus necesidades diferentes?, ¿Cuáles son sus habilidades, capacidades, ambiente? Una vez tenga identificados estos aspectos, podrá determinar qué tan confiable es la información que proviene de cada entrevista con los usuarios.
- II. **Proceso:** para conocer el proceso que intervendrá el sistema, es necesario realizar preguntas como: ¿Cuál es la razón por la que se quiere resolver este problema?, ¿Cómo se resuelve el problema actualmente?, ¿Quiénes intervienen en el proceso?, ¿Qué actividades se realizan durante el proceso?, una vez identificado el proceso, se tendrá una visión completa de los objetivos del sistema.
- III. **Sistema:** Para conocer las necesidades del sistema, es necesario realizar preguntas como: ¿Qué necesita el usuario?, ¿Por qué lo necesita?, ¿Qué información previa requiere lo que necesita?, ¿Quién puede acceder a la información que genera la necesidad?, ¿Qué información generará lo que necesita?, es muy importante que la persona elicitadora concentre el esfuerzo de la entrevista en las actividades y acciones identificadas en el proceso, o que intervengan en este, ya que si no se

hace, se pueden estar añadiendo necesidades que probablemente no aporten valor al sistema y solo incrementen costos.

Esta técnica es muy útil, sin embargo, depende de la habilidad de la persona elicitadora para comprender el ambiente, procesar la información obtenida y resolver los problemas potenciales en las necesidades descritas.

- **Lluvia de ideas**

Esta técnica se utiliza en empresas donde varias personas intervengan en el mismo proceso, o en donde se necesite generar nuevas ideas, métodos o soluciones para una necesidad. Las condiciones a tener en cuenta en la aplicación de esta técnica son:

- Realizar sesiones en grupo, para que la producción de ideas sea más efectiva.
- Encontrar un lugar tranquilo en donde realizar la sesión de lluvia de ideas
- Tomar nota de todas las ideas que se generen
- No realizar críticas a las ideas que surjan, ya que esto puede ser un inhibidor de nuevas ideas
- Llegar con conocimientos previos del tema a generar ideas, para asegurar que se entienda el concepto de la idea.

Es importante que la lluvia de ideas sea una sesión moderada, y asegurar que todos participantes hablen y se expresen libremente, para crear una atmosfera en donde los participantes se sientan seguros y no restrinjan su producción de ideas

- **Cuestionarios**

Esta técnica se utiliza para un público en general, donde son muchos los intervenidos y es complicado realizar alguna de las dos técnicas anteriores. Las encuestas que se generen deben responder preguntas tanto del cliente, del usuario, del proceso, de las necesidades del sistema, y del tipo de usuario que intervienen en el proceso, para poder determinar la veracidad de la información (Buenas tareas, 2016)

- **Otras:**

Existen otras técnicas como: Desarrollo conjunto de aplicaciones (JAD), desarrollo de prototipos, observación, escenarios, que permiten enfocar a la persona elicitadora de requisitos en distintas estrategias para lograr una mejor abstracción de las necesidades del cliente (García, 2007).

Los requisitos no funcionales, se refieren a los requisitos relacionados a las propiedades emergentes del sistema, como la fiabilidad, los tiempos de respuesta, la capacidad de almacenamiento, las restricciones del sistema debido a los ambientes de despliegue, como capacidad de dispositivos o interfaces con otros sistemas. En esta identificación pueden intervenir áreas como:

- Sistema: en esta área se deben identificar propiedades del sistema como: escalabilidad, tiempos de respuesta de la aplicación, uso de memoria, portabilidad, disponibilidad del

sistema, licenciamientos necesarios por componentes externos, cantidad de idiomas que soporte el sistema, entre otros inherentes al sistema.

- Organizacionales: en esta área se deben identificar lineamientos organizacionales de ambas partes (cliente y proveedor) que afecten el desarrollo o el despliegue del sistema.
- Externos: en esta área se identifican requisitos legales que intervengan en el sistema o puedan generar una “sanción” al cliente o al proveedor que desarrolla el sistema.

Los diagramas UML¹ que se pueden usar en este documento, para apoyar el entendimiento de las necesidades del sistema, son los siguientes:

- Diagrama de actividades: este diagrama puede representar el flujo del proceso, con sus actividades y decisiones, permitiendo que se observe el comportamiento entre dependencias, actividades, resultados y condicionales.
- Diagrama de estados: este diagrama puede representar los estados que puede tomar una entidad, y los posibles flujos para llegar a otros estados, se debe establecer de manera clara el estado inicial y el estado final para no complicar el diagrama con las transiciones.

Diagrama de secuencia: Este diagrama puede describir secuencias complicadas de las actividades, mostrando las entidades que intervienen, las acciones y resultados que se obtienen de cada intervención y claramente el inicio y el fin de la actividad.

¹ UML: lenguaje unificado de modelado, es un lenguaje para hacer modelos y es independiente de los métodos de análisis y diseño.

Aplicación documento de requisitos

Para la elicitación de requisitos en este proyecto se utilizó la técnica de Entrevistas, obteniendo toda la información de proceso y necesidades del sistema, así como la identificación de requisitos no funcionales a tener en cuenta en la construcción. Para documentar el resultado obtenido, se utilizó un editor de texto y el documento se encuentra en el Anexo 5.1 Documento de requisitos

b. Documento de casos de uso

Los casos de uso son especificaciones del sistema, redactadas en secuencias de acciones realizadas por entidades, estos se documentan teniendo en cuenta los siguientes aspectos:

- Actor: es una persona o entidad que desempeña un papel en la interacción con el sistema
- Acción: es lo que sucede cuando el actor interactúa con el sistema para lograr un objetivo, como buena práctica esta acción debe empezar con un verbo en infinitivo.
- Asociación: es una relación entre los 2 anteriores, puede ser actor-acción o acción-acción, está última tiene 3 formas de relación: Asociación, Dependencia y Generalización.
- Escenarios: este define el comportamiento ante la finalización de la acción, ya sea que finaliza con éxito o fallido.
- Límite: es la frontera en la cual la acción se ejecuta y define su alcance.

Se debe tener en cuenta que un caso de uso siempre debe describir el objetivo que se cumplirá cuando el usuario ejecute la acción, también las interacciones con otros actores y otros casos de uso, los casos de uso deben estar descritos en lenguaje sencillo, ya que los usuarios no están familiarizados normalmente con este tipo de documentación, especialmente en acciones complejas del sistema, y por último, el objetivo del caso de uso es describir que debe hacer el sistema y sus interacciones con los actores, no se debe mencionar en ningún momento el cómo debe hacerlo, ya que se infringe las buenas prácticas y se limita el sistema desde la concepción.

Aplicación documento de casos de uso

Para la documentación de los casos de uso en este proyecto, se utilizó un editor de texto y el diagramador YUML (Yuml, 2016), el cual es capaz de generar diagramas de casos de uso con actores y acciones a través de líneas de texto que se definen,. El documento se encuentra en el anexo 5.2 Documento de Casos de uso.

c. Prototipos no funcionales

Los prototipos no funcionales, son vistas de las funcionalidades basadas en las especificaciones de los requisitos, con el objetivo de guiar al cliente claramente, mostrando una visión del sistema y sus acciones, y poder validar los requisitos funcionales definidos.

Se debe tener cuidado al construir prototipos no funcionales, ya que estos ofrecen ventajas al momento de la validación con el cliente, pero también pueden ser objeto de discusiones si el usuario final no entiende el objetivo del prototipo no funcional, adicionalmente, requiere de más

tiempo de documentación y si no se realizan de manera adecuada, pueden desviar la atención del usuario y desarrolladores en aspectos poco importantes, dejando de lado los estratégicos.

Algunas herramientas permiten la construcción avanzada de prototipos, como por ejemplo Balsamiq Mockups, esta permite crear de manera sencilla y en bosquejo, las vistas del sistema, las acciones que el usuario pueda realizar y el desplazamiento entre vistas o acciones mostradas en los prototipos. (Cobian, 2013) (Metodología gestión de requerimientos, 2016) (Herrera, 2016) (Junta de andalucia, 2016) (Universidad de Chile, 2016).

Aplicación prototipos no funcionales

Para la construcción de prototipos no funcionales, se utilizó la herramienta de Visio de Office, esta provee un conjunto de controles básicos que pueden ser arrastrados y nombrados dentro de un prototipo. Los prototipos funcionales se encuentran en el anexo 5.3 Prototipos no funcionales

4.2 Diseño

Una de las etapas más importantes dentro de la construcción del software es el diseño de alto nivel, que permite representar la estructura del sistema, sirviendo de comunicación entre las personas involucradas en el desarrollo y ayudando a realizar diversos análisis que orienten el diseño de la aplicación.

La fase de Diseño es una de las fases más importantes en el ciclo de vida del proyecto, ya que esta define la estructura del sistema, recibe como entradas las salidas de la fase de análisis de requisitos, las cuales son:

- Documento de especificación de requisitos.
- Documento de casos de uso
- Prototipos no funcionales (bosquejos) del sistema

Esta fase tiene como salida dos documentos:

- Documento de diseño de alto nivel
- Documento de diseño detallado

Para el análisis y documentación del diseño de alto nivel, se llevan a cabo las siguientes actividades:

- **Definir el tipo de arquitectura que regirá el diseño y la implementación del sistema:**
En esta actividad, se estudian tanto los requisitos funcionales como los no funcionales, para establecer el tipo de arquitectura que más se ajusta al sistema que se va a construir. Existen muchos tipos de arquitectura, y depende del arquitecto de Software definir la más adecuada. Entre los modelos más comunes de arquitectura se encuentra cliente-servidor, arquitectura multicapas, arquitectura orientada a servicios, arquitectura dirigida por eventos, no es necesario escoger una de las anteriores, se pueden definir múltiples arquitecturas compatibles, con el objetivo de satisfacer las necesidades del proyecto de la mejor manera posible.
- **Definir los módulos del sistema:**

En esta actividad, se definen los módulos del Software, las principales características y necesidades de cada módulo, la comunicación de la información entre módulos, las características críticas del software que puedan llegar a afectar el rendimiento o buen comportamiento del sistema. Además se debe definir el orden de implementación de esos módulos, según las necesidades de información que se tengan entre ellos.

- **Definir los lineamientos de la implementación:**

En esta actividad se define el lenguaje que se utilizará para el desarrollo del sistema, también el motor de almacenamiento de datos, las herramientas para el desarrollo de Software y los componentes de Software externos que deben ser utilizados en este

- **Definir los componentes físicos del despliegue:**

En esta actividad se definen los servidores que intervienen en el despliegue de la aplicación, con las características como RAM, procesador, sistema operativo, también se debe definir si se pueden compartir con otras aplicaciones o deben ser exclusivos, se deben definir las principales características de los computadores del cliente, para conocer si se afecta el rendimiento del sistema según el tipo de arquitectura seleccionado. Se debe definir la topología de red que deben tener los servidores, para asegurar el correcto rendimiento de la aplicación no solo por procesamiento de datos sino también por transferencia.

- **Definir interoperabilidad entre otros sistemas:**

En esta actividad se define cuáles de los sistemas existentes puede afectar directamente el sistema que se va a construir, ya sea porque se necesita una fuente de información directa o porque se necesita alimentar otros sistemas, se define además la forma de obtener o

entregar los datos, teniendo en cuenta los componentes del despliegue y el rendimiento del sistema

- **Definir los componentes físicos del sistema:**

En esta actividad se define los componentes del sistema, dependiendo del tipo de arquitectura definida, se pueden construir aplicaciones de escritorio, servicios de Windows, servicios de comunicación de datos, aplicaciones Web y otros, esta definición debe tener en cuenta el tipo de arquitectura, los componentes físicos del despliegue y la interoperabilidad entre otros sistemas.

- **Definir los atributos emergentes:**

En esta actividad se definen los atributos como usabilidad, mantenimiento, seguridad, calidad, también las políticas de recuperación de datos, o de la aplicación, ante eventos que no permitan el funcionamiento del sistema.

- **Construir prototipos funcionales:**

En esta actividad, se deben realizar pruebas técnicas de requisitos complejos, o implementación para validar el uso de componentes externos, con el objetivo de validar el correcto funcionamiento y el cumplimiento de las necesidades documentadas.

Todas estas características que se identifican en las actividades anteriores, determinarán la forma de construcción y aseguraran las propiedades emergentes del sistema, definidas en los requisitos no funcionales.

Para el análisis y documentación del diseño detallado, se llevan a cabo las siguientes actividades:

- **Revisar detalladamente el documento de requisitos**

Esta actividad permite determinar los lineamientos de construcción del sistema, los componentes a tener en cuenta y los prototipos funcionales construidos, todo esto para generar una guía de condiciones técnicas que debe tener en cuenta el desarrollador para una implementación correcta de cada actividad.

- **Identificar las actividades de implementación:**

Esta actividad sirve tanto para la gerencia del proyecto como para la documentación del diseño detallado, aquí se identifican todas las actividades que necesitan ser implementadas y el orden en el que se deben construir, así como también la persona idónea para la construcción, dependiendo de lo anterior, se realiza la guía para la implementación de la actividad

- **Definir y modelar la base de datos**

Esta actividad permite completar el modelo de base de datos, identificando todas las posibles relaciones entre tablas, y reduciendo al mínimo la ambigüedad que se pueda presentar en los documentos anteriores. Aquí se debe definir cada campo y el tipo de dato que se utilizará, además, se deben definir los campos índices según la variabilidad de estos, la necesidad de consulta y las restricciones de claves de negocio únicas

- **Definir y modelar la aplicación**

Esta actividad permite completar el diagrama de clases, identificando las relaciones entre entidades, los métodos y atributos que las componen, al igual que en la actividad anterior, busca reducir al mínimo la ambigüedad que se pueda presentar en los documentos anteriores.

- **Identificar y documentar los patrones de implementación:**

Esta actividad permite dar solución a los problemas comunes en la implementación de software, permitiendo identificar las porciones de código reusable, estandarizar la forma de implementación, facilitar el entendimiento y evitar la documentación e implementación de actividades con el mismo objetivo. Existen varios tipos de patrones que ya están identificados y documentados, entre ellos están los de comportamiento, los de estructura, y los propios del sistema identificados en los requisitos.

- **Documentar el diseño detallado:**

Esta actividad pretende dar lineamientos técnicos a los desarrolladores en la construcción de cada actividad, con el objetivo de asegurar su correcta implementación, ya que tiene en cuenta todas las relaciones y condiciones entre entidades que le aplican, las necesidades de rendimiento de la aplicación, los atributos de calidad que intervienen, las condiciones especiales del despliegue e interoperabilidad, entre otras, todas documentadas en los entregables anteriores.

El documento de diseño detallado, proveerá una guía minuciosa de condiciones técnicas a desarrollar en cada actividad de implementación, que, junto con los documentos de requisitos funcionales y casos de uso, proporcionará todos los insumos al desarrollador para realizar un trabajo de calidad que cumpla con todas las necesidades especificadas, además minimizará errores al lograr determinar las interacciones de la actividad desarrollada con otras actividades, módulos o sistemas externos.

Es muy importante que las personas involucradas en esta fase, sean profesionales con experiencia, e incluso que cuenten con certificaciones relevantes para el diseño de alto nivel, detallado o al lenguaje de programación que se va a utilizar.

Los diagramas UML que se pueden usar en este documento, además de los diagramas descritos en la sección de análisis de requisitos, son los siguientes:

- Diagrama de clases: este diagrama puede representar las entidades de negocio que se construirán en el desarrollo, sus atributos, sus métodos y la relación con otras entidades de negocio
- Diagrama de modelado de datos: este diagrama representa la base de datos que se utilizará para la implementación del sistema, en este se identifican relaciones foráneas, índices de campos que mejoran el rendimiento de consultas, campos que componen cada tabla y restricciones de negocio como campos únicos o combinaciones de campos únicos.
- Diagrama de despliegue: este diagrama muestra los componentes físicos del despliegue de la aplicación, la interacción entre ellos y la topología de red relevante que influya en las decisiones arquitectónicas. (Cervantes, 2010) (Ageraldo, 2005) (Tedeschi, 2016) (Velázquez Cruz, 2012)

Aplicación:

Para la documentación del diseño de alto nivel y el diseño detallado en este proyecto, se utilizó un editor de texto y el diagramador Enterprise Architect, el cual permite crear modelos de datos SQL y modelos de clases, con la opción de generar el código fuente correspondiente. El

documento de diseño de alto nivel se encuentra en el anexo 5.4 Documento de arquitectura de la solución

El documento de diseño detallado se encuentra en el anexo 5.5 Documento de Diseño detallado de la solución

4.3 Implementación

La fase de implementación tiene como objetivo construir el sistema, recibe las siguientes entradas de las fases anteriores:

- Documento de especificación de requisitos.
- Documento de casos de uso
- Prototipos no funcionales (bosquejos) del sistema
- Documento de diseño detallado

En esta fase, el equipo de desarrolladores traduce el diseño detallado a código fuente, y se obtienen los primeros resultados tangibles para el cliente, a través de las siguientes salidas:

- Código fuente
- Instaladores de las aplicaciones implementadas
- Manual de instalación del sistema
- Manual de parametrización del sistema
- Manual de usuario

Como en las fases anteriores, es importante que el personal esté capacitado técnicamente para la codificación, para esto, existen certificaciones y cursos que certifican a las personas en el conocimiento del lenguaje de implementación. Dos de estas certificaciones son la Microsoft Certified Solutions Developer (MCSD) (Microsoft, 2016) y la (Oracle Certified Professional, Java SE 7 Programmer, 2016).

Es muy importante que la empresa asigne el personal idóneo para cada actividad, y también que tenga documentadas y divulgadas las técnicas y lineamientos que rigen el desarrollo de software en la organización, estos lineamientos comúnmente están basados en las buenas prácticas de desarrollo, y permiten una codificación con altos criterios de calidad y estandarizado, lo que facilita el mantenimiento y la escalabilidad del sistema, y da organización a la estructura interna del desarrollo.

Existen varias actividades adicionales al desarrollo que se deben ejecutar en esta fase, para asegurar la calidad del sistema y del código fuente, las actividades son las siguientes:

- **Pruebas de desarrollador:**

En esta actividad, el desarrollador prueba todas las funcionalidades implementadas en cada actividad del desarrollo, asegurando todos los flujos, y determina si las salidas que ocurren al terminar la actividad son las correctas, una vez se ejecuten estas pruebas, se da por completada la actividad.

- **Pruebas de par:**

En esta actividad, un compañero de la misma capacidad técnica o superior, revisa la funcionalidad e implementación de la actividad del desarrollo, para encontrar posibles errores, malas prácticas utilizadas en la codificación, fallas en el uso de lineamientos y posibles problemas inherentes a las propiedades emergentes del sistema.

- **Pruebas técnicas:**

El líder técnico del proyecto o el arquitecto de Software, debe asegurar en cada actividad o conjunto de actividades del desarrollo, el correcto uso de los lineamientos brindados en el diseño detallado de las actividades evaluadas.

Existen herramientas que apoyan el desarrollo de software, y permiten a un desarrollador rendir mejor en sus funciones, generalmente vienen como plug-in del ambiente de desarrollo utilizado, y añaden características o funciones tanto generales como específicas, sobre componentes utilizados en la implementación del sistema, para facilitar su uso y comprensión. (Velázquez Cruz, 2012) (Canarias, 2012) (Plugin Portal, 2016) (Visual studio gallery, 2016)

Aplicación:

Para la construcción del sistema, se utilizó Visual Studio y el lenguaje Visual Basic .NET, adicionalmente el motor de base de datos fue SQL Server. Toda la construcción se basó en el diseño detallado y en la especificación de requisitos.

4.4 Pruebas del sistema

Esta fase consiste en realizar las pruebas al sistema, para certificar que todas las funcionalidades especificadas en el documento de análisis de requisitos están correctamente implementadas, el objetivo es encontrar el mayor número de errores para entregar un sistema confiable al cliente.

Esta fase recibe como entradas los siguientes documentos:

- Documento de especificación de requisitos.
- Documento de casos de uso
- Prototipos no funcionales (Bosquejos) del sistema

Y produce las siguientes salidas:

- Diseño de pruebas funcionales
- Informe de ejecución de las pruebas
- Certificación del producto final

Como en todas las fases anteriores, es muy importante contar con un equipo capacitado técnicamente en ejecución de pruebas, el instituto ISTQB (International software testing qualification board, 2016) otorga diferentes certificaciones en pruebas funcionales y una de estas es Foundation Level, esta enseña un lenguaje internacionalmente reconocido en la industria de testing y certifica a la persona como profesional de pruebas de sistema.

Existen varios tipos de pruebas que se pueden aplicar al sistema en esta fase, estas pueden ser:

- **Prueba de Regresión:**

El objetivo es determinar si hay cambios recientes en una parte de la aplicación y si estos tienen efecto adverso en otras partes.

- **Pruebas de Humo (Smoke Testing o Ad Hoc):**

El objetivo es detectar los errores en los despliegues y estar constantemente probando el sistema para detectar riesgos de baja calidad en este, evitando un sobre esfuerzo en la integración final del sistema.

- **Pruebas de Desempeño:**

El objetivo es validar el tiempo de respuesta para las transacciones o funciones de negocios aplicando las condiciones especificadas de volumen normal de datos y volumen máximo de datos.

- **Pruebas de Carga:**

El objetivo es verificar el tiempo de respuesta del sistema para transacciones que requieran de carga de información.

- **Pruebas de Stress:**

Verificar que el sistema funcione apropiadamente bajo condiciones como baja memoria, máximo de clientes conectados simultáneamente, múltiples usuarios ejecutando una misma acción.

- **Pruebas de Recuperación y Tolerancia a fallas:**

El objetivo de esta prueba es verificar que los procesos de recuperación (manual o automática) restauran apropiadamente la información ante casos de falla, y llevan al sistema a un estado deseado antes de la falla, se ejecutan acciones como interrupciones de energía, de comunicación con el servidor, de acceso a disco en el servidor, de ciclos incompletos (consultas de datos o sincronizaciones interrumpidas).

- **Pruebas de Seguridad y Control de Acceso:**

El objetivo de la prueba es medir el nivel de seguridad de la aplicación verificando que un actor solo pueda acceder a las funciones y datos que su usuario tiene permitido

- **Pruebas de usabilidad:**

El objetivo es verificar La navegación a través de los objetos que reflejan las funcionalidades del negocio y requisitos, se realiza una navegación ventana por ventana, usando los modos de acceso (tabuladores, movimientos del mouse, teclas rápidas, etc), también se verifican los objetos de la ventana y características, tales como menús, medidas, posiciones, estados y focos.

Una de las salidas más importantes de esta fase es el informe de ejecución de pruebas, ya que con este, la empresa puede tomar medidas correctivas en caso de estar fallando mucho en el proceso de ingeniería de Software, generalmente este informe separa por fase los errores encontrados, y el tipo de error encontrado, lo que da información suficiente para determinar las acciones correctivas que se deben implementar. (Pérez Hernández, 2013) (Abad Londoño, 2005)

Aplicación:

El diseño de los casos de prueba del sistema y la documentación de la ejecución, está como trabajo futuro del proyecto.

5 Documentos de ingeniería de software

5.1 Documento de requisitos

5.1.1 Objetivos del sistema

5.1.1.1 Objetivo 1. Apoyo de procesos

Apoyar los procesos de ventas, compras e inventarios de los almacenes, permitiendo el registro y manejo de la información en tiempo real y globalizado.

5.1.1.2 Objetivo 2. Administración logística:

Implantar un sistema integral para la administración óptima de las ventas y créditos ofrecidos en los diferentes almacenes, con la opción de avisos y recordatorios de pagos a los clientes, codeudores y personal del almacén y la distribución de publicidad para aumentar las ventas.

5.1.1.3 Objetivo 3. Integración de la información:

Integrar en línea la operación diaria de los almacenes, tanto en salidas de dinero como en entradas, para conocer de manera efectiva y oportuna los indicadores de gestión y realizar una toma de decisiones acertada.

5.1.1.4 Objetivo 4. Reportes y manejo de la información:

Generar informes basados en la información registrada en el sistema y mantener los datos seguros definiendo políticas de restauración de datos en caso de pérdida o corrupción de estos.

5.1.2 Detalle de las necesidades

En el momento, los almacenes se manejan con papelería manual, es decir, cada semana llega un papel en el cual se especifican los movimientos de la semana, sobre este papel, el contador presenta las cuentas a la DIAN y lleva la contabilidad de Pérdidas y Ganancias. Se necesita un sistema de información, que mejore el manejo actual de la información y brinde la posibilidad de tener informes en tiempo real de la información, además, controlar el flujo de inventario para evitar descuadres o pérdidas de artículos.

El sistema debe ser construido como una solución global para un conjunto de negocios, ya que cada almacén por aparte maneja políticas de ventas diferentes de los otros, además, en caso de abrir otro punto de venta, la aplicación debe ser lo más adaptable posible para disminuir el impacto.

Se han identificado 8 módulos para la construcción del sistema, que apoyarán los procesos de ventas, compras e inventarios, y que se describen a continuación en orden de importancia.

1. **Cuentas por Cobrar (CxC):** Este módulo pretende satisfacer la necesidad de tener un registro actualizado de los clientes que poseen créditos en algún almacén y el estado actual de los créditos, también se necesitan informes de clientes atrasados y morosos en sus obligaciones, y en muchas ocasiones, generar listados de clientes que han dejado de pagar para comenzar con los cobros jurídicos y embargos necesarios.

Por medio de la sistematización de este módulo, se podrán controlar fácilmente los clientes morosos que no cumplen con las obligaciones, dando la posibilidad de tomar acciones legales a tiempo y poder recuperar la inversión.

2. **Facturación:** Este módulo pretende satisfacer la necesidad de tener almacenadas en un sistema todas las ventas de un almacén, tanto para generar informes internos, como para informes de entidades gubernamentales tipo Dian. Además, por medio de este módulo, se podrá conocer el comportamiento de los clientes en los pagos programados, y ayudará a tomar una decisión más acertada al momento de otorgar un crédito al cliente, basándose en la historia de pagos de créditos anteriores.

Por medio de la sistematización de este módulo, se podrá conocer las ventas reales a una fecha sin necesidad de pedir soporte a las personas encargadas de las ventas en cada

almacén, además pretende mitigar la pérdida de tiempo en preparación de informes legales e informes para contabilidad, ya que la persona encargada de este tema, cobra por horas de consultoría al almacén.

3. **Inventario:** Este módulo permitirá controlar de manera efectiva los pedidos de mercancía a la distribuidora, sin necesidad de dejar agotar un producto de consumo masivo y sin tener sobre inventariado un artículo de poco movimiento. Además se podrá controlar el flujo de artículos en algún almacén y evitar robos como ha sucedido en algunas ocasiones. No menos importante, este módulo debe permitir consultar y actualizar el precio de los artículos y la existencia de estos para los otros requisitos, con especial énfasis en los requisitos de Cotizaciones y facturación.

Por medio de la sistematización de este módulo se pretende mitigar el riesgo de robos en los almacenes, y obtener informes de auditoría para conocer el estado real del almacén en un momento, viéndolo desde su inventario y los precios de costo y de venta que tienen los ítems.

4. **Cotizaciones:** Este módulo pretende que los usuarios del sistema o usuarios externos, puedan generar cotizaciones y planes de pago de los productos de interés, una vez sea completada la cotización, la puedan imprimir para ser presentada en el almacén y proceder con la compra y generación del crédito. Con esto se lograría, creando la conciencia del cliente para lograr sus propias cotizaciones, disminuir el tiempo de atención a una persona en el almacén, y aumentar la consulta de precios de artículos y cotizaciones que atraigan a

nuevos clientes que no pueden llegar hasta el almacén de manera continua, sino que ya sabiendo lo que necesitan y cuanto les cuesta, entonces puedan pedirlo.

Por medio de la sistematización de este módulo, se pretende que el almacén tome fuerza en la parte de ventas, ya que cada cliente conoce el precio del artículo que va a comprar y su financiación de la manera como a él le convenga.

5. **Cuentas por Pagar (CxP):** Este módulo pretende reunir todas las obligaciones del almacén hacia otras entidades y que tienen un plazo máximo de pago y un plazo con descuento, organizando de manera que se obtenga el mayor descuento posible para generar ganancias en el manejo de los pagos oportunos.

Por medio de la sistematización de este módulo, se pretende lograr el ahorro de los intereses por mora, o el ahorro por el pronto pago, que muchas veces se deja perder por desconocer estas fechas límites, además, se pretende identificar que cuentas se han pagado, en qué fecha y con qué referencia de pago, ya que ha habido ocasiones en las que el proveedor pierde sus soportes de pago, y el almacén ha tenido que responder por dinero por no conservar estos soportes en ninguna parte.

6. **Balances de P&G:** Este módulo pretende generar balances de Pérdidas y Ganancias para los administradores del negocio, y así tener un detalle del estado del negocio y permitir decisiones interventoras acertadas, que lleven al negocio a un mejor estado. También es

necesario presentar este balance al contador registrado de cada almacén para presentarlo ante autoridades gubernamentales como la Dian.

Por medio de la sistematización de este módulo, se pretende lograr una mejor visión de la empresa en un período contable, identificando las ganancias netas y las utilidades para establecer metas de desempeño a corto plazo. También pretende ayudar a generar los informes de la Dian de Discos Magnéticos.

7. **Codeudores:** Este módulo pretende organizar los codeudores de un cliente que solicita un crédito, manejando la información de contacto y bienes raíces registrados al momento de la solicitud, para ser entregada a un abogado cobrador en caso de fallar en el pago del crédito. También mantener a los codeudores informados acerca del estado del crédito en el momento de consulta o por reportes periódicos si así lo prefieren.

Por medio de la sistematización de este módulo, se pretende lograr un mejor manejo de los codeudores y aumentar su responsabilidad en un crédito, así como también incentivar a estos codeudores a que presionen al responsable del pago para que ellos no sufran las consecuencias y así disminuir en gran manera las pérdidas del almacén.

8. **Publicidad comercial:** Este módulo pretende que el cliente esté enterado de las últimas novedades en promociones o en artículos nuevos, para generar cultura de compras a los almacenes, esta publicidad está enfocada a la necesidad del cliente y se debe mantener actualizada.

Por medio de la sistematización de este módulo, se pretende enviar notificaciones a los correos de los clientes para que aprovechen las promociones vigentes en el almacén y así aumentar el flujo de ventas.

El sistema debe ser una solución en línea, con acceso desde cualquier ubicación a los diferentes usuarios, permitiendo el flujo de información en el momento que se necesita y consultada en tiempo real, debe estar respaldada por una conexión confiable y un hosting que preste servicio inmediato ante un fallo, ya que la aplicación, a medida que se utilice en los almacenes, adquirirá mayor necesidad y mayor manejo.

El sistema debe ser confiable en su información almacenada, y a su vez mantener respaldada toda la información al día, en caso de que ocurra cualquier incidente poder acceder al respaldo de la información y no perder todos los datos.

5.1.3 Análisis de los requisitos

5.1.3.1 Cuentas por cobrar (CxC)

Las cuentas por cobrar son los créditos que se les otorgan a los clientes, en el momento se manejan créditos con una cuota inicial entre el 40% y 50% del total financiado y con máximo un plazo de 15 meses para el pago, solo en clientes de confianza este plazo se puede aumentar a 18 meses.

En la actualidad, se tiene una hoja en Excel donde se registran las cuotas programadas para el pago de un crédito de un cliente, teniendo que buscar la información de manera manual para generar los listados de clientes morosos y también para registrar los pagos de las obligaciones de cada cliente, no hay ningún control por parte del almacén para el aviso a los clientes del vencimiento de las cuotas y no hay tampoco un control de cuotas vencidas de los clientes, por lo que en repetidas ocasiones no se han hecho pagos totales de los créditos y no se ha podido tomar ninguna acción por el hecho de no contar con una acción preventiva a tiempo.

Lo principal que se tiene que tener en cuenta en este requisito es el control de los créditos y de las cuotas de un cliente, para proporcionar al almacén una herramienta de control de CxC efectiva y que sea de gran ayuda por medio de consultas e informes.

A continuación se listan los principales requisitos:

- **Control de codeudores**

El sistema deberá registrar los codeudores de los clientes a cada crédito, estos serán los encargados de responder por la obligación en caso de que el cliente la incumpla. El sistema deberá requerir mínimo 2 codeudores para cada crédito, sin estos, no se podrá otorgar el crédito.

- **Control de créditos**

El sistema deberá identificar los créditos con un número único consecutivo y controlar los créditos de los clientes, ingresando créditos nuevos y modificando los existentes.

El sistema deberá controlar que la única modificación permitida sea la del estado del crédito, este estado puede ser:

- Pagado
- En curso
- Cancelado

El sistema deberá permitir cancelar un crédito, en caso de tener que recurrir a cobro jurídico y decomiso del artículo comprado.

El sistema deberá registrar una única cuota inicial, registrando el valor de esta al momento de la generación del crédito.

- **Control de cuotas del crédito**

El sistema deberá controlar las cuotas de pago programadas, estableciendo una fecha de vencimiento y un valor, al momento del cliente solicitar un crédito, también deberá crear el plan de pagos, y cada pago o abono realizado se registrará como cumplimiento del plan estipulado de pago.

Para mayor flexibilidad el sistema deberá definir un parámetro de control para el tope máximo de cuotas permitidas, pero el administrador podrá seleccionar N cuotas sin restricciones de parámetros. Cuando la última cuota sea pagada, EL sistema deberá pasar el estado del "crédito" a "pagado".

- **Listado de clientes con cuotas atrasadas**

El sistema deberá proveer un listado de los clientes morosos, para decidir si son necesarias acciones preventivas contra el no pago del crédito, o solo con una llamada de atención al cliente se puede obtener el pronto pago.

El sistema deberá mostrar este listado con una ayuda visual por código de colores, con la siguiente notación:

- Negro: Son los clientes que poseen 1 sola cuota atrasada
- Naranja: Son los clientes que poseen 2 cuotas de atraso
- Rojo: Son los clientes que poseen 3 o más cuotas atrasadas

El sistema deberá permitir exportar este listado a Excel, registrando la información detallada del crédito y la información relevante del cliente y la de sus codeudores.

- **Detalle del crédito de un cliente**

El sistema deberá permitir la visualización del detalle del crédito, este se refiere a cuanto el cliente ha avanzado en el pago de la obligación, detallando los valores de intereses, capital y pagos por mora de cada cuota pagada, también detallando las cuotas pendientes, las fechas de vencimiento de cada cuota y el total del crédito.

- **Manejo de cobro de mora por retraso en el pago**

El sistema deberá reprogramar el valor de pago de una cuota, generando un cobro adicional al valor de esta por cobro de mora.

El sistema deberá manejar el cobro por mora como un porcentaje, que se definirá en los parámetros y podrá ser modificable por el administrador.

- **Manejo de pagos a cuotas del crédito**

El sistema deberá registrar los pagos de cuotas a un crédito de un cliente, mostrando el valor a pagar y en caso de existir, el cobro de mora por retraso en el pago.

El sistema deberá generar un recibo de pago para el cliente, identificando el número de la cuota pagada y el valor total de esta.

El sistema deberá permitir realizar pagos de diferente valor al de la cuota establecida para el crédito, estos pagos se controlaran de 3 maneras

- Inferior al valor de cuota, en este caso, el sistema deberá registrar el valor como un pago parcial de una cuota, que luego será cancelada en su totalidad..
- Superior al valor de una o más cuotas, EL sistema deberá establecer como pagadas las cuotas que cubra el valor del pago, y en caso de que exista diferencia de valores con, registrar la diferencia como un pago parcial de la próxima cuota.

El sistema deberá controlar que el valor del pago de diferente valor al de la cuota, no sea superior al valor total del crédito.

- **Envío de recordatorios de pago a los clientes y codeudores, 2 días antes del vencimiento de una cuota (Debe ser opcional para el cliente y los codeudores)**

El sistema deberá generar avisos automáticos de recordatorio de pago de cuotas de un crédito, para mantener informado al cliente y los codeudores.

El sistema deberá manejar un parámetro que permita decidir si envía o no correos con esta información a cada cliente y a sus codeudores registrados.

- **Aviso de cuotas vencidas**

El sistema deberá generar un aviso vía mail al empleado de las cuotas que se vencieron el día anterior y no fueron canceladas, con el fin de que el empleado llame a cada cliente a recordarle el pago de la deuda.

- **Aviso a codeudores.**

El sistema deberá generar un aviso vía mail a los codeudores del cliente, explicando el estado del crédito del amparado y el procedimiento para el cobro jurídico, con el fin de que lleguen a un acuerdo de pago y no se tenga que proceder con el cobro jurídico.

5.1.3.2 *Facturación*

La facturación es el registro de artículos comprados al almacén por parte de un cliente, en cada factura, se debe especificar en detalle los artículos comprados y el cliente al que se le vendieron esos artículos.

En la actualidad, la facturación se maneja de manera manual, al cliente se le entrega una factura de talonario marcada y numerada, con el detalle de la compra realizada.

Esta factura es hecha a mano y no se tiene un registro sistematizado para el manejo de estas.

Lo principal de este requisito es sistematizar el proceso de facturación para que se puedan escoger los ítems que un cliente desee comprar y el sistema genere la factura de manera automática, manteniendo un registro de ventas completo y un histórico de compras para determinar tendencias del mercado.

Los principales requisitos son:

- **Gestionar facturas**

El sistema deberá permitir la creación de facturas a partir de la selección de ítems de inventario a comprar por parte del cliente, estas facturas poseen un número único de identificación asignado por el sistema en base a la ley.

Cuando una factura no tenga validez, ya sea por error del empleado o por cambio en la decisión del cliente, el sistema deberá permitir la anulación de la factura, y deberá exigir al empleado el registro de un motivo por el cual se anula. Además, el sistema deberá adicionar al inventario los ítems antes descontados por la generación de la factura.

Una factura puede ser pagada de inmediato (En efectivo) o se puede generar un crédito por su valor.

- **Manejo de rangos de facturación con fechas de vencimiento según DIAN.**

El sistema deberá manejar los rangos de facturas asignados por la Dian, entre los cuales se numeran todas las facturas. En caso de cambiar los rangos de facturación, el sistema deberá aplicar los cambios de manera inmediata y facturar bajo los nuevos parámetros.

- **Descontar de inventario los ítems seleccionados**

El sistema deberá descontar de inventario la cantidad de ítems seleccionados en cada factura, para lograr un control efectivo del inventario.

- **Permitir crear un crédito sobre el valor de una factura**

El sistema deberá mostrar la información del crédito cargando como valor a financiar el valor de la factura, y permitir cotizar diferentes maneras de pago. Cuando el empleado termine la cotización, el sistema deberá crear tanto el crédito como la factura de la compra.

- **Imprimir recibos de facturas con detalles de artículos para los clientes**

Cada vez que se genere una factura, el sistema deberá generar un registro impreso para el cliente, con el cual se presentaran los reclamos en caso de existir y contendrá información del detalle de los ítems comprados y su valor, el total del pago y como información adicional el usuario y clave para ingreso a la aplicación en caso de poseer un crédito.

- **Generar listados de facturas en un rango de fechas**

El sistema deberá generar listados de facturas creadas en un rango de fechas, con el fin de presentar informes y tener un control de ventas más preciso.

- **Permitir generar descuentos en los ítems de una factura.**

En varias ocasiones se obsequian algunos ítems de inventario para aquellos clientes que compren un artículo en especial, por ende, el sistema deberá permitir descuentos hasta del 100% sobre el valor de un ítem, ya que debe estar registrado en la factura y debe ser retirado del inventario, pero no debe ser cargado al costo del cliente.

- **Separar IVA del valor de la factura, según el IVA de cada artículo**

El sistema deberá separar el IVA del valor total de la factura, por motivos de presentación de informes en la Dian, este valor del IVA es diferente en cada artículo. Esta información no tiene que ser vista por el cliente en la factura impresa, pero el sistema deberá manejar un registro de manera interna por cada factura y por cada ítem en una factura.

- **Permitir adicionar otros cargos a la factura**

El sistema deberá permitir la posibilidad de adicionar otros cargos a la factura. Estos cargos serán definidos como servicios en la aplicación y tendrán un costo definido, pero no tendrán un inventario ya que son servicios prestados por terceros al almacén, entre los servicios se pueden encontrar:

- Matrícula del vehículo
- SOAT del vehículo
- Marcado del vehículo (Imprenta de las placas en diferentes partes)
- Marcado de cascos
- Otros...

5.1.3.3 Inventario

El inventario es el control de artículos a la venta en el almacén, cada artículo posee un valor propio y una cantidad de existencias, un valor de compra y un valor de venta y un porcentaje de IVA.

En la actualidad, el inventario se controla por parte del empleado, este tiene que tener en cuenta cuantos artículos tiene y cuando se hace necesario pedir más para abastecer el inventario, además, el empleado debe registrar la diferencia entre el valor de compra de cada artículo y el valor de venta de este, de manera manual.

Este requisito pretende optimizar el manejo del inventario en el almacén, sistematizando y controlando todo el proceso y liberar de tareas tediosas al empleado, reduciéndole las responsabilidades a la consulta de stock de artículos y el registro de entrada de estos, ya que el sistema se encargaría del resto.

Los principales requisitos son:

- **Controlar inventario**

El sistema deberá permitir el ingreso y modificación de ítems ofrecidos en el almacén, estos ítems tienen un valor de venta y un porcentaje de IVA exigido por la Dian, cada ítem será identificado por un código único dentro del sistema, que corresponde al código de barras del artículo.

El sistema deberá además, permitir el ingreso de 1 o más imágenes asociadas a cada artículo.

- **Definir propiedades de cada ítem**

El sistema deberá permitir la creación de campos definidos por el empleado que brinden propiedades sobre el ítem, estos campos deben estar acompañados de un valor asociado para cada ítem.

El sistema deberá mostrar estos campos con sus valores respectivos al momento de generar una cotización.

- **Manejar stock de artículos**

El sistema deberá controlar la existencia de cada artículo, para avisar en el momento de la generación de la factura si un artículo se encuentra agotado, y así el empleado saber que no puede registrar ese ítem,

- **Registrar lotes de entrada de artículo**

Un lote de entrada es un abastecimiento de inventario, se registra el valor de compra del ítem en ese lote y la cantidad de ítems ingresados a cada artículo y se mantiene una cantidad existente actualizada.

El sistema deberá permitir el manejo de estos lotes registrando entradas de ítems al inventario existente.

- **Registrar salida de artículos, con número de factura**

Cada vez que sale un ítem de inventario, el sistema deberá descontar el ítem del lote más antiguo con existencia mayor a 0 (cero), en caso de no existir ningún lote con

existencias mayores que 0 (cero), quiere decir que no hay existencia del artículo en el almacén.

- **Listar artículos existentes en el almacén**

El sistema deberá generar un informe de todos los artículos manejados en el almacén, y su cantidad de ítems en el momento. El informe se debe poder filtrar por artículo y por cantidades superiores a (cero).

- **Listar en un rango de fecha, lotes de entrada de un artículo**

El sistema deberá permitir la consulta de los lotes de entrada de un artículo al sistema, mostrando su valor de compra, la fecha de ingreso al sistema y la cantidad de ítems ingresados.

5.1.3.4 Cotizaciones

Las cotizaciones son una herramienta tanto para el empleado como para los clientes, por medio de esta pueden averiguar el precio de un artículo sin necesidad de desplazarse hasta el almacén o sin gastar el tiempo de una llamada telefónica, además, pueden crear planes de pago de créditos sin ningún compromiso y de manera gratuita.

En el momento estas cotizaciones se tienen al manejo del empleado del almacén, y se realizan de manera manual según la cuota inicial y el número de cuotas que el cliente desee, o el valor de estas

La ventaja que ofrecerá este requisito, es que los clientes puedan acceder a crear sus cotizaciones y planes de pago más convenientes, para así liberar carga al empleado y ofrecer un mejor servicio al cliente, como no tiene ningún compromiso ni costo, se espera que por medio de publicidad la clientela aumente y por ende las ventas.

Los principales requisitos son:

- **Ofrecer una herramienta libre de uso a los clientes.**

EL sistema deberá permitir generar cotizaciones a los usuarios no registrados en la aplicación, permitiéndoles ver el total de una compra y la posibilidad de generar un plan de pago acomodado sin ninguna intervención del empleado del almacén.

- **Permitir al empleado pasar de una cotización a una venta con factura**

Luego de que el empleado de la información al cliente y este consienta la compra, el sistema deberá tomar la información de la cotización y llevarla a una factura para ser completada la venta.

- **Permitir al cliente imprimir su cotización para tenerla como constancia.**

El sistema deberá permitir imprimir la cotización generada a su medida, para tenerla como constancia y como apoyo al empleado al momento de generar la compra en presencia del cliente.

- **Permitir añadir servicios al valor de la cotización**

El sistema deberá permitir añadir servicios adicionales a la cotización para sumarlos al valor del crédito, permitiéndole al cliente generar la cotización por el valor exacto de su compra y no por un valor menor.

5.1.3.5 Cuentas x Pagar (CxP)

Las cuentas por pagar son deudas en las que incurren el almacén por la adquisición de productos, estas cuentas, por lo general, vienen con varias fechas de pago, cada fecha corresponde a un descuento por pronto pago. Una factura por pagar puede tener hasta 3 fechas de pago con diferentes porcentajes de descuento, y su buen manejo, puede dejar al almacén con mayores ganancias, o puede en algún momento dejar el almacén sin fluidez para su funcionamiento, por eso, es necesario una vista de las obligaciones de pago del almacén, con las fechas de pago máximas registradas para determinar si se puede o no optar por el pronto pago para generar mayores utilidades.

En la actualidad, las facturas llegan de manera impresa al almacén y se guardan hasta que llega la fecha de pago, en pocas ocasiones se accede al descuento que otorga la entidad por el pronto pago, además, se ha dado el caso que se tiene que pagar dineros por la pérdida del registro de facturación del proveedor, generando descuadres notables en la contabilidad del almacén.

Por medio de este requisito, se pretende lograr un registro completo de los pagos realizados a los proveedores, para cuando ocurran descuadres en su contabilidad, el sistema pueda minimizar y en el mejor de los casos corroborar contra el proveedor, el pago de todas las facturas que se generan en la operación diaria. También se pretende lograr obtener descuentos importantes en el pago temprano de las facturas, con alertas del sistema por vencimiento de estas.

Los principales requisitos son:

- **Registrar factura:** El sistema deberá permitir el registro de una factura y el proveedor al que le corresponde, en la factura se debe especificar el precio y fecha de pago final, además el usuario debe tener la posibilidad de ingresar otras 2 fechas de vencimiento, con un descuento otorgado por pronto pago.
- **Alertar el vencimiento de fechas de pago:** El sistema deberá enviar notificaciones por correo electrónico, alertando a los usuarios del sistema el vencimiento de una factura o de un plazo de pago con descuento, el correo debe ser enviado el día anterior al vencimiento.

- **Registrar pagos de factura:** El sistema deberá permitir el registro del pago a una factura ingresada en el sistema, con el valor pagado y la fecha de pago.
- **Listar las facturas próximas a vencer:** El sistema deberá generar un listado de las facturas que están prontas a vencer o que un plazo con descuento está por cumplirse, el informe debe mostrar con una identificación de colores el plazo que esta por vencer, el proveedor, el descuento si tiene y el valor a pagar. Los colores que se utilizaran será: Verde para el primer plazo, Amarillo para el segundo y rojo para el plazo final.
- **Listar facturas pagadas:** El sistema deberá generar un listado de las facturas pagadas a un proveedor en un rango de fechas, con el valor correspondiente de pago, la fecha de pago, las fechas de vencimiento y la fecha final de pago.

5.1.3.6 Sistema

A través de este análisis, se han identificado funcionalidades del sistema que permitirán el fácil manejo, configuración y adaptación al negocio de todos los módulos.

Las funcionalidades identificadas son:

- **Administrar clientes:** El sistema deberá permitir un control de clientes, procurando la actualización constante de estos con la información más reciente, en este control, se podrán registrar o actualizar clientes cuando soliciten un crédito o cancelen una factura en efectivo

(por motivos de leyes), o desde la administración de clientes, también se debe permitir la actualización de datos del cliente en línea y activar o desactivar las opciones de envío de correos a este. La búsqueda de los clientes registrados se realizará por número de cedula del cliente, en caso de no existir se debe crear obligatoriamente el registro de la persona y en caso de que exista, se debe dar la opción de actualizar los datos.

- **Administrar proveedores:** El sistema deberá permitir un control de proveedores, en donde se puedan registrar proveedores y estos se puedan actualizar o inactivar en el sistema, también se debe almacenar la información de uno o varios contactos del proveedor, el área a la que corresponde, el mail y el teléfono de contacto.
- **Administrar categorías:** El sistema deberá permitir ingresar, modificar o eliminar categorías para los ítems de los inventarios, estas categorías permiten identificar si un ítem es un repuesto, un electrodoméstico, un accesorio, una moto, entre otras.
- **Administrar marcas:** El sistema deberá permitir ingresar, modificar o eliminar marcas de los ítems de inventario, estas marcas son las compañías que producen los diferentes ítems registrados en el sistema.
- **Modularización:** Esta funcionalidad permite crear módulos, en el momento se ha identificado como modulo cada requisito mencionado, por medio de esta funcionalidad, se controlaran los permisos de acceso a los diferentes usuarios del sistema.

- **Usuarios y Roles:** Cada usuario en el sistema contará con un rol (perfil), que le otorgará acceso a las diferentes opciones del sistema. Los roles identificados hasta el momento son:
 - **Cliente**
 - **Empleado**
 - **Contador**
 - **Administrador**

- **Configuración dinámica:** La configuración de la aplicación debe estar cargada en la Base de Datos, para que los cambios tomen efecto de manera inmediata y no sea necesario reiniciar la aplicación con cada cambio en los parámetros de funcionamiento.

5.1.4 Actores

Tabla 1. Actores

Rol	Principal participación
Empleado del Almacén	Encargado de manejar la operación diaria del almacén como generar facturas, estudiar créditos, cobrar cuotas de créditos vigentes, entre otras funciones
Administrador del almacén	Encargado de apoyar la operación del almacén, tomando decisiones de pedidos, manejando los balances de pérdidas y ganancias, y desembolsando el dinero para pagar las facturas de los proveedores.
Administrador del sistema	Encargado de la configuración y funcionamiento del sistema, entre las operaciones se encuentra la generación de consultas dinámicas, la configuración de parámetros y las copias de seguridad.

Fuente: Elaboración propia.

5.1.5 Requisitos Funcionales

5.1.5.1 Modulo: Administrativo

5.1.5.1.1 REQ 1. Administrar clientes.

El sistema deberá permitir al empleado registrar clientes que soliciten un crédito o cancelen una factura en efectivo (por motivos de leyes), también se debe permitir la actualización de datos del cliente y activar o desactivar las opciones de envío de correos publicitarios o con la información de créditos a este.

La búsqueda de los clientes registrados se realizará por número de cedula del cliente, en caso de no existir se debe crear obligatoriamente el registro de la persona y en caso de que exista, se debe dar la opción de actualizar los datos.

5.1.5.1.2 REQ 2. Administrar proveedores.

El sistema deberá permitir registrar y actualizar la información de proveedores, también se debe almacenar la información de uno o varios contactos del proveedor, el área a la que corresponde, el mail y el teléfono de contacto.

5.1.5.1.3 REQ 3. Administrar categorías.

El sistema deberá permitir al empleado ingresar, modificar o eliminar categorías para los ítems de los inventarios, estas categorías permiten identificar si un ítem es un repuesto, un electrodoméstico, un accesorio, una moto, entre otras.

5.1.5.1.4 REQ 4. Administrar marcas.

El sistema deberá permitir al empleado ingresar, modificar o eliminar marcas de los ítems de inventario, estas marcas son las compañías que producen los diferentes ítems registrados en el sistema.

5.1.5.1.5 REQ 5. Administrar perfiles.

El sistema deberá permitir al administrado del sistema ingresar, modificar o eliminar perfiles de usuarios, estos perfiles se podrán configurar para otorgar o denegar permisos a las opciones de la aplicación.

5.1.5.1.6 REQ 6. Administrar usuarios.

El sistema deberá permitir al administrador del sistema ingresar y modificar usuarios, a cada usuario se le debe asignar un perfil que definirá los permisos sobre las acciones de la aplicación.

5.1.5.1.7 REQ 7. Administrar parámetros

El sistema deberá permitir al administrador del sistema modificar parámetros de configuración, los parámetros no se pueden ni crear ni eliminar, solo modificar los valores.

5.1.5.1.8 REQ 8. Log de la aplicación:

El sistema deberá registrar las acciones realizadas sobre el sistema almacenando fecha, hora, acción y usuario.

5.1.5.1.9 REQ 9. Consultas dinámicas:

El sistema deberá permitir al administrador del sistema crear y realizar consultas en lenguaje SQL exportables a MS Excel, TXT o visualizables en Web

5.1.5.2 *Modulo: Cuentas por cobrar*

5.1.5.2.1 REQ 10. Administrar codeudores.

El sistema deberá permitir al empleado crear o modificar la información de los codeudores, un codeudor se ingresa como un cliente en su información básica, pero se debe completar con los documentos exigidos por la empresa (Carta Laboral, certificado de vivienda y otros que el codeudor presente voluntariamente).

5.1.5.2.2 REQ 11. Asociar codeudores.

El sistema deberá permitir al empleado asociar los codeudores de los clientes a cada crédito, estos serán los encargados de responder por la obligación en caso de que el cliente la incumpla.

5.1.5.2.3 REQ 12 Administrar créditos.

El sistema deberá permitir al empleado ingresar créditos nuevos, el crédito que se ingrese quedará en estado “En Curso” y se deberá identificar con un número único consecutivo

5.1.5.2.4 REQ 13 Cancelar créditos.

El sistema deberá permitir al empleado cancelar un crédito, en caso de tener que recurrir a cobro jurídico y decomiso del artículo comprado, esta acción generará una notificación al administrador del almacén en la cual informará la cancelación del crédito y la justificación de esta.

5.1.5.2.5 REQ 14 Calcular cuota inicial de créditos.

El sistema deberá calcular el valor de la cuota inicial, según el parámetro “*Porcentaje valor cuota inicial*” y el valor de la factura, de la siguiente manera:

Ecuación 1. Calcular cuota inicial de créditos.

$$CI = Valor\ factura * Porcentaje\ valor\ cuota\ inicial$$

Fuente: Elaboración propia

5.1.5.2.6 REQ 15 Crear cuotas del crédito.

El sistema deberá crear las cuotas de pago programadas, creando el número de cuotas seleccionado (no mayor al número de cuotas establecida en el parámetro “*Número de cuotas permitidas en créditos*”) con el total a cancelar en cada cuota y la fecha de vencimiento.

5.1.5.2.7 REQ 16 Pagar créditos.

El sistema deberá actualizar el estado del crédito a Pagado cuando la última cuota sea pagada.

5.1.5.2.8 REQ 17. Notificar codeudores de cuotas vencidas

El sistema deberá notificar a los codeudores, por medio de un correo electrónico, las cuotas vencidas que tiene un crédito al cual se encuentren asociados, con el aviso legal del cobro, para que estén enterados de las consecuencias si este no se cumple.

5.1.5.2.9 REQ 18. Notificar al empleado de cuotas vencidas.

El sistema deberá enviar un correo electrónico al empleado con las cuotas que se encuentren vencidas a la fecha, con el fin de que el empleado llame a cada cliente a recordarle el pago.

5.1.5.3 *Modulo: Facturación.*

5.1.5.3.1 REQ 19. Crear facturas.

El sistema deberá permitir al empleado crear una factura luego de la selección de los ítems que la componen, cada vez que se cree una factura, el sistema deberá genera un numero único de identificación a partir de un rango de facturación vigente y un consecutivo de facturación.

5.1.5.3.2 REQ 20. Generar número de factura.

El sistema deberá generar un número único de identificación tomando el rango de facturación vigente y sumándole con un consecutivo de facturación que aumentará con cada factura generada.

5.1.5.3.3 REQ 21. Anular facturas.

El sistema deberá permitir al empleado anular facturas que por algún motivo no tengan validez, exigiendo el motivo de la anulación, luego de la anulación, el sistema deberá a descargar los ítems de la factura y volverlos a cargar al inventario.

5.1.5.3.4 REQ 22. Pagar facturas.

El sistema deberá permitir al empleado del almacén registrar el pago de la factura, el tipo de pago puede ser en efectivo o crédito, en caso de ser crédito, el sistema deberá presentar la opción de crear un crédito pidiendo número de cuotas, cuota inicial y codeudores. Cuando el empleado ingrese toda la información del crédito, el sistema deberá presentar la opción para crear el plan de pago de este.

5.1.5.3.5 REQ 23. Administrar rangos de facturación.

El sistema deberá permitir al administrador del almacén, ingresar los rangos de facturación otorgados por la DIAN, con la fecha de vencimiento, el valor inicial del rango y el valor final del rango.

5.1.5.3.6 REQ 24. Imprimir Recibos de facturas.

El sistema deberá permitir al empleado del almacén, imprimir la factura para ser entregada al cliente, con los valores de los ítems seleccionados, el total y el IVA de la factura.

5.1.5.3.7 REQ 25. Ingresar descuentos en ítems

El sistema deberá permitir al empleado del almacén registrar descuentos sobre los ítems de una factura, este descuento se ingresará en porcentaje y modificará el valor de la factura

5.1.5.3.8 REQ 26. Calcular IVA por artículo

El sistema deberá calcular el IVA total de la factura según el porcentaje de IVA de cada artículo, sumando todos los IVA de los ítems seleccionados en la factura.

5.1.5.3.9 REQ 27. Descontar Ítems de inventario

El sistema deberá descontar del inventario los ítems registrados a una factura

5.1.5.3.10 REQ 28. Agregar Ítems al inventario

El sistema deberá sumar al inventario los ítems registrados a una factura cuando esta sea anulada

5.1.5.4 *Modulo: Inventario*

5.1.5.4.1 REQ 29. Ingresar Ítems de inventario

El sistema deberá permitir al empleado la creacion de artículos de inventario, el sistema pide el artículo, la cantidad, la cantidad para notificar “out of stock”, el costo real del artículo, el valor comercial, el porcentaje de IVA exigido por la DIAN, el valor neto que se calcula con el valor total menos el porcentaje de IVA y una o más imágenes para identificar el artículo

5.1.5.4.2 REQ 30. Modificar Inventario

El sistema deberá permitir al empleado la modificación de la información de un artículo, si la modificación corresponde a la cantidad de existencia, entonces el sistema deberá solicitar una justificación y notificar al administrador de este evento con la justificación, el valor anterior del artículo y la cantidad nueva de esta

5.1.5.4.3 REQ 31. Ingresar o eliminar propiedades

El sistema deberá permitir al empleado del almacén ingresar o eliminar propiedades que distingan a un artículo en particular, estas propiedades serán utilizadas al momento de generar una cotización en línea en la que se consulta el detalle de un artículo.

5.1.5.4.4 REQ 32. Notificar “Out of Stock”

El sistema deberá generar una notificación automática cuando detecte que un artículo esté en el límite o por debajo de este, esta notificación debe ser colocada en la pantalla principal del sistema y además enviarla por mail al administrador para estar enterado.

5.1.5.4.5 REQ 33. Ingresar lotes de inventario

El sistema deberá permitir al empleado registrar un lote con la cantidad de artículos y el valor de compra de cada uno, el sistema debe sumar la cantidad del lote con la cantidad existente actual del artículo, y calcular el costo real con una suma promediada con la siguiente formula:

$$\frac{(\text{Costo Articulo antiguo} * \text{Cantidad Articulo Antiguo} + \text{Costo Articulo Nuevo} * \text{Cantidad Artículos Nuevos})}{\text{cantidad de artículos antiguos} + \text{Cantidad de artículos Nuevos}}$$

5.1.5.4.6 REQ 34. Listar artículos existentes en el almacén

El sistema deberá permitir al empleado listar todos los artículos manejados en el almacén, y su cantidad de ítems en el momento. El informe se debe poder filtrar por artículo y por cantidades superiores a cero.

5.1.5.4.7 REQ 35. Listar en un rango de fecha, lotes de entrada de un artículo

El sistema deberá permitir al empleado consultar los lotes de entrada de un artículo al sistema, mostrando su valor de compra, la fecha de ingreso al sistema y la cantidad de ítems ingresados.

5.1.5.5 Modulo: Cotizaciones

5.1.5.5.1 REQ 36. Generar Cotizaciones

El sistema deberá permitir a un usuario sin autenticación realizar cotizaciones permitiéndole seleccionar uno o más artículos y ver las imágenes y las propiedades de estos, además el sistema debe permitir al usuario imprimir o exportar a PDF la cotización realizada, además el usuario podrá adicionar uno o más servicios prestados por el almacén y sumarlos al valor de la cotización.

5.1.5.5.2 REQ 37. Facturar Cotización

El sistema deberá permitir al empleado realizar una cotización y luego de esto, el empleado tendrá la opción de generar una factura con los ítems y servicios seleccionados. Luego de seleccionar la opción de facturar, se realiza el *REQ 19. Crear Factura*

5.1.5.6 *Modulo: Cuentas por Pagar*

5.1.5.6.1 REQ 38. Registrar Factura

El sistema deberá permitir al empleado registrar las facturas de cobro al almacén, ingresando la información de precio, fecha máxima de pago, máximo 2 fechas de pago anticipado con su porcentaje de descuento correspondiente y el proveedor que genera la factura

5.1.5.6.2 REQ 39. Notificar vencimiento

El sistema deberá generar una notificación a los empleados y al administrador del almacén de los próximos vencimientos de las facturas por pagar, la notificación se deberá enviar según el parámetro “Días de anticipación para notificar facturas por pagar” y con la información del valor a pagar, si la fecha de vencimiento tiene asociado un descuento, entonces se debe notificar la fecha de pago, la fecha máxima de pago, el porcentaje de descuento y el valor aplicando el porcentaje de descuento.

5.1.5.6.3 REQ 40. Registrar pago de factura

El sistema deberá permitir al empleado o el administrador del sistema ingresar el soporte de pago de una factura, esta acción desactivara las alertas que la misma factura pueda generar si se pagó con anticipación.

5.1.5.6.4 REQ 41. Apagar alertas de facturas

El sistema deberá permitir al administrador del almacén apagar notificaciones de alertas que no se hayan registrado en el sistema, pero que no se deban alertar por alguna razón, como la acción es realizada por el administrador del almacén no se requiere justificación, pero esta se muestra opcional por si se desea entrar un texto.

5.1.5.6.5 REQ 42. Listar las facturas próximas a vencer

El sistema deberá permitir al empleado generar un listado de las facturas que están prontas a vencer o que un plazo con descuento está por cumplirse, el informe debe mostrar con una identificación de colores el plazo que esta por vencer, el proveedor, el descuento si tiene y el valor a pagar. Los colores que se utilizaran será: Verde para el primer plazo, Amarillo para el segundo y rojo para el plazo final.

5.1.5.6.6 REQ 43. Listar facturas pagadas

El sistema deberá permitir al empleado generar un listado de las facturas pagadas a un proveedor en un rango de fechas, con el valor de la factura y el valor que se pagó, la fecha de pago, la fecha en la que se pagó y la fecha final de pago, adicionalmente el sistema permitirá al empleado descargar el archivo de soporte de pago de cada factura.

5.1.6 Requisitos No Funcionales

5.1.6.1 Disponibilidad

5.1.6.1.1 RNF 1. Versiones futuras.

Es altamente probable que el sistema requiera de futuras versiones y actualizaciones.

5.1.6.2 Escalabilidad

5.1.6.2.1 RNF 2. Incremento en usuarios y transacciones.

El sistema deberá soportar un incremento proporcional a la existente en los 2 almacenes en donde se realizará el piloto de esta aplicación.

5.1.6.2.2 RNF 3. Navegadores.

El sistema deberá operar en Internet Explorer versión 8 y 9, Mozilla Firefox versión 13 y 14.

5.1.6.3 *Seguridad*

5.1.6.3.1 RNF 4. Auditorías.

El sistema es altamente vulnerable a saboteos, debido a que manejará información crediticia de clientes. El sistema deberá registrar el usuario que efectúa cada acción en el sistema.

5.1.6.4 *Técnicos*

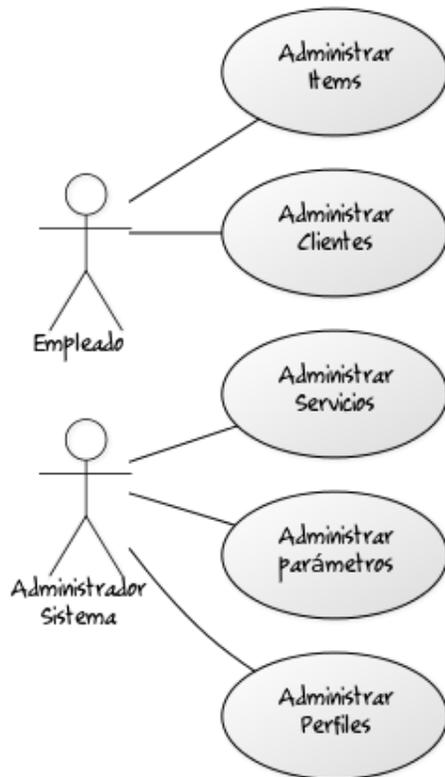
5.1.6.4.1 RNF 5. Base de datos.

El sistema deberá acceder a una base de datos de SQL SERVER 2012

5.2 Documento de Casos de uso

5.2.1 Administración

Diagrama 2 Casos de uso – Administración



Fuente: Elaboración propia con base en la herramienta YUML (Yuml, 2016).

5.2.1.1 Administrar clientes

Tabla 2. Administrar clientes

Descripción:	Por medio de este, los actores pueden crear, eliminar, modificar y consultar los clientes
Frecuencia:	Muy Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén
Pre-Condiciones	El cliente debe tener un número de identificación única (Cedula) para ser registrado en el sistema.
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none"> 1. El actor ingresa a la opción de administrar Clientes 2. El sistema muestra el listado de los clientes registrados en el momento, permitiendo buscar por ciudad, cedula o nombre del cliente registrado. 3. El actor selecciona la opción de ingresar nuevo cliente 4. El sistema solicita la información obligatoria (Cedula, Nombre, Apellido, Ciudad, Dirección, teléfono, Celular) y permite ingresar un Email opcional. 5. El actor ingresa los campos y guarda 6. El sistema comprueba que no existan clientes con la misma cedula ya registrados. 7. En caso de que exista un registro con la misma cedula, el sistema informa que ya existe el cliente, en caso contrario, el sistema crea el registro e informa que se creó con éxito.
Flujo Alternativo 1: Modificar Clientes	<p>Este flujo comienza en el paso 2 del caso de uso, cuando el usuario selecciona un cliente del listado para modificación.</p> <ol style="list-style-type: none"> 1 El sistema muestra la información del cliente seleccionado 2 El actor cambia los campos necesarios, excepto la cedula, ya que es el identificador único del cliente y guarda.

-
- 3 El sistema actualiza la información y retoma el paso 2 de la secuencia normal

Flujo Alternativo 2: Eliminar Clientes

Este flujo comienza en el paso 2 del caso de uso, cuando el usuario selecciona un cliente del listado para modificación.

1. El sistema muestra la información del cliente seleccionado
2. El actor selecciona la opción de eliminar.
3. El sistema verifica que no existan asociaciones con esta cedula en las demás tablas relacionadas, en caso de que existan, no permite eliminar el registro y muestra un mensaje al usuario informando, en caso contrario, elimina el registro.
4. El sistema actualiza la información y retoma el paso 2 de la secuencia normal

Fuente: Elaboración propia.

5.2.1.2 Administrar parámetros

Tabla 3. Administrar parámetros.

Descripción:	Por medio de este, los actores pueden modificar y consultar los parámetros de la aplicación
Frecuencia:	Muy Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén
Pre-Condiciones	N/A
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none"> 1. El actor ingresa a la opción de administrar parámetros 2. El sistema muestra el listado de los parámetros controlados en el sistema

-
3. El actor selecciona un parámetro
 4. El sistema muestra el valor del parámetro con la opción de modificarlo
 5. El actor ingresa los campos y guarda
 6. El sistema comprueba que el valor sea válido para el parámetro
-

Fuente: Elaboración propia.

5.2.1.3 Administrar perfiles

Tabla 4. Administrar perfiles.

Descripción:	Por medio de este, los actores pueden crear, eliminar, modificar y consultar los perfiles
Frecuencia:	Muy Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén
Pre-Condiciones	El cliente debe tener un número de identificación única (Cedula) para ser registrado en el sistema.
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none"> 1. El actor ingresa a la opción de administrar perfiles 2. El sistema muestra el listado de los perfiles registrados en el momento 3. El actor selecciona la opción de ingresar nuevo perfil 4. El sistema solicita la información obligatoria 5. El actor ingresa los campos y guarda 6. El sistema comprueba que no existan perfiles con la misma descripción ya registrados. 7. En caso de que exista un registro con la misma descripción, el sistema informa que ya existe, en caso

contrario, el sistema crea el registro e informa que se creó con éxito.

Flujo Alternativo 1: Modificar Clientes Este flujo comienza en el paso 2 del caso de uso, cuando el usuario selecciona un perfil del listado para modificación.

- 1 El sistema muestra la información del perfil seleccionado
- 2 El actor cambia los campos necesarios y guarda.
- 3 El sistema actualiza la información y retoma el paso 2 de la secuencia normal

Flujo Alternativo 2: Eliminar Clientes Este flujo comienza en el paso 2 del caso de uso, cuando el usuario selecciona un cliente del listado para eliminar.

1. El sistema muestra la información del perfil seleccionado
2. El actor selecciona la opción de eliminar.
3. El sistema verifica que no existan asociaciones con este perfil en las demás tablas relacionadas, en caso de que existan, no permite eliminar el registro y muestra un mensaje al usuario informando, en caso contrario, elimina el registro.
4. El sistema actualiza la información y retoma el paso 2 de la secuencia normal

Fuente: Elaboración propia.

5.2.1.4 Administrar ítems

Tabla 5. Administrar ítems.

Descripción:	Por medio de este, los actores pueden crear, eliminar, modificar y consultar los ítems de inventario
Frecuencia:	Muy Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta

Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén
Pre-Condiciones	El ítem debe tener un código único generado, generalmente es utilizado el mismo código de barras de la fábrica. En caso de que no lo posea, se debe generar un código único.
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none"> 1. El actor ingresa a la opción de administrar ítems 2. El sistema muestra el listado de los ítems registrados en el momento, permitiendo buscar por código o descripción de estos 3. El actor selecciona la opción de ingresar nuevo ítem 4. El sistema solicita la información obligatoria (código, descripción, valor, valor de venta, % IVA, Imágenes, propiedades y cantidad) 5. El actor ingresa los campos y guarda 6. El sistema comprueba que no existan ítems con el mismo código ya registrados. 7. En caso de que exista un registro con el mismo código, el sistema informa que ya existe el ítem, en caso contrario, el sistema crea el registro e informa que se creó con éxito.
Flujo Alternativo 1: Modificar Clientes	<p>Este flujo comienza en el paso 2 del caso de uso, cuando el usuario selecciona un ítem del listado para modificación.</p> <ol style="list-style-type: none"> 1 El sistema muestra la información del ítem seleccionado 2 El actor cambia los campos necesarios, excepto el código del ítem, ya que es el identificador único y guarda. 3 El sistema actualiza la información y retoma el paso 2 de la secuencia normal
Flujo Alternativo 2: Eliminar Clientes	<p>Este flujo comienza en el paso 2 del caso de uso, cuando el usuario selecciona un ítem del listado para eliminar.</p> <ol style="list-style-type: none"> 1. El sistema muestra la información del ítem seleccionado 2. El actor selecciona la opción de eliminar. 3. El sistema verifica que no existan asociaciones con este ítem en las demás tablas relacionadas, en caso de que existan, no permite eliminar el registro y muestra

-
- un mensaje al usuario informando, en caso contrario, elimina el registro.
4. El sistema actualiza la información y retoma el paso 2 de la secuencia normal
-

Fuente: Elaboración propia.

5.2.1.5 Administrar servicios

Tabla 6. Administrar servicios.

Descripción:	Por medio de este, los actores pueden crear, eliminar, modificar y consultar los servicios adicionales ofrecidos por el almacén
Frecuencia:	Muy Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén
Pre-Condiciones	
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none"> 1. El actor ingresa a la opción de administrar servicios 2. El sistema muestra el listado de los servicios registrados en el momento, permitiendo buscar por código o descripción de estos 3. El actor selecciona la opción de ingresar nuevo servicio 4. El sistema solicita la información obligatoria (código, descripción y valor) 5. El actor ingresa los campos y guarda 6. El sistema comprueba que no existan servicios con el mismo código ya registrados. 7. En caso de que exista un registro con el mismo código, el sistema informa que ya existe el servicio,

en caso contrario, el sistema crea el registro e informa que se creó con éxito.

Flujo Alternativo 1: Modificar Servicios

Este flujo comienza en el paso 2 del caso de uso, cuando el usuario selecciona un servicio del listado para modificación.

- 1 El sistema muestra la información del servicio seleccionado
- 2 El actor cambia los campos necesarios, excepto el código del servicio, ya que es el identificador único y guarda.
- 3 El sistema actualiza la información y retoma el paso 2 de la secuencia normal

Flujo Alternativo 2: Eliminar Servicios

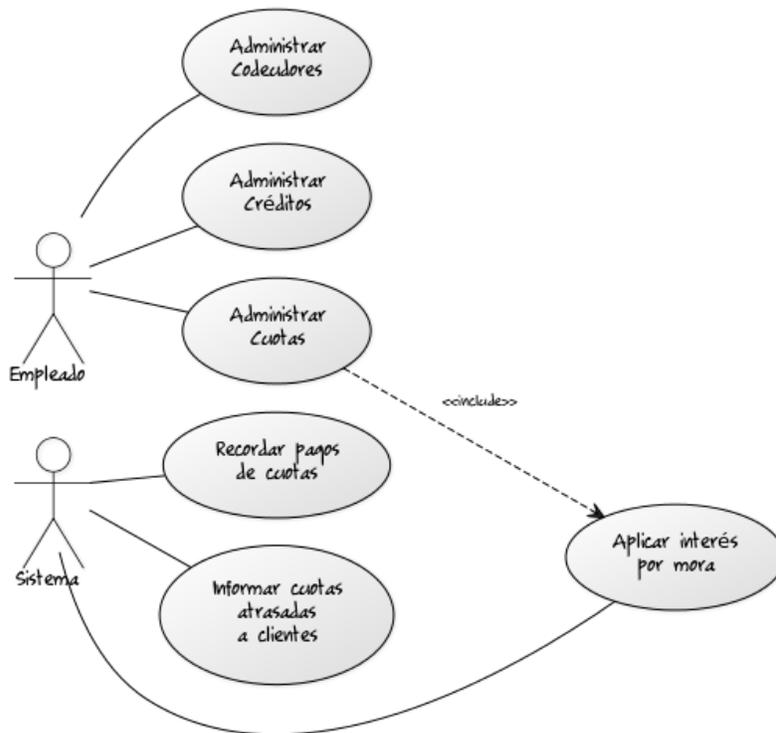
Este flujo comienza en el paso 2 del caso de uso, cuando el usuario selecciona un servicio del listado para eliminar.

1. El sistema muestra la información del servicio seleccionado
2. El actor selecciona la opción de eliminar.
3. El sistema verifica que no existan asociaciones con este servicio en las demás tablas relacionadas, en caso de que existan, no permite eliminar el registro y muestra un mensaje al usuario informando, en caso contrario, elimina el registro.
4. El sistema actualiza la información y retoma el paso 2 de la secuencia normal

Fuente: Elaboración propia.

5.2.2 Cuentas por cobrar

Diagrama 3 Diagrama casos de uso - Cuentas por cobrar



Fuente: Elaboración propia con base en la herramienta YUML (Yuml, 2016).

5.2.2.1 Administrar codeudores

Tabla 7. Administrar codeudores

Descripción:	Por medio de este, los actores pueden crear, eliminar, modificar y consultar los codeudores de los clientes
Frecuencia:	Muy Alta
Importancia	Muy Alta
Urgencia	Urgente

Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén
Pre-Condiciones	El codeudor debe haberse creado como cliente, con toda la información requerida ingresada
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none"> 1. El actor ingresa a la opción de administrar Codeudores 2. El sistema muestra el listado de los clientes registrados en el momento, permitiendo buscar por ciudad, cedula o nombre del cliente registrado. 3. El actor selecciona un cliente 4. El sistema muestra los codeudores registrados de ese cliente 5. El actor selecciona registrar Nuevo Codeudor 6. El sistema solicita la identificación del codeudor 7. El actor ingresa los campos y guarda 8. El sistema comprueba que no exista el codeudor para el mismo cliente ya registrado 9. En caso de que exista un registro con la misma cedula, el sistema informa que ya existe el codeudor, en caso contrario, el sistema crea el registro e informa que se creó con éxito.
Flujo Alternativo 1: Eliminar Codeudores	<p>Este flujo comienza en el paso 4 del caso de uso, cuando el usuario selecciona un cliente del listado para modificación.</p> <ol style="list-style-type: none"> 1 El sistema muestra la información del cliente seleccionado 2 El actor selecciona el codeudor que desea eliminar 3 El sistema pide confirmación del usuario 4 El sistema actualiza la información y retoma el paso 4 de la secuencia normal

Fuente: Elaboración propia.

5.2.2.2 Administrar créditos

Tabla 8. Administrar créditos.

Descripción:	Por medio de este, los actores pueden inactivar, registrar abonos y consultar los créditos de los clientes
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén
Pre-Condiciones	N/A
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none"> 1. El actor ingresa a la opción de administrar Créditos 2. El sistema muestra el listado de los créditos registrados en el momento, permitiendo buscar por cliente, factura, estado del crédito y fechas de creación 3. El sistema muestra los créditos registrados según los criterios de búsqueda 4. El actor selecciona un crédito. 5. El sistema muestra la información de cuotas pagadas y adeudadas a la fecha 6. El actor selecciona la opción de registrar abono 7. El sistema pide la información del valor a pagar por el cliente. 8. El actor ingresa el valor y si este es mayor que la cuota, descuenta de la próxima cuota el excedente y establece como pagada la cuota actual, y si es menor, registra como pago parcial de la cuota. 9. El sistema actualiza la información del crédito con el abono
Flujo Alternativo 1: Inactivar Créditos	<p>Este flujo comienza en el paso 4 del caso de uso, cuando el usuario selecciona un crédito del listado para modificación.</p> <ol style="list-style-type: none"> 1 El sistema muestra la información del crédito seleccionado

	<ol style="list-style-type: none"> 2 El actor selecciona el crédito que desea eliminar 3 El sistema pide confirmación del usuario 4 El sistema actualiza la información y retoma el paso 4 de la secuencia normal
Flujo Alternativo 2: Recalcular Créditos	<p>Este flujo comienza en el paso 4 del caso de uso, cuando el usuario selecciona un crédito del listado para modificación.</p> <ol style="list-style-type: none"> 1 El sistema muestra la información del crédito seleccionado 2 El actor selecciona la opción de recalcular crédito. 3 El sistema pide el número de cuotas y la confirmación del usuario 4 El sistema recalcula el crédito tomando como base el total de la deuda capital y los intereses del nuevo número de cuotas, actualiza la información del crédito y retoma el paso 4 de la secuencia normal

Fuente: Elaboración propia

5.2.2.3 Administrar cuotas

Tabla 9. Administrar cuotas.

Descripción:	Por medio de este, los actores pueden registrar pagos parciales, pagos totales y descuentos en las cuotas
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén
Pre-Condiciones	N/A
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none"> 1. Se ejecuta el CU Administrar Créditos hasta el paso 5 2. El usuario selecciona una cuota 3. El sistema muestra el valor a cancelar y los intereses por mora si registra, y pide un descuento en porcentaje o en pesos

-
4. El usuario ingresa el valor a pagar y el descuento si aplica, si es en porcentaje, el sistema debe convertirlo en pesos.
 5. El sistema guarda la información del pago con la condición de que si el pago es menor al valor de la cuota, esta no se marca como pagada, y tampoco recibirá pagos mayores al valor de la cuota más los intereses por mora.
-

Fuente: Elaboración propia

5.2.2.4 Aplicar intereses por mora

Tabla 10. Aplicar interés por mora.

Descripción:	Por medio de este, el sistema aplica los intereses por mora a las cuotas atrasadas de los créditos activos.
Frecuencia:	Alta
Importancia	Baja
Urgencia	No es Urgente
Estabilidad	Alta
Actor Primario	Sistema
Actores Secundarios	N/A
Pre-Condiciones	N/A
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none"> 1. El sistema consulta diariamente los créditos activos que presenten cuotas atrasadas 2. El sistema calcula el interés por mora del día según un parámetro del porcentaje de cobro 3. El sistema adiciona el valor del interés por mora a cada cuota atrasada

Fuente: Elaboración propia

5.2.2.5 Recordar pagos de cuotas

Tabla 11. Recordar pagos de cuotas.

Descripción:	Por medio de este, el sistema envía correos a los clientes del almacén con la información completa, mayormente con el email valido.
Frecuencia:	Alta
Importancia	Baja
Urgencia	No es Urgente
Estabilidad	Alta
Actor Primario	Sistema
Actores Secundarios	N/A
Pre-Condiciones	N/A
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none">1. El sistema consulta diariamente los créditos activos que presenten cuotas con un número de días parametrizable para el vencimiento de la cuota.2. Si la cuota se encuentra en el rango de días, se debe enviar un correo al cliente del almacén informando que la cuota vencerá en el número de días X, donde X es la diferencia de la fecha actual y la fecha de vencimiento de la cuota.3. El sistema marca la cuota como avisada para no enviar el mismo correo 2 veces a la misma persona.

Fuente: Elaboración propia.

5.2.2.6 Informar cuotas atrasadas a clientes

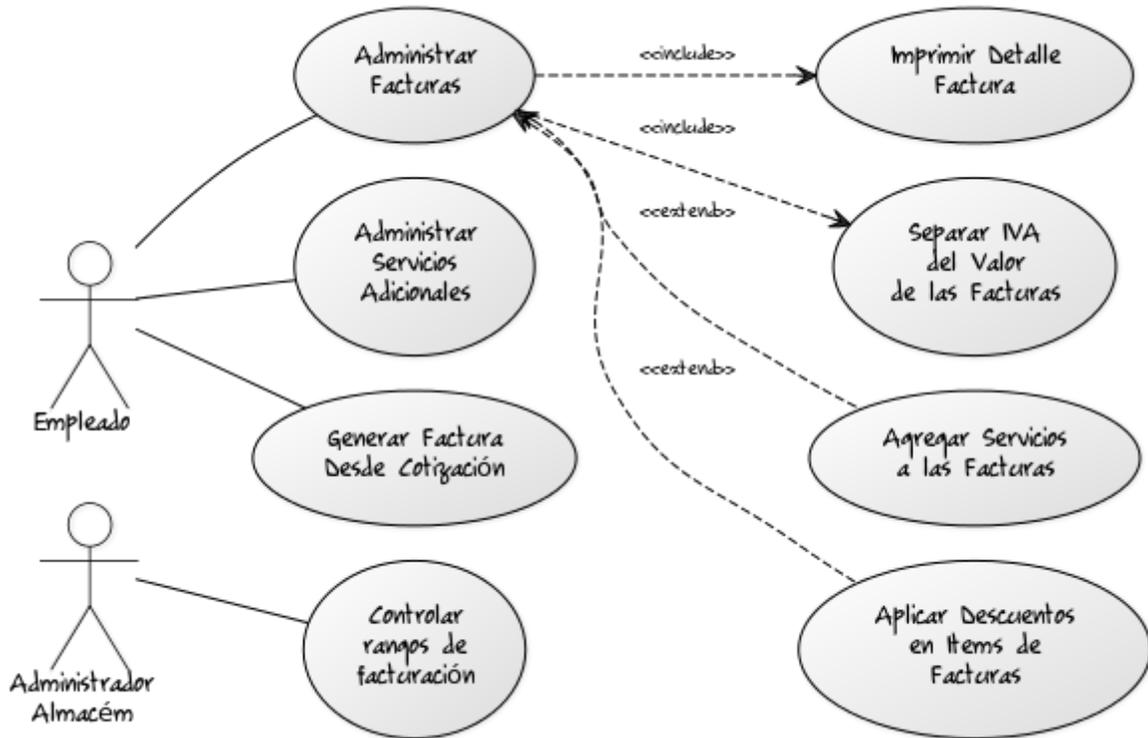
Tabla 12. Informar cuotas atrasadas a clientes.

Descripción:	Por medio de este, el sistema envía correos a los clientes del almacén con la información completa, mayormente con el email valido.
Frecuencia:	Alta
Importancia	Baja
Urgencia	No es Urgente
Estabilidad	Alta
Actor Primario	Sistema
Actores Secundarios	N/A
Pre-Condiciones	N/A
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none">1. El sistema consulta diariamente los créditos activos que presenten cuotas vencidas.2. El sistema cuenta el número de cuotas que tiene el cliente sin pagar y estén vencidas.3. Si hay más de una cuota, se debe advertir al cliente sobre el cobro jurídico en el que puede incurrir en caso de incumplimiento.4. El sistema marca la cuota debe como avisada para no enviar el mismo correo 2 veces a la misma persona.

Fuente: Elaboración propia.

5.2.3 Facturación

Diagrama 4 Diagrama casos de uso - Facturación



Fuente: Elaboración propia con base en la herramienta YUML (Yuml, 2016).

5.2.3.1 Administrar facturas

Tabla 13. Administrar facturas.

Descripción:	Por medio de este, el sistema almacenará la información de facturas generadas por las compras de los clientes
Frecuencia:	Media
Importancia	Muy Alta

Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén
Pre-Condiciones	N/A
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario ingresa a la opción de Administrar Facturas 2. El sistema lista las facturas existentes en el sistema, organizadas por la fecha de pago 3. El usuario selecciona una factura de la lista 4. El sistema muestra la información en detalle de la factura, mostrando los campos de Tipo de pago, valor, información del crédito en caso de que exista, usuario tomador y fiadores 5. El usuario retorna al listado
Flujo Alternativo 1	El Flujo comienza en el paso 3 de la secuencia normal
Anular Factura	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de anular la factura, pidiendo al usuario que ingrese una justificación para la acción. 2. El usuario ingresa la justificación. 3. El sistema guarda la justificación y anula la factura, en caso de poseer un crédito, lo pasa a estado anulado.
Flujo Alternativo 1	El Flujo comienza en el paso 3 de la secuencia normal
Imprimir Factura	<ol style="list-style-type: none"> 1. El usuario selecciona la opción de imprimir la factura 2. Se ejecuta el Caso de Uso Facturación: Imprimir detalle Factura

Fuente: Elaboración propia.

5.2.3.2 Controlar rangos de facturación.

Tabla 14. Controlar rangos de facturación.

Descripción:	Por medio de este, el sistema manejará los rangos de facturación impuestos por la DIAN
---------------------	---

Frecuencia:	Muy Baja
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	administrador del almacén
Actores Secundarios	N/A
Pre-Condiciones	N/A
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario ingresa a la opción de Administrar Rangos de Facturación 2. El sistema lista el rango actual de facturación con la posibilidad de listar los rangos anteriores. 3. El usuario selecciona la opción de nuevo rango 4. El sistema pide el rango inicial, el rango final y la fecha de vencimiento 5. El usuario ingresa los datos necesarios 6. El sistema valida que no existan valores de rangos anteriores iguales, inactiva el rango actual y activa el nuevo rango para las facturas ingresadas.
Flujo Alternativo 1 Eliminar Rango	<p>El Flujo comienza en el paso 2 de la secuencia normal</p> <ol style="list-style-type: none"> 1. El usuario selecciona la opción de eliminar un rango del listado 2. El sistema comprueba que no sea el rango actual y que no tenga facturas registradas, en caso contrario, muestra un mensaje de advertencia al usuario. 3. El sistema elimina el rango.
Flujo Alternativo 1 Imprimir Factura	<p>El Flujo comienza en el paso 3 de la secuencia normal</p> <ol style="list-style-type: none"> 1. El usuario selecciona la opción de imprimir la factura 2. Se ejecuta el Caso de Uso Facturación: Imprimir Factura

Fuente: Elaboración propia.

5.2.3.3 Imprimir detalle de factura

Tabla 15. Imprimir detalle de factura.

Descripción:	Por medio de este, el sistema imprimirá las facturas
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Sistema
Actores Secundarios	N/A.
Pre-Condiciones	EL sistema genera el llamado para este caso de uso
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none"> 1. El sistema consulta la factura deseada 2. El sistema pide seleccionar la impresora deseada para esta acción, permitiendo imprimir o crear PDF 3. El usuario selecciona el modo de impresión 4. El sistema genera el documento.

Fuente: Elaboración propia.

5.2.3.4 Aplicar descuentos en ítems de facturas

Tabla 16. Aplicar descuentos en ítems de facturas.

Descripción:	Por medio de este, el usuario podrá registrar descuentos de productos en porcentaje o en valor.
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén
Pre-Condiciones	Debe existir al menos un ítem en la factura al cual aplicarle el descuento
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none"> 1. El usuario ingresa un ítem a la factura

	<ol style="list-style-type: none"> 2. El sistema detecta el ítem o el servicio registrado y da la posibilidad de registrar un porcentaje o un valor de descuento. 3. El usuario registra el valor de descuento 4. El sistema recalcula el total de la factura
Flujo Alternativo Porcentaje	<p>Este caso de uso comienza en el paso 3 cuando el usuario registra un porcentaje</p> <ol style="list-style-type: none"> 1. El usuario ingresa un valor en porcentaje del total del ítem 2. El sistema calcula el valor del descuento del ítem 3. El caso de uso continúa en el paso 4 de la secuencia normal

Fuente: Elaboración propia.

5.2.3.5 Separar IVA del valor de las facturas

Tabla 17. Separar IVA del valor de las facturas.

Descripción:	Por medio de este, el sistema separa el valor neto de la factura y el IVA, para reportes de la DIAN
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Sistema
Actores Secundarios	N/A
Pre-Condiciones	Debe registrarse la factura para su cobro
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none"> 1. El sistema toma el valor del IVA de cada ítem y saca la sumatoria total del IVA de los ítems que componen la factura 2. El sistema guarda el total de la factura y el IVA por aparte.

Fuente: Elaboración propia.

5.2.3.6 Administrar servicios adicionales

Tabla 18. Administrar servicios adicionales.

Descripción:	Por medio de este, el usuario podrá crear, modificar o inactivar servicios adicionales a la venta de ítems.
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén
Pre-Condiciones	Debe registrarse la factura para su cobro
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none"> 1. El sistema lista los servicios creados en el sistema, mostrando el valor del servicio y su descripción. 2. El actor selecciona nuevo servicio 3. El sistema pide ingresar la descripción del servicio y el valor de este. 4. El actor ingresa los datos solicitados 5. El sistema guarda el servicio registrado y vuelve al paso 1 de la secuencia normal.
Flujo Alternativo 1 Modificar Servicio	<p>Este flujo comienza en el paso 1 cuando el actor selecciona un servicio creado anteriormente.</p> <ol style="list-style-type: none"> 1. El sistema muestra la información del servicio con la posibilidad de modificar el valor y el campo activo 2. El actor modifica los campos necesarios. 3. El sistema guarda y retorna al paso 1 de la secuencia normal de pasos

Fuente: Elaboración propia.

5.2.3.7 Agregar servicios a las facturas

Tabla 19 Agregar servicios a las facturas.

Descripción:	Por medio de este, el usuario podrá agregar servicios adicionales a la venta de ítems en las facturas de compra.
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén
Pre-Condiciones	Debe existir una factura para añadir los servicios.
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none">1. El sistema lista los servicios creados en el sistema, mostrando el valor del servicio y su descripción.2. El actor selecciona un servicio de la lista y lo agrega a la factura3. El sistema pide ingresar el descuento de este, en caso de que exista4. El actor ingresa los datos solicitados5. El sistema adiciona el valor del servicio con el descuento en el detalle de la factura, y actualiza el total.

Fuente: Elaboración propia.

5.2.3.8 Generar factura desde una cotización

Tabla 20. Generar factura desde una cotización.

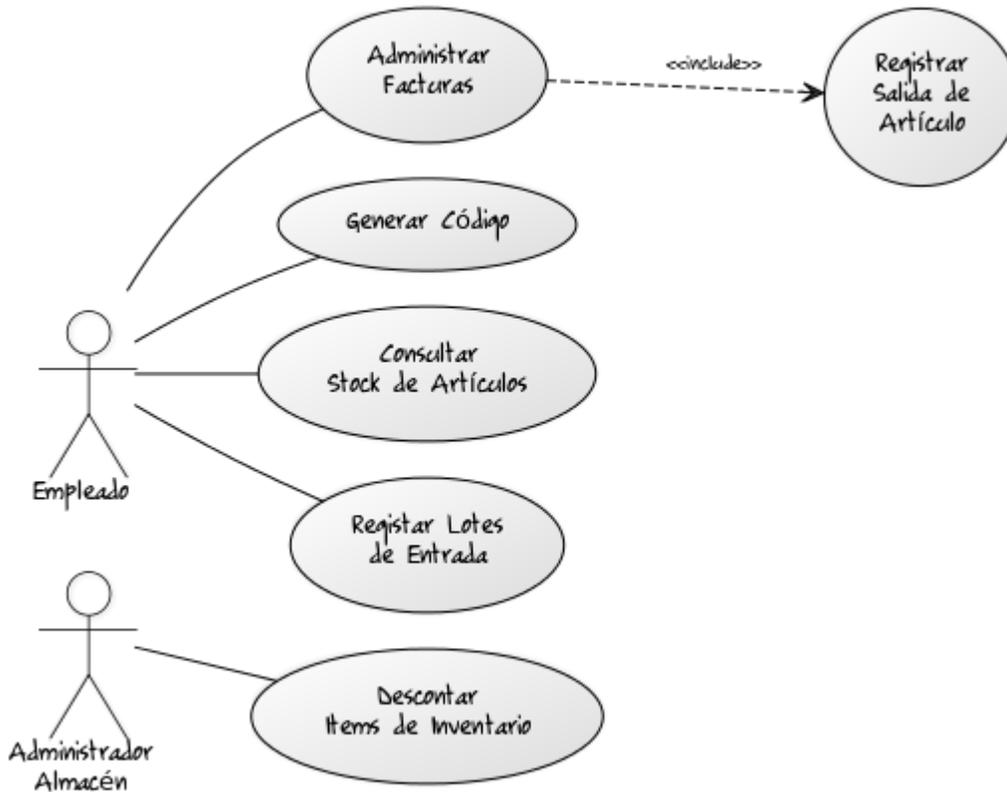
Descripción:	Por medio de este, el usuario podrá generar una factura de una cotización previa
Frecuencia:	Alta

Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén
Pre-Condiciones	Debe generarse una cotización con al menos 1 ítem facturable.
Post-Condiciones	N/A.
Secuencia Normal	<ol style="list-style-type: none"> 1. EL sistema muestra al actor la opción de generar factura 2. El actor selecciona la opción 3. El sistema consulta el rango de facturación de la DIAN y el número actual de la factura, genera un nuevo consecutivo para la factura actual y agrega los ítems de la cotización. 4. El actor valida la información y procede con la generación e impresión de la factura. 5. El sistema guarda el nuevo consecutivo de factura
Flujos Alternos	Este flujo comienza en el paso 4 de la secuencia normal,
Modificar Factura	cuando el actor no valida la información de la factura
Generada	<ol style="list-style-type: none"> 1. El actor cancela la generación y sigue en la cotización. 2. El actor modifica los datos necesarios para generar nuevamente la factura 3. El caso de uso continua en el paso 3 de la secuencia normal

Fuente: Elaboración propia.

5.2.4 Inventario

Diagrama 5 Diagrama de caso de uso - Inventario



Fuente: Elaboración propia con base en la herramienta YUML (Yuml, 2016).

5.2.4.1 Generar código único

Tabla 21. Generar código único.

Descripción:	Por medio de este, el usuario podrá obtener un código único para registrar un ítem del inventario
Frecuencia:	baja

Importancia	Muy Alta
Urgencia	Poco Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén
Pre-Condiciones	Se debe tener ubicarse en la opción de ingresar nuevo ítem, y que este no posea identificación única por código de barras
Post-Condiciones	El actor finaliza el guardado del ítem.
Secuencia Normal	<ol style="list-style-type: none"> 1. EL sistema genera un código GUID de identificación. 2. El sistema detecta si debe imprimir el código en una impresora de códigos de barra por medio de un parámetro “Imprimir código generado” 3. El sistema muestra el código generado
Flujos Alternos	Este flujo comienza en el paso 2 de la secuencia normal, cuando el código se imprime
Imprimir Código	<ol style="list-style-type: none"> 1. EL sistema muestra las impresoras disponibles. 2. El actor selecciona la impresora 3. El sistema imprime el código generado 4. Se retorna a la secuencia normal de pasos en el paso 3

Fuente: Elaboración propia.

5.2.4.2 Consultar stock de artículos

Tabla 22. Consultar stock de artículos.

Descripción:	Por medio de este, el usuario podrá consultar la disponibilidad de ítems al generar una factura
Frecuencia:	baja
Importancia	Muy Alta
Urgencia	Poco Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén

Pre-Condiciones	El actor debe haber seleccionado la opción de generar factura, luego de crear una cotización con más de 1 ítem.
Post-Condiciones	El actor finaliza el guardado de la factura
Secuencia Normal	<ol style="list-style-type: none"> 1. El sistema consulta los ítems generados en la cotización 2. EL sistema consulta para cada ítem, la cantidad de inventario que posee, si esta es mayor que 0 entonces permite proceder con la factura, en caso contrario, el sistema avisa que el ítem no tiene inventario y no puede ser vendido.

Fuente: Elaboración propia.

5.2.4.3 Registrar lotes de entrada

Tabla 23. Registrar lotes de entrada.

Descripción:	Por medio de este, el usuario podrá ingresar lotes de inventario de un Ítem
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Poco Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén
Pre-Condiciones	El ítem que ingresa debe existir.
Post-Condiciones	
Secuencia Normal	<ol style="list-style-type: none"> 1. El actor ingresa a la opción de registrar lote de entrada. 2. El sistema pide los datos de ítem, valor de compra, valor de venta (Se muestra el ultimo valor de venta registrado con la opción de actualizarlo) y cantidad. 3. El actor ingresa los datos solicitados y guarda 4. EL sistema calcula el precio de compra del articulo por medio de la siguiente formula:

$$5. X = \frac{Cantidad_i * Valor_i + Cantidad_a * Valor_a}{Cantidad_i + Cantidad_a}$$

Cantidad(i): Cantidad existente en el inventario

Cantidad(a): Cantidad actual que ingresa al inventario

Valor(i): Valor de los ítems del inventario existente

Valor(a): Valor de los ítems que se ingresan al inventario

6. El sistema actualiza el valor de compra del ítem en el inventario y guarda el resto de información..
-

Fuente: Elaboración propia.

5.2.4.4 Registrar salida de artículo

Tabla 24. Registrar salida de artículo.

Descripción:	Por medio de este, el sistema descontará ítems vendidos
Frecuencia:	Muy Alta
Importancia	Muy Alta
Urgencia	Poco Urgente
Estabilidad	Alta
Actor Primario	Sistema
Actores Secundarios	N/A
Pre-Condiciones	El ítem que ingresa debe existir Se debe descontar después de generada una factura
Post-Condiciones	Actualizar las cantidades existentes de los ítems
Secuencia Normal	<ol style="list-style-type: none"> 1. El sistema descuenta la cantidad vendida de cada ítem en la factura al inventario. 2. Si algún ítem queda cercano al parámetro establecido “Cantidad aviso de inventario” y el parámetro “Enviar correo de aviso de inventario” está activo, entonces el sistema envía un correo al empleado y al administrador del almacén informando que el ítem se encuentra por debajo del valor de aviso y se debe realizar un pedido.

Fuente: Elaboración propia.

5.2.4.5 Descontar ítems de inventario

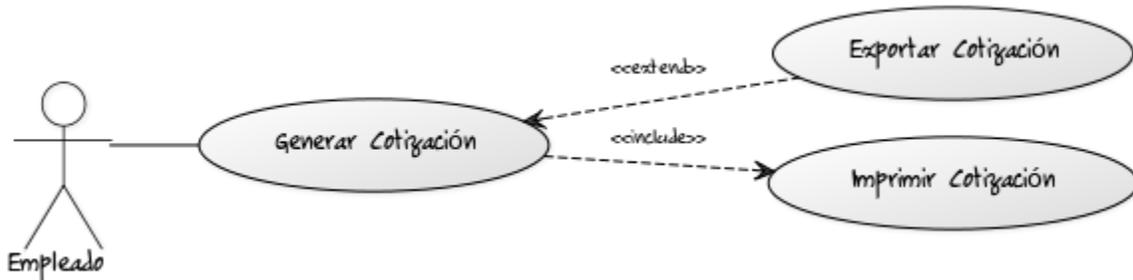
Tabla 25. Descontar ítems de inventario.

Descripción:	Por medio de este, el usuario podrá descontar ítems defectuosos, robados o faltantes
Frecuencia:	Baja
Importancia	Muy Alta
Urgencia	Poco Urgente
Estabilidad	Alta
Actor Primario	Administrador del Sistema, administrador del almacén
Actores Secundarios	N/A
Pre-Condiciones	El ítem que ingresa debe existir
Post-Condiciones	N/A
Secuencia Normal	<ol style="list-style-type: none">1. El usuario selecciona el ítem del que desea descontar existencias, la cantidad a descontar y la justificación del descuento de existencias.2. El sistema verifica que exista la cantidad ingresada, en caso de que no exista, avisa al usuario que no se puede descontar, en caso contrario, descuenta la cantidad ingresada por el usuario y actualiza la cantidad existente.3. Si algún ítem queda cercano al parámetro establecido “Cantidad aviso de inventario” y el parámetro “Enviar correo de aviso de inventario” está activo, entonces el sistema envía un correo al empleado y al administrador del almacén informando que el ítem se encuentra por debajo del valor de aviso y se debe realizar un pedido.

Fuente: Elaboración propia.

5.2.5 Cotizaciones

Diagrama 6 Diagrama de casos de uso - Cotiozaciones



Fuente: Elaboración propia con base en la herramienta YUML (Yuml, 2016).

5.2.5.1 Generar cotización

Tabla 26. Generar cotización.

Descripción:	Por medio de este, el sistema generará cotización de la venta de artículos
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén, Usuarios externos
Pre-Condiciones	El usuario se registre en el sistema con los datos basicos
Post-Condiciones	
Secuencia Normal	<ol style="list-style-type: none"> 1. El actor ingresa a la opción de Generar Cotización 2. El sistema lista los ítems y los servicios para seleccionar de la lista

-
3. El actor selecciona los ítems y servicios con la cantidad que desea y los agrega a la cotización
 4. El sistema agrega el ítem o servicio seleccionado y actualiza el valor total de la cotización.
 5. El actor termina la cotización
 6. El sistema guarda la información de la cotización y muestra la opción de imprimir o exportar.

Flujo Alternativo
Usuario Interno

Este flujo empieza en el paso 4 de la secuencia normal.

1. Si el usuario es interno, el sistema pide valores de descuento para los ítems o servicios adicionados.
2. El sistema descuenta el valor ingresado por el actor de la cotización.
3. Retorna al paso 5 de la secuencia normal.

Flujo Alternativo
Imprimir Cotización

Este flujo comienza en el paso 6 de la secuencia normal, cuando el actor decide imprimir o exportar la cotización.

1. Se ejecuta el caso de uso Imprimir Cotización.

Fuente: Elaboración propia.

5.2.5.2 Imprimir cotización

Tabla 27. Imprimir cotización.

Descripción:	Por medio de este, el sistema imprime cotizaciones de la venta de artículos
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Sistema
Actores Secundarios	N/A
Pre-Condiciones	Se haya creado una cotización en el sistema.

Post-Condiciones	N/A
Secuencia Normal	<ol style="list-style-type: none"> 1. El sistema consulta las impresoras disponibles 2. EL usuario selecciona la impresora a utilizar 3. El sistema imprime la cotización con el nombre del usuario, número de cotización, fecha y detalles de los artículos seleccionados.

Fuente: Elaboración propia.

5.2.5.3 Exportar cotización

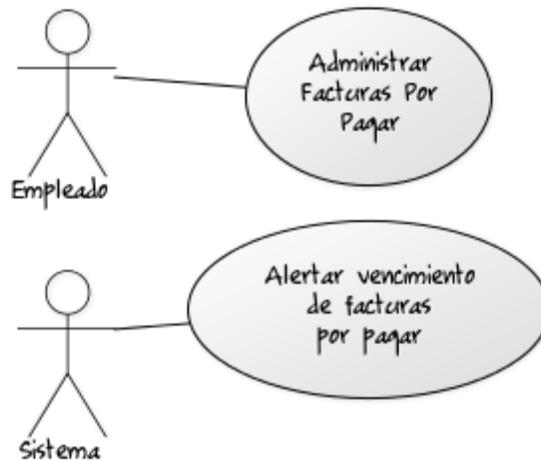
Tabla 28. Exportar cotización.

Descripción:	Por medio de este, el sistema exporta cotizaciones de la venta de artículos
Frecuencia:	Baja
Importancia	Baja
Urgencia	Poco Urgente
Estabilidad	Alta
Actor Primario	Sistema
Actores Secundarios	N/A
Pre-Condiciones	Se haya creado una cotización en el sistema.
Post-Condiciones	N/A
Secuencia Normal	<ol style="list-style-type: none"> 1. El sistema exporta la cotización con el nombre del usuario, número de cotización, fecha y detalles de los artículos seleccionados, en formato Excel 2010

Fuente: Elaboración propia.

5.2.6 Cuentas x pagar

Diagrama 7 Diagrama de casos de uso - Cuentas por pagar



Fuente: Elaboración propia con base en la herramienta YUML (Yuml, 2016).

5.2.6.1 Administrar facturas por pagar

Tabla 29. Administrar facturas por pagar.

Descripción:	Por medio de este, el usuario podrá crear facturas, anular facturas o registrar pagos de estas
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén, Usuarios externos
Pre-Condiciones	Debe existir el proveedor registrado en el sistema

Post-Condiciones	N/A
Secuencia Normal	<ol style="list-style-type: none"> 1. El actor ingresa a la opción de Administrar Facturas 2. El sistema lista las facturas registradas en el ultimo mes 3. El usuario selecciona una factura 4. El sistema muestra el detalle de la factura y pide los datos del pago si esta se encuentra en estado pendiente por pagar (Numero de referencia, banco, valor, descuento y fecha de pago) 5. El actor ingresa los datos solicitados 6. El sistema guarda los datos y actualiza la factura como pagada.
Flujo Alterno Anular Factura	<p>Este flujo empieza en el paso 4 de la secuencia normal.</p> <ol style="list-style-type: none"> 1. El actor selecciona la opción de anular factura 2. El sistema pide la justificación y anula la factura 3. El sistema envía un correo a los usuarios internos informando la anulación de la factura y el motivo. 4. El sistema regresa al paso 2 de la secuencia normal.
Flujo Alterno Registrar Nueva Factura	<p>Este flujo comienza en el paso 2 de la secuencia normal</p> <ol style="list-style-type: none"> 1. El actor selecciona la opción de Registrar Nueva Factura. 2. El sistema solicita los datos (Numero de factura, proveedor, fecha de pago, fecha1 de pago con descuento, descuento fecha 1, fecha2 de pago con descuento, descuento fecha2 y valor) 3. El actor ingresa los datos solicitados 4. El sistema guarda la factura por pagar y regresa al paso 2 de la secuencia normal

Fuente: Elaboración propia.

5.2.6.2 Alertar vencimiento de facturas por pagar

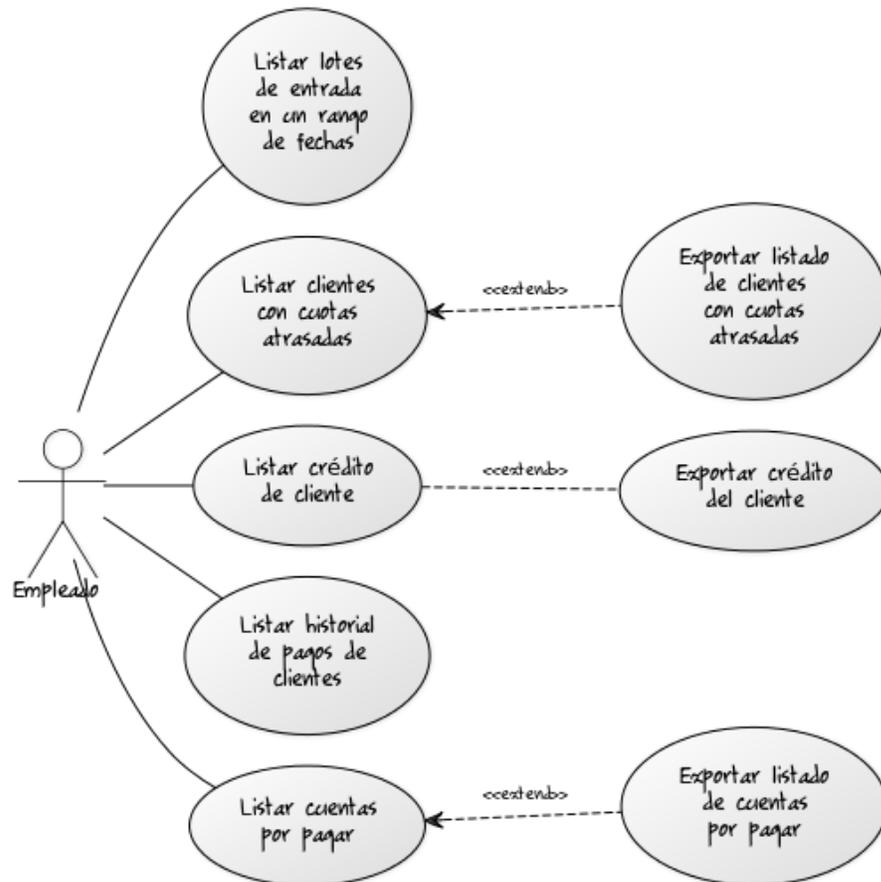
Tabla 30. Alertar vencimiento de facturas por pagar.

Descripción:	Por medio de este, el sistema alerta a los usuarios de las facturas próximas a vencer
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Sistema
Actores Secundarios	N/A
Pre-Condiciones	Deben existir facturas por pagar registradas y en estado pendiente
Post-Condiciones	N/A
Secuencia Normal	<ol style="list-style-type: none">1. El sistema consulta cada día las facturas pendientes por pagar.2. Por cada factura determina si tiene descuento por pronto pago y recalcula el total del pago teniendo en cuenta el descuento en la fecha actual.3. El sistema envía un correo a los usuarios con la anterior información

Fuente: Elaboración propia.

5.2.7 Reportes

Diagrama 8 Diagrama de casos de uso - Reportes



Fuente: Elaboración propia con base en la herramienta YUML (Yuml, 2016).

5.2.7.1 Listar lotes de entrada en un rango de fechas

Tabla 31. Listar lotes de entrada en un rango de fechas.

Descripción:	Por medio de este, el usuario podrá consultar los lotes de entrada de artículos en el inventario
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén, Usuarios externos
Pre-Condiciones	N/A
Post-Condiciones	N/A
Secuencia Normal	<ol style="list-style-type: none"> 1. El actor ingresa a la opción de lotes de entrada 2. El sistema lista por defecto los lotes de entrada de todos los artículos entre la fecha actual y la fecha actual – 1 mes mostrando los campos: <ul style="list-style-type: none"> • Artículo • Fecha de Ingreso • Cantidad • Valor • Valor Venta <p>Y muestra los filtros:</p> <ul style="list-style-type: none"> • Fecha inicial • Fecha Final • Artículo(s)
Flujo Alternativo	Este flujo comienza cuando el actor cambia uno de los filtros de datos
Filtro de datos	<ol style="list-style-type: none"> 1. El actor modifica uno de los filtros del reporte 2. El sistema muestra la información filtrada 3. Se retoma el paso 2 de la secuencia normal.

Fuente: Elaboración propia.

5.2.7.2 Listar clientes con cuotas atrasadas

Tabla 32. Listar clientes con cuotas atrasadas.

Descripción:	Por medio de este, el usuario podrá consultar los clientes que tengan cuotas atrasadas
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén, Usuarios externos
Pre-Condiciones	N/A
Post-Condiciones	N/A
Secuencia Normal	<ol style="list-style-type: none"> 1. El actor ingresa a la opción de Clientes atrasados 2. El sistema lista por defecto todos los clientes con mas de 1 o mas cuotas de atraso, mostrando los campos: <ul style="list-style-type: none"> • Identificación Cliente • Nombre • Teléfono • Dirección • Cuotas Atrasadas • Fecha Pago (Mayor Atraso) <p>Y muestra los filtros:</p> <ul style="list-style-type: none"> • Identificación Cliente • Nombre • Cuotas Atrasadas 3. El sistema muestra la opción de exportar si existe mas de 1 registro
Flujo Alterno	Este flujo comienza cuando el actor cambia uno de los filtros de datos
Filtro de datos	<ol style="list-style-type: none"> 1. El actor modifica uno de los filtros del reporte 2. El sistema muestra la información filtrada 3. Se retoma el paso 2 de la secuencia normal.

Flujo Alternativo Exportar	Este flujo comienza cuando el decide exportar los datos a Excel 1. Se ejecuta el Caso de Uso Exportar Listado de Clientes con Cuotas Atrasadas
-------------------------------	---

Fuente: Elaboración propia.

5.2.7.3 Exportar listado de clientes con cuotas atrasadas

Tabla 33. Exportar listado de clientes con cuotas atrasadas.

Descripción:	Por medio de este, el sistema exportará la información consultada
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Sistema
Actores Secundarios	N/A
Pre-Condiciones	Que exista más de 1 cliente con cuotas atrasadas
Post-Condiciones	N/A
Secuencia Normal	<ol style="list-style-type: none"> 1. El sistema consulta el listado de clientes con cuotas atrasadas. 2. El sistema valida que existan registros y genera el archivo de Excel formato 2007 con estos, en caso de que no existan registros, no crea el archivo.

Fuente: Elaboración propia.

5.2.7.4 Listar crédito de cliente

Tabla 34. Listar crédito de cliente.

Descripción:	Por medio de este, el actor podrá consultar el estado actual del crédito de un cliente
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén, Usuarios externos
Pre-Condiciones	N/A
Post-Condiciones	N/A
Secuencia Normal	<ol style="list-style-type: none"> 1. El actor ingresa a la opción de listado de créditos 2. El sistema pide la identificación del cliente 3. El actor ingresa la identificación y selecciona la opción de consultar o exportar 4. El sistema lista por defecto todos los créditos en estado abierto, con los siguientes campos: <ul style="list-style-type: none"> • Fecha • Valor Crédito • Numero de cuotas • Cuotas Pagadas • Cuotas atrasadas • Fecha de Atraso (Corresponde a la fecha de la cuota atrasada más antigua) • Número de factura • Estado del crédito <p>Y muestra los filtros:</p> <ul style="list-style-type: none"> • Identificación Cliente • Cuotas Atrasadas • Numero de Factura • Estado del Crédito 5. El sistema muestra la opción de exportar si existe más de 1 registro

Flujo Alternativo Filtro de datos	Este flujo comienza cuando el actor cambia uno de los filtros de datos <ol style="list-style-type: none"> 1. El actor modifica uno de los filtros del reporte 2. El sistema muestra la información filtrada 3. Se retoma el paso 2 de la secuencia normal.
Flujo Alternativo Exportar	Este flujo comienza cuando el actor decide exportar los datos a Excel después de ingresar la identificación del cliente <ol style="list-style-type: none"> 1. Se ejecuta el Caso de Uso Exportar Crédito Cliente
Flujo Alternativo Consultar Pagos	Este flujo comienza cuando el actor selecciona un crédito del listado <ol style="list-style-type: none"> 1. Se ejecuta el Caso de Uso Listar Historial de Pagos de Cliente

Fuente: Elaboración propia.

5.2.7.5 Listar historial de pagos de clientes

Tabla 35. Listar historial de pagos de clientes.

Descripción:	Por medio de este, el sistema lista la información de pagos pendientes y realizados de un crédito
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Sistema
Actores Secundarios	N/A
Pre-Condiciones	El actor debe seleccionar un crédito.
Post-Condiciones	N/A
Secuencia Normal	<ol style="list-style-type: none"> 1. El sistema consulta el crédito seleccionado 2. El sistema muestra el listado de cuotas con los siguientes campos <ul style="list-style-type: none"> • Numero de cuota

-
- Valor
 - Fecha de Pago programada
 - Fecha de pago real
 - Atrasado (Si/No)
 - Días de atraso
 - Intereses Por Mora
3. El sistema filtra la información por los siguientes campos
- Fecha de Pago Programada
 - Atrasado
-

Fuente: Elaboración propia.

5.2.7.6 Exportar crédito del cliente

Tabla 36. Exportar crédito del cliente.

Descripción:	Por medio de este, el sistema exporta la información de pagos pendientes y realizados de un crédito
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Sistema
Actores Secundarios	N/A
Pre-Condiciones	El actor debe seleccionar un crédito.
Post-Condiciones	N/A
Secuencia Normal	<ol style="list-style-type: none"> 1. El sistema valida que exista el crédito seleccionado y consulta la información de pagos 2. El sistema exporta la información consultada en formato Excel 2007

Fuente: Elaboración propia.

5.2.7.7 Listar cuentas por pagar

Tabla 37. Listar cuentas por pagar.

Descripción:	Por medio de este, el actor podrá consultar las cuentas por pagar y cuentas pagadas en un rango de fechas
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente
Estabilidad	Alta
Actor Primario	Empleado del almacén
Actores Secundarios	Administrador del Sistema, administrador del almacén, Usuarios externos
Pre-Condiciones	N/A
Post-Condiciones	N/A
Secuencia Normal	<ol style="list-style-type: none"> 1. El actor ingresa a la opción de listado de Cuentas 2. El sistema lista la información de cuentas cuya fecha de vencimiento se encuentre entre la fecha actual y un mes adelante y no hayan sido canceladas, con los siguientes campos: <ul style="list-style-type: none"> • Numero de factura • Proveedor • Fecha Pago 1 – Descuento • Fecha Pago 2 – Descuento • Fecha Final Pago • Estado (Pendiente – Pagada -- Todas) • Valor Real • Valor Pagado <p>Filtros:</p> <ul style="list-style-type: none"> • Numero de Factura • Proveedor • Fecha de pago • Estado <ol style="list-style-type: none"> 3. El actor selecciona una cuenta del listado 4. El sistema muestra el detalle de la factura con la opción de registrar el pago de esta si se encuentra en estado pendiente

Flujo Alternativo Filtro de datos	Este flujo comienza cuando el actor cambia uno de los filtros de datos <ol style="list-style-type: none"> 1. El actor modifica uno de los filtros del reporte 2. El sistema muestra la información filtrada 3. Se retoma el paso 2 de la secuencia normal.
Flujo Alternativo Exportar	Este flujo comienza cuando el actor decide exportar los datos a Excel después de ingresar la identificación del cliente <ol style="list-style-type: none"> 2. Se ejecuta el Caso de Uso Exportar Crédito Cliente
Flujo Alternativo Consultar Pagos	Este flujo comienza cuando el actor selecciona un crédito del listado <ol style="list-style-type: none"> 2. Se ejecuta el Caso de Uso Listar Historial de Pagos de Cliente
Flujo Alternativo Reportar Pago de Cuenta X Pagar	Este flujo comienza cuando el actor ingresa la información de pago de una cuenta pendiente del listado <ol style="list-style-type: none"> 1. El sistema ingresa la información de pago de la cuenta. 2. El sistema calcula la fecha de pago y el descuento obtenido en la fecha 3. El sistema actualiza el estado de la cuenta e ingresa el valor pagado.

Fuente: Elaboración propia.

5.2.7.8 Exportar listado de cuentas por pagar

Tabla 38. Exportar listado de cuentas por pagar.

Descripción:	Por medio de este, el sistema exporta la información de las cuentas por pagar en estado pendiente o canceladas
Frecuencia:	Alta
Importancia	Muy Alta
Urgencia	Urgente

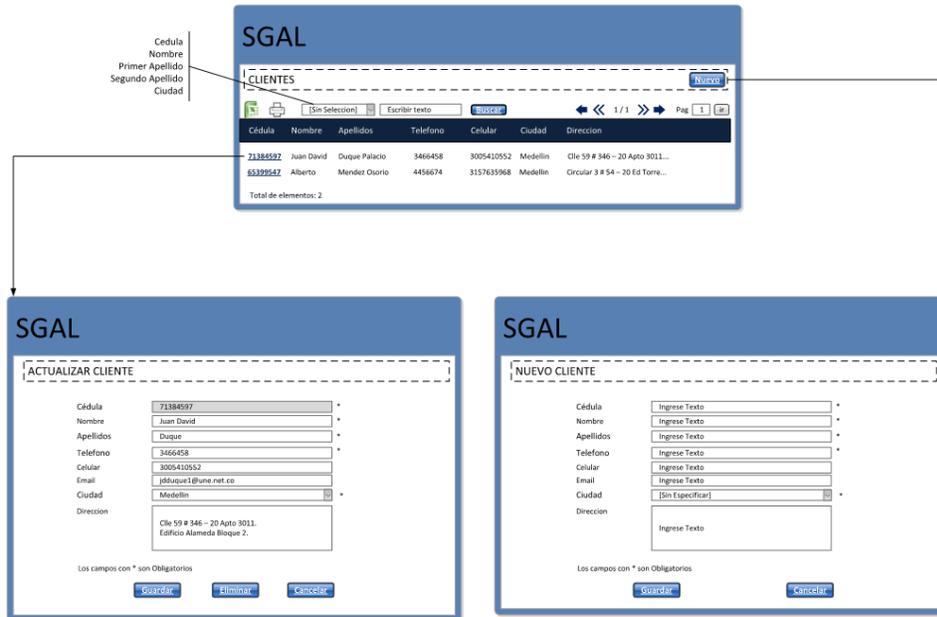
Estabilidad	Alta
Actor Primario	Sistema
Actores Secundarios	N/A
Pre-Condiciones	El actor debe seleccionar un crédito.
Post-Condiciones	N/A
Secuencia Normal	<ol style="list-style-type: none"> 1. El sistema valida el rango de fechas y consulta todas las cuentas por pagar que se encuentren en el rango ingresado para cualquiera de las fechas de pago 2. El sistema exporta la información consultada en formato Excel 2007

Fuente: Elaboración propia.

5.3 Prototipos no funcionales

5.3.1 Administración de clientes

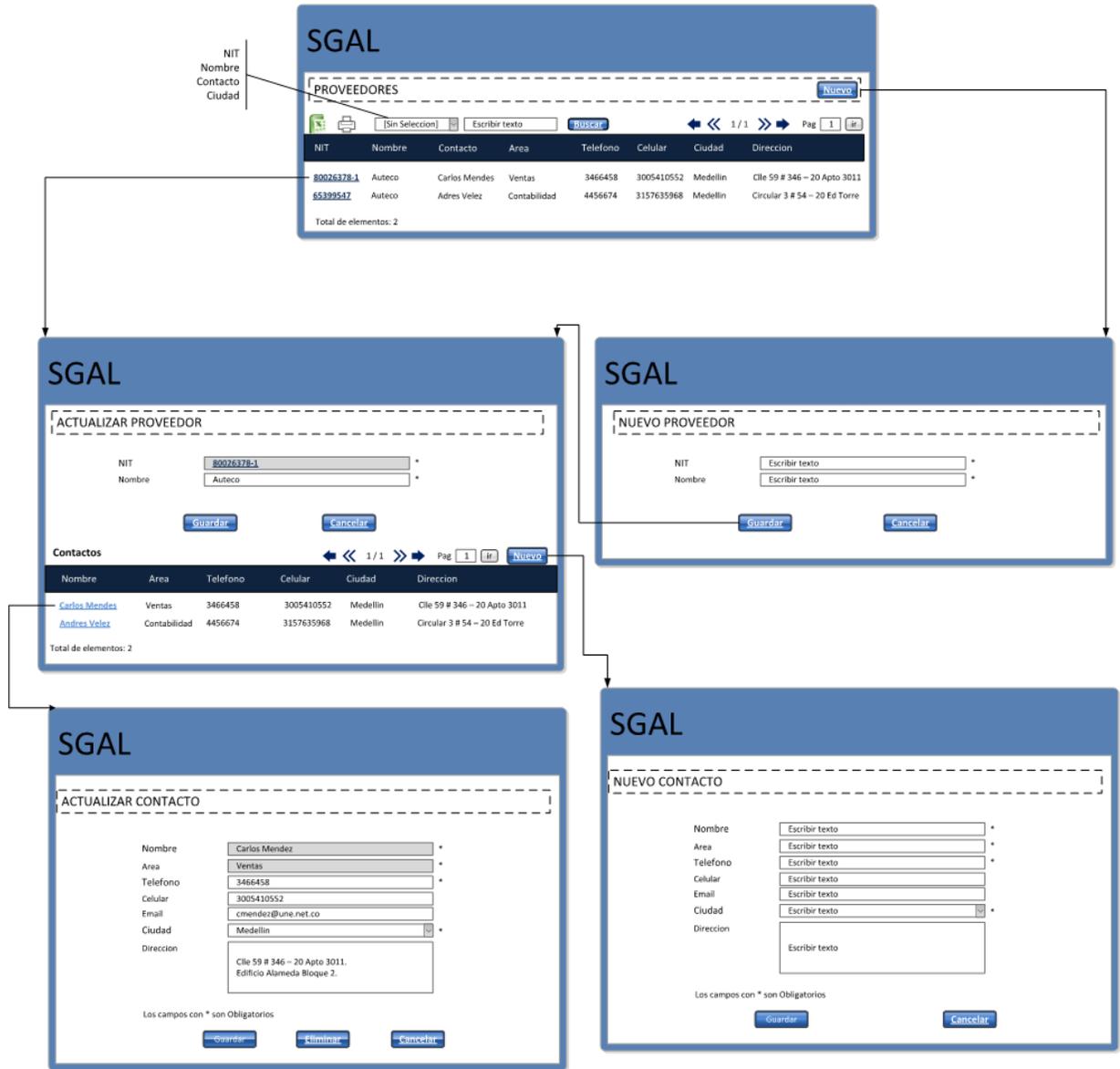
Ilustración 1 Prototipo no funcional - Administración de clientes



Fuente: Elaboración propia con base en Visio.

5.3.2 Administración de clientes

Ilustración 2 Prototipo no funcional - Administración de clientes



Fuente: Elaboración propia con base en Visio.

5.3.3 Administración de crédito

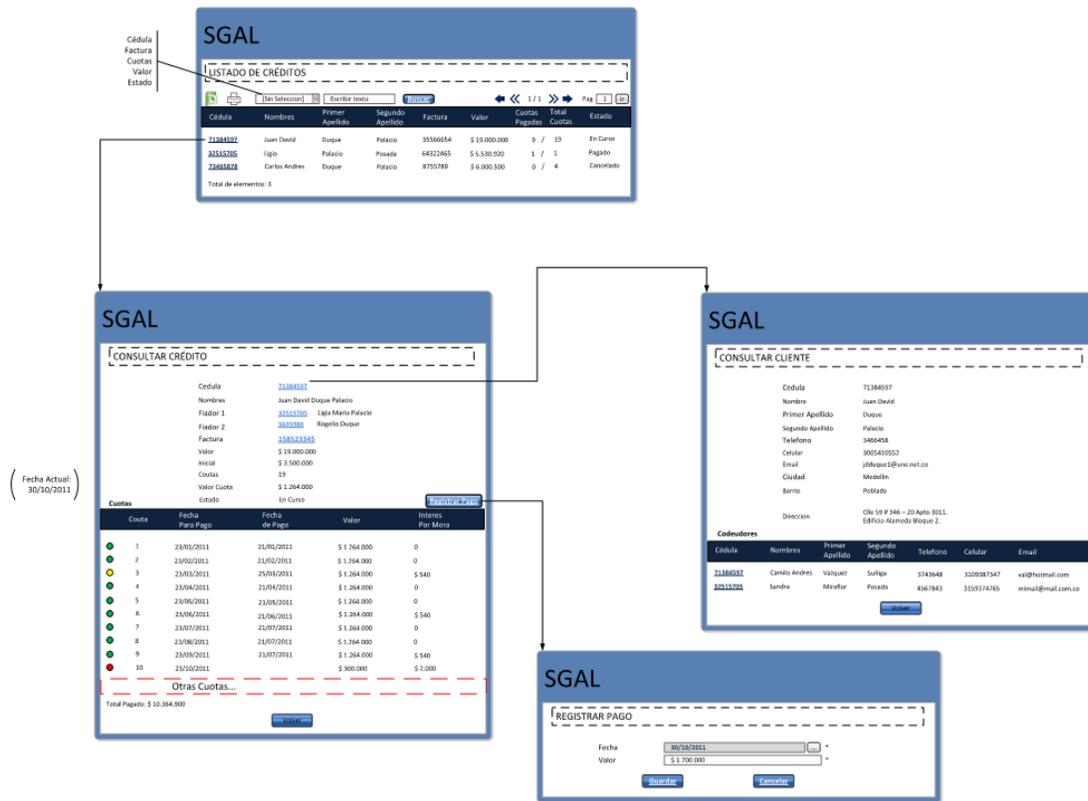
Ilustración 3 Prototipo no funcional - Administración de crédito



Fuente: Elaboración propia con base en Visio.

5.3.4 Administración de ítems del inventario

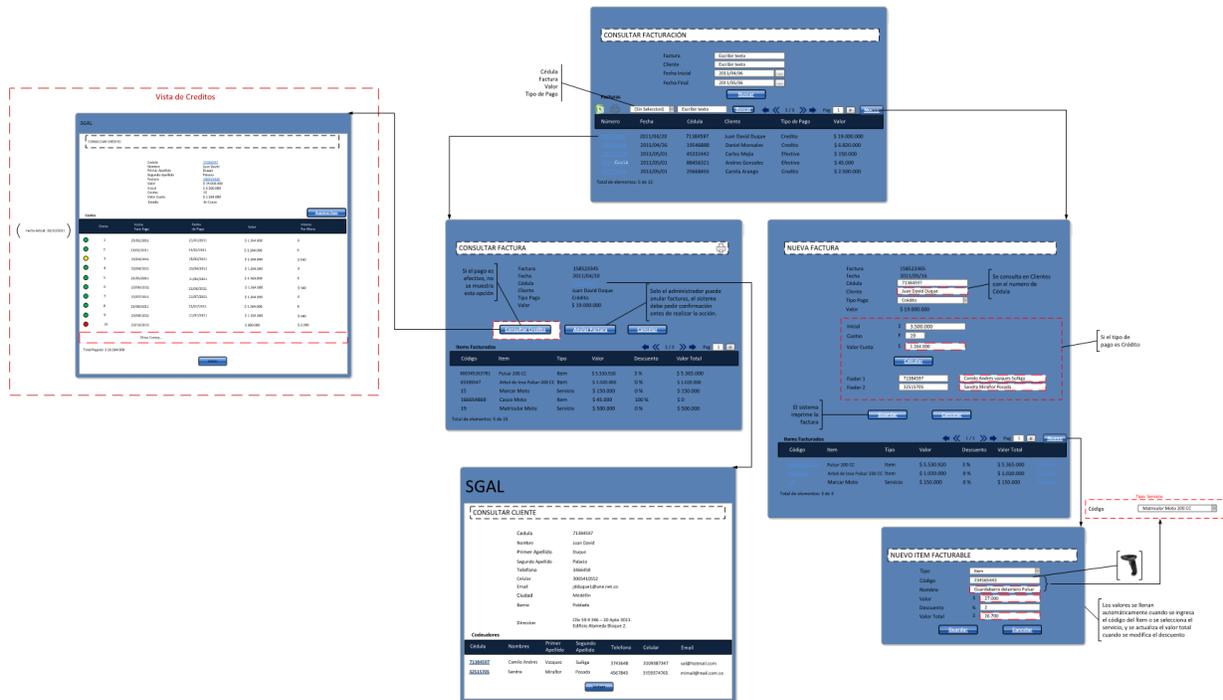
Ilustración 4 Prototipo no funcional - Administración de ítems del inventario



Fuente: Elaboración propia con base en Visio.

5.3.5 Administración de facturas

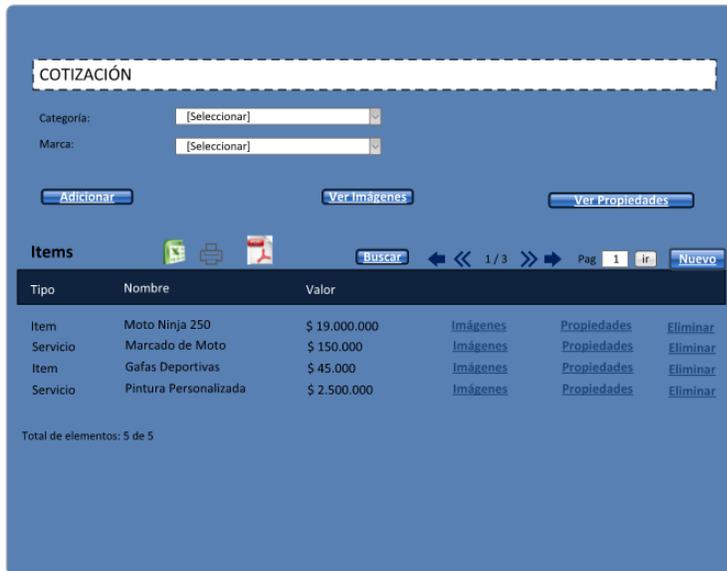
Ilustración 5 Prototipo no funcional - Administración de facturas



Fuente: Elaboración propia con base en Visio.

5.3.6 Administración de cotizaciones

Ilustración 6 Prototipo no funcional - Administración de cotizaciones



Fuente: Elaboración propia con base en Visio.

5.4 Documento de arquitectura de la solución

Este documento provee al usuario especializado una vista de la arquitectura del Sistema de SGAL.

La plantilla de este documento se basó en las especificaciones de RUP (Rational Unified Process) para el documento de arquitectura de software.

Este documento proporciona una descripción de la arquitectura del sistema, haciendo uso de diversas visiones arquitectónicas para representar diversos aspectos del sistema. Se realiza con el fin de documentar las decisiones significativas de arquitectura que se han tomado en el sistema.

5.4.1 Definiciones

Tabla 39. Definiciones.

Palabra	Definición
Paquetes	Agrupaciones de casos de uso y actores por funcionalidad que proveen.
Actor	Alguien o algo externo al sistema que interactúa con él.
Caso de Uso	Secuencia de acciones que el sistema realiza, la cual proporciona un resultado de valor observable.
MS Visio	Se refiere a la herramienta que permite realizar el modelado de los diagramas presentados en este documento.
SQL Injection	Técnica de Hacking que compromete la seguridad del sistema a través de campos de entrada que van directamente a consultas SQL, y no se les hace ningún tratamiento de seguridad.
Round trip	Se refiere al proceso de hacer la petición desde el cliente, su llegada al servidor, procesamiento de esta y la llegada de la respuesta al cliente.
ADO.NET	Parte de la plataforma Microsoft .NET que permite el acceso a diversas bases de datos y otras fuentes de datos.
User control	Tipo de control de .NET que permite encapsular funcionalidades que incluyen interfaz gráfica (a diferencia de los componentes), permitiendo reutilización y simplicidad en las páginas que los usan.
Microsoft AJAX	Sub-framework contenido en el .NET framework, que facilita que las páginas se puedan actualizar parcialmente, sin el impacto en rendimiento y experiencia de usuario que tienen las actualizaciones complejas de página cuando sólo cambia una parte de esta
AJAX Update panel	Componente de AJAX que demarca la zona que se refrescará parcialmente.

Servicio de Windows	Aplicación especialmente diseñada para que se ejecute en background en el servidor, sin interfaz de usuarios, y que es usada para ejecutar tareas periódicas y/o de procesamiento intensivo, que no pueden realizarse en la aplicación web.
Enumeracion es en .NET	Tipo especial de estructura, donde se asignan valores a literales de texto, ayudando a evitar el uso de constantes numéricas explícitas en el código.
Master page	Implementación de .NET que permite reutilización gráfica y de código, permitiendo uniformidad y centralización.
Datatable	Objeto de .NET que representa una tabla en memoria.

Fuente: Elaboración propia.

5.4.2 Acrónimos

Tabla 40. Acrónimo.

Sigla	Significado
RUP	Rational Unified Process
UML	Unified Modeling Language
BD	Base de Datos
CRUD	Create, Read, Update, Delete

Fuente: Elaboración propia.

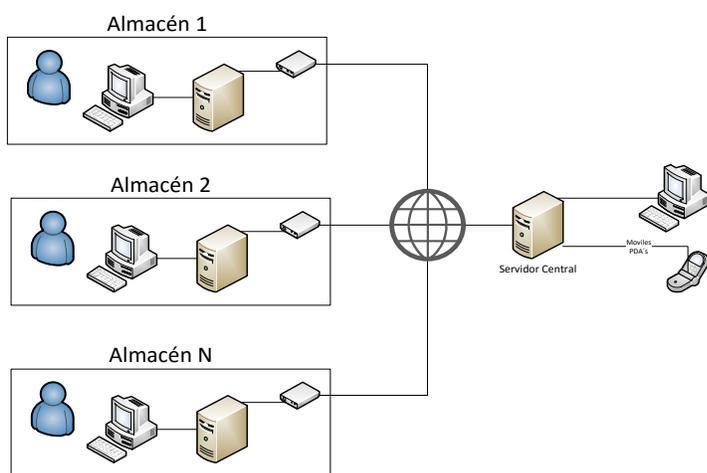
5.4.3 Arquitectura Base

El sistema está diseñado para trabajar de manera centralizada, sin embargo, la región en la que se encuentra cada almacén es de poca accesibilidad y sobretodo poca disponibilidad del servicio de internet. Por este motivo, se ha decidido implementar una solución local en cada almacén, en la que la parte de operación sea manejada localmente y la parte de reportes e

informes se maneje de manera global sincronizando la información a través de un servicio instalado en cada almacén, este servicio será el encargado de permitir visualizar informes globales de operación de los almacenes en conjunto o en particular.

En la siguiente imagen se ilustra como:

Ilustración 7. Arquitectura base.



Fuente: Elaboración propia.

5.4.4 Representación arquitectónica

A continuación se muestra la arquitectura los distintos subsistemas que componen la aplicación, ésta está dividida en tres capas:

- Capa de Presentación
- Capa de Lógica de Negocio
- Capa de Acceso a Datos

El documento se ha estructurado empleando la representación de la arquitectura de acuerdo con el modelo “4 + 1 vistas” propuesto por IBM Rational. La representación se realizará a fin de mostrar diferentes perspectivas del producto software, empleando las vistas siguientes:

- **Perspectiva Funcional** – Vista de Casos de Usos. Presenta la arquitectura desde la perspectiva del usuario final. Esta vista se desarrolla a través del Modelo de Casos de Uso (usando Diagramas de Casos de Uso de UML).
- **Perspectiva Estructural** – Vista Lógica. Presenta la arquitectura desde la perspectiva del desarrollador. Permite mostrar la organización de las piezas fundamentales de la arquitectura, organizando los elementos de diseño (clases, tablas, etc.).
- **Perspectiva de Construcción** – Vista de Implementación. Presenta la arquitectura desde la perspectiva del programador, definiendo los componentes software a ser desarrollados, la distribución de las clases, tablas y demás.

- **Perspectiva Dinámica** – Vista de los Procesos / Tareas. Presenta la arquitectura desde la perspectiva del desarrollador a fin de definir aspectos de concurrencia, comunicación entre procesos, sincronizaciones, etc.
- **Perspectiva del Despliegue** – Vista de Despliegue. Presenta la arquitectura desde la perspectiva del implantador de la solución. Define como los componentes de la arquitectura serán desplegados sobre la infraestructura de TI definida.
- **Perspectiva de los Datos** – Vista de Datos (Opcional). Presenta la arquitectura de datos que soportará los requisitos de información del sistema software. Se emplea el Modelo Entidad Relación.

5.4.5 Restricciones especiales

- La aplicación deberá desarrollarse en .NET 4.0
- La aplicación deberá desarrollarse para SQL Server 2014
- La aplicación debe usar la suite de controles ASP.Net AJAX.

5.4.6 Herramientas de desarrollo

- Se usará el entorno de desarrollo Visual Studio 2013 profesional.

5.4.7 Tecnología empleada en el desarrollo

- El sistema usará el motor de base de datos SQL Server 2014R2.
- La aplicación web y el servicio de Windows se desarrollará para la plataforma .NET Versión 4.
- El acceso a datos se realizara con ADO.NET con proveedor nativo.
- Para la interfaz gráfica se implementará con ASP.Net
- Se usarán algunos controles de Microsoft AJAX control Toolkit.

5.4.8 Interoperabilidad

No se identifica la necesidad de exponer vistas o servicios de información para otros sistemas

5.4.9 Necesidades de parte del cliente

Los datos deben estar descentralizados, y debe haber un proceso central que unifique los datos para la generación de informes a los socios de diversos almacenes.

5.4.10 Otras decisiones arquitectónicas

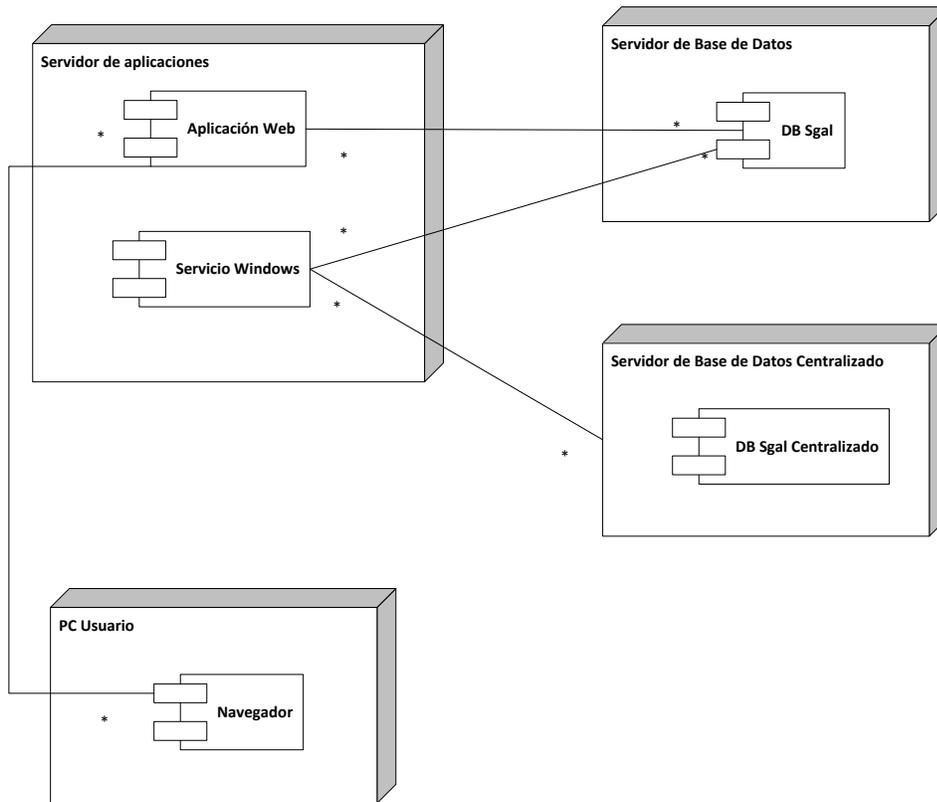
5.4.10.1 Mantenibilidad de datos

Para dar mejor rendimiento a la aplicación, se tendrán los datos, en la base de datos de la aplicación local, y un proceso de sincronización centralizará todos los datos relevantes.

5.4.11 Vista de despliegue

La vista despliegue se presenta en este punto del documento con el objetivo de mostrar globalmente los subsistemas que componen la solución, así como los otros sistemas con los que interactúan.

Ilustración 8. Vista de despliegue.



Fuente: Elaboración propia.

5.4.12 Nodos y sistemas

A continuación se describen brevemente los sistemas y subsistemas que componen la aplicación, con el fin de dar una visión global al usuario, más adelante en este documento se profundizará en cada uno de ellos.

5.4.12.1 Servidor de Base de Datos (SQL Server 2008 R2):

Es el repositorio principal y el que contendrá la base de datos de la solución.

5.4.12.2 Servidor de aplicaciones (Windows server 2008):

Este servidor alojará todas las aplicaciones de servidor que componen la solución SGAL, que son:

- Sitio web, encargado de aplicación SGAL, con todas sus funcionalidades
- Servicio de Windows: encargado de realizar tareas periódicas y/o de procesamiento intensivo.

5.4.12.3 PC usuarios

PC de los usuarios del sistema Web, este permite la administración de los datos y registro de facturas, entre otras.

5.4.13 Identificación y descripción de modelos y módulos

5.4.13.1 Módulo web

Esta arquitectura presenta una implementación de capas, y cada capa es una librería de clases, con el objetivo de compartir capas subyacentes con diferentes proyectos de front-end, por ejemplo, con servicios de Windows, servicios web o aplicaciones de escritorio.

5.4.13.2 Descripción de las capas:

5.4.13.2.1 Capa de datos

- a. Permite abstracción del motor a las capas superiores.
- b. Permite implementar soporte fácilmente para otros motores de base de datos.
- c. Implementa paginado en servidor de base de datos, con el fin de evitar traer innecesariamente gran cantidad de registros al servidor de aplicaciones.
- d. Se usa exclusivamente procedimientos almacenados.
- e. Uso de proveedor de acceso a datos nativo, por rendimiento y mejores características.

5.4.13.2.2 Capa de reglas de negocio

- a. Coordina las transacciones, sin perder abstracción del motor de base de datos.

- b. Mantenimiento de trazabilidad de las operaciones y de los errores, llevando esta información a base de datos.
- c. Coordina el manejo estructurado de excepciones.
- d. Separación entre métodos y propiedades de los objetos de negocio, llevando las propiedades a una capa transversal llamada “objetos de transferencia”, con el objetivo de organizar y optimizar.

5.4.13.2.3 Capa de presentación

- a. Se hace uso intensivo de MS AJAX, con el fin mejorar el rendimiento, la disminuir la cantidad de “round trips” completos al servidor y mejorar la experiencia de usuario.
- b. Los popups modales se hacen con ayuda de AJAX, según las nuevas tendencias.
- c. Se mantiene uniformidad en el diseño y en la forma de hacer las cosas, con el fin de hacer más intuitivo el uso de la aplicación.

5.4.13.2.4 Otras capas

Capa de objetos de transferencia: como se describe en la sección 5.2.2, estos objetos contienen las propiedades de los objetos de negocio, y sirven para transferir valores entre las capas.

Capa de funcionalidades comunes: Tiene funciones simples y reutilizables, también contiene enumeraciones generales, las cuales se usan intensivamente para ordenar el código y facilitar revisión y mantenimiento.

5.4.13.3 Autenticación:

El modelo de autenticación permite la autenticación por base de datos.

Una vez el usuario es autenticado, se busca en la base de datos de la aplicación, que esté autorizado para ingresar, y se le asigna el perfil correspondiente dependiendo de la configuración del usuario que ingresa a la aplicación.

5.4.13.4 Autorización:

Los perfiles son configurables a nivel de recurso (página / funcionalidad), y acción que puede realizar sobre esta (CRUD).

La aplicación web, y en general las aplicaciones de front-end son las encargadas de verificar la autorización del usuario. En el caso de las aplicaciones web, esta funcionalidad se implementa en la masterpage, y cada página se presenta ante esta y le pregunta por los permisos del usuario.

5.4.13.5 Servicio de Windows

Este servicio es necesario para ejecutar ciertas tareas que deben ejecutarse periódicamente, a una hora específica y/o por su complejidad, no pueden ejecutarse en línea en la aplicación web.

Se implementarán 3 tipos de tareas:

- Ejecutan periódicamente, como es el caso de la revisión de inconsistencias en registros de programación.
- Ejecución determinados días y determinadas horas.
- Ejecución programada manualmente, y que generalmente reciben parámetros de quien la programa, por ejemplo, envío a costos y a facturación.

La aplicación web expondrá una interfaz gráfica que permitirá monitorear y controlar las tareas que ejecuta el servicio.

Algunas tareas en las que no se requiere mucha precisión en la hora de ejecución, se ejecutarán en cola, en un mismo hilo. Las tareas que tienen prioridad, se ejecutarán en su propio hilo, pero estas se disminuirán al máximo, para prevenir que sobrecarguen el servidor de aplicaciones.

En el documento de diseño detallado se encuentran los detalles de cada una de las operaciones del servicio.

5.4.14 Vista lógica

5.4.14.1 Paquetes de diseño importantes arquitectónicamente

5.4.14.1.1 Inventario

Se debe controlar el inventario en Stock generando un código único que no exista en la base de datos, este código será un encriptado numérico que se pueda generar en un código de barras, se generará con la fecha actual incluyendo el usuario que lo genera y el código de la aplicación.

5.4.14.1.2 Facturación

Para la facturación se debe involucrar el inventario, las cotizaciones y la generación de créditos y cuotas, todo esto se debe hacer en una sola transacción de la BD, para que en caso de que falle, se pueda recuperar el sistema y no queden datos inconsistentes.

5.4.14.2 Control de concurrencia

Para el control de concurrencia en acceso a registros, cuando no hay interacción de usuarios durante el proceso de modificación, se hará con el método estándar de transacciones, usando el nivel de aislamiento apropiado para evitar bloquear más registros de los necesarios.

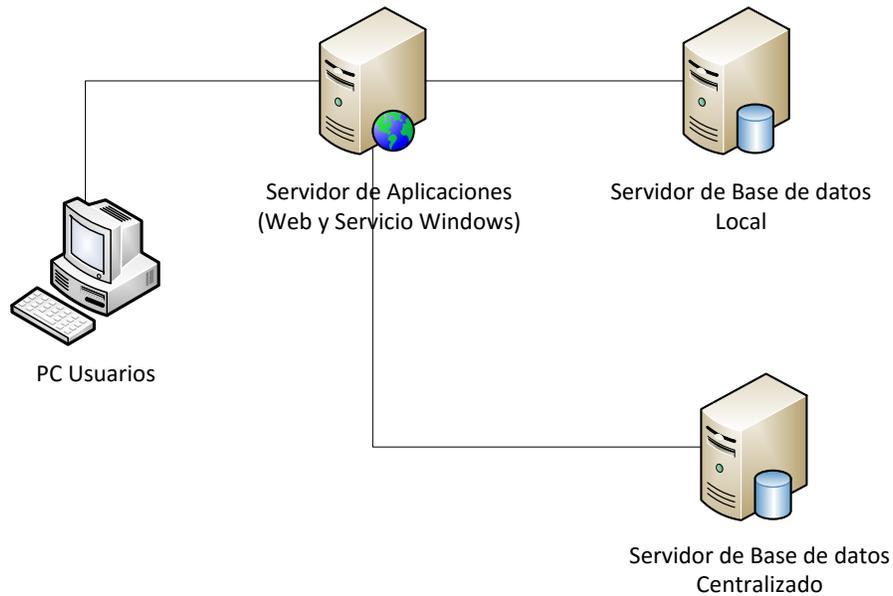
5.4.14.3 Sincronización

Los procesos de sincronización ya están definidos en las secciones 4.1.3 y 3.8.2

5.4.14.4 Vista de implementación

Como complemento al diagrama despliegue, el diagrama de infraestructura tecnológica muestra los equipos involucrados en la aplicación, de una manera más natural.

Ilustración 9. Vista de implementación.



Fuente: Elaboración propia.

5.4.14.5 *Perspectiva general*

La implementación se realizará siguiendo las mejores practicas de arquitectura. La arquitectura está basada en el patrón multicapa, con una capa de presentación, una de reglas de negocio y otra de acceso datos; con unas funciones comunes a todas las capas, lo mismo que los objetos de transferencia de información. Así mismo, esta arquitectura.

5.4.14.6 Capas

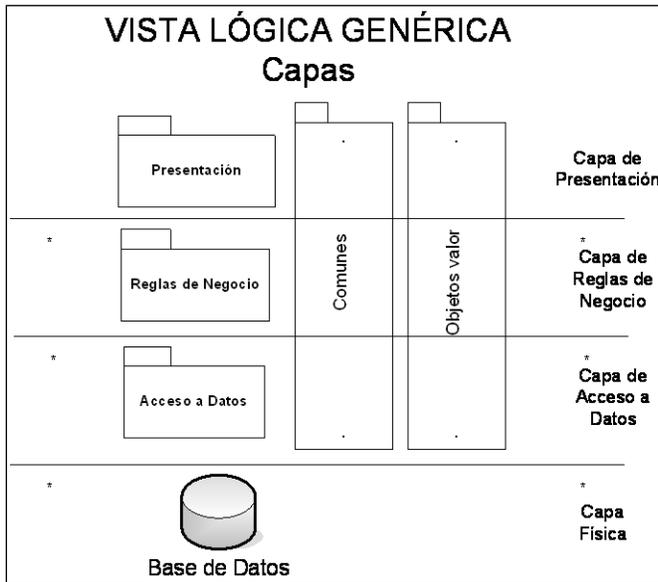
La capa de presentación es sólo para recibir información, realizar validaciones básicas de datos, y visualizar resultados. No debe haber ninguna regla de negocio, proceso o similar en esta capa. En ningún momento esta capa puede hacer llamados directos a acceso a datos, de hecho, no debe tener directamente referenciada esta librería.

La capa de reglas de negocio debe contener todas las reglas, procesos, manejo de transacciones, etc. Haciendo uso de la capa de datos como simple acceso a la base de datos. En ningún momento, esta capa puede hacer invocaciones a la capa de presentación.

La capa de acceso a datos permite el acceso a la base de datos, realizando operaciones básicas tipo CRUD; en esta capa o en los procedimientos almacenados NO se debe realizar procesos complejos, ni transacciones, mientras no se indique explícitamente en el presente documento. Las transacciones se deben controlar desde la capa de reglas de negocio, según se presenta en la arquitectura de referencia.

En la siguiente figura podemos observar la distribución de los paquetes en cada una de las capas del sistema:

Tabla 41. Vista lógica.



Fuente: Elaboración propia.

5.4.15 Orden de implementación

Se deberán implementar las administraciones en primer lugar, luego se debe continuar con el registro de créditos e información de cuotas para el manejo de cuotas vencidas de clientes.

A continuación se debe implementar el manejo de las facturas con los rangos de facturación, la generación de totales y las impresiones necesarias. Luego se debe continuar con el inventario, permitiendo registrar y retirar elementos y descontarlos desde la facturación.

Por último, se debe implementar el módulo cuantas por pagar y de reportes, que no presentan gran riesgo en la aplicación por el bajo manejo de datos sensibles.

5.4.16 Vista de datos

En el documento de diseño detallado se encuentra toda la vista de datos: modelo de datos y diccionario de datos.

5.4.16.1 Tamaño y rendimiento

- La aplicación deberá realizar cualquier operación transaccional (No informes ni procesos pesados) en un tiempo igual o inferior a 3 segundos, siempre que se cumplan los requisitos técnicos, en especial el de disponibilidad de ancho de banda y latencia inferior a 200ms para un paquete de 1KB.
- Aunque la aplicación no tendrá un gran volumen de usuarios, si realiza operaciones complejas, especialmente a nivel de base de datos, también manejará un volumen considerable de información, por lo que es necesario hacer énfasis en la optimización de consultas e indexación.

5.4.17 Calidad

5.4.17.1 Plan de back ups y recuperación de desastres

El plan de backup y recuperación de desastres es una tarea que en su mayoría recae en el área de TI del cliente, sin embargo, cada implantación proveerá una funcionalidad de recuperación de datos.

Para mantener controlado el tamaño de la base de datos, se recomienda al cliente realizar un proceso periódico de archivar información antigua de la tabla de log.

El detalle de los demás componentes de la aplicación que se deben respaldar, como carpetas de aplicación y de archivos.

5.4.17.2 Atributos de calidad

5.4.17.2.1 Usabilidad

Desde la arquitectura de referencia del sitio web, sección 5.2.1.3, se muestran varias implementaciones encaminadas a que la aplicación sea amigable para el usuario, y que le permita ser eficiente.

Las acciones encaminadas a mejorar el rendimiento (sección 13.2.2) también aportan bastante en usabilidad.

5.4.17.2.2 Rendimiento

En el diseño de las funcionalidades se tuvo muy en cuenta el rendimiento, considerando transmitir el mínimo de información necesaria hacia los clientes y los servidores de aplicaciones.

También en el diseño de la base de datos se tuvo en cuenta que las consultas necesarias para atender solicitudes de operaciones exigentes, como de mando y clientes Silverlight y mobile, no fueran muy exigentes para el servidor, incluso en algunos casos se procedió a desnormalizar para conseguir mejor rendimiento.

En etapas tempranas del desarrollo se harán pruebas de rendimiento en el ambiente de pruebas del cliente para verificar si se materializan estos resultados teóricos, de no ser así, se procederá a afinar índices, consultas y posiblemente desnormalizar más para conseguir el rendimiento necesario.

Es altamente recomendado que se active la compresión de contenido dinámico para el sitio web ver sección 5.4.3, pues el costo a nivel de procesador es mínimo y se gana bastante en ahorro de canal y por lo tanto mejoran los tiempos de respuesta.

5.4.17.2.3 Mantenibilidad

El uso de capas, de patrones, la separación de funcionalidades, la documentación de arquitectura y diseño y la coherencia en el desarrollo lograda con la arquitectura, hace que sea fácil hacer mantenimiento y agregar o cambiar funcionalidades.

La aplicación será desarrollada utilizando las versiones más recientes de las últimas, lo que permite tener una vida útil de la aplicación mucho más larga.

Las tecnologías se usan para lo que fueron diseñadas, sin forzar implementaciones, teniendo un modelo de desarrollo limpio, lo que facilita la actualización a versiones más recientes de las tecnologías, cuando estas lleguen.

5.4.17.2.4 Seguridad

Además de los mecanismos de autenticación y autorización que se presentan en la arquitectura de referencia; la aplicación está preparada para soportar SSL, que permite que la información que viaja entre el servidor y el cliente esté encriptada.

Como el sistema soporta seguridad integrada y autenticación por base de datos, las credenciales de dominio pueden viajar por la red en texto plano, por lo que se recomienda al cliente evaluar el riesgo que esto implica y considerar el uso de SSL en el sitio web.

Para el web service de mobile no es tan importante ni es sugerido, pues los usuarios de esta aplicación no usarán cuentas de dominio, y el SSL castiga mucho el rendimiento.

Se debe tener en cuenta que el uso de SSL incrementa el uso de procesador en el servidor de aplicaciones aumenta el tráfico de red.

Como mecanismos reactivos, la aplicación contará con un sistema de bitácora que almacenará información de quién hizo qué (CRUD), cuando y desde cuál equipo (IP).

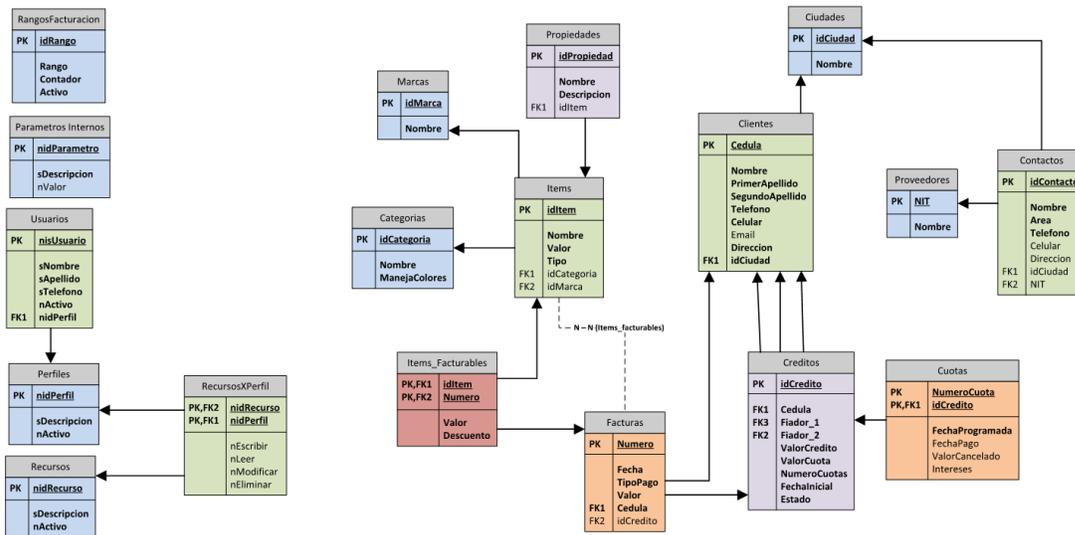
5.5 Documento de Diseño detallado de la solución

Esta sección, junto con la de arquitectura, proveen una visión del diseño de la arquitectura del Sistema *SGAL* de modo que facilite su desarrollo y posterior mantenimiento; está orientado principalmente a desarrolladores, arquitectos y soporte técnico.

Esta sección está diseñada para ser usada como referencia principal en la etapa de desarrollo de la aplicación, aquí se podrán encontrar diagramas del sistema como el diagrama de clases, diagrama de actividades y modelo de datos, así como cada uno de sus prototipos con la referencia del(los) caso de uso que cumple, y los lineamientos de desarrollo importantes a tener en cuenta durante su implementación

5.5.1 Diagrama de datos

Ilustración 10. Diagrama de datos.



Fuente: Elaboración propia.

5.5.2 Diseño de módulos/subsistemas

5.5.2.1 Autenticación de usuarios

La autenticación de usuarios para la aplicación Confiabilidad en el suministro de materiales (CSM) se hará a través del módulo principal que es la aplicación Calidad y conformidad de plantas (AD), al igual que se hace para Evaluación sensorial. Para ello se debe hacer lo siguiente:

5.5.2.2 *Módulo de Administración*

5.5.2.2.1 Administrar clientes

CU Administrar Clientes implementado completamente

CU Administrar codeudores implementado completamente

Lineamientos de implementación:

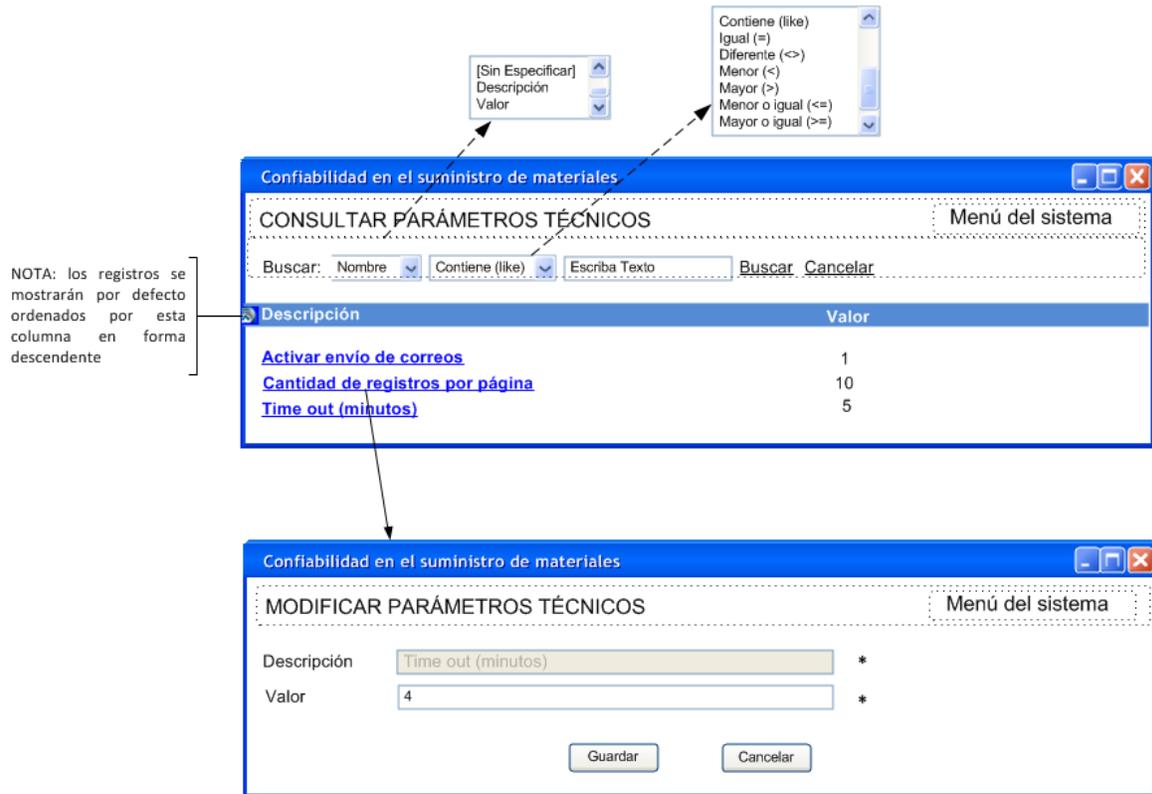
- Listado
 - Crear el acceso a datos de la tabla tbClientes
 - Crear la página Clientes.aspx
 - Se deben listar los registros de la tabla de clientes mostrando las columnas que se visualizan en el prototipo.
 - Se debe mostrar el buscador permitiendo consultar por los campos de nombre, cédula, apellidos y ciudad
 - Se debe mostrar el UCPaginador si la cantidad de registros es mayor a la especificada en el parámetro interno “Cantidad de registros para el listado”.
 - Se debe crear un link a la pagina ClientesActualizar.aspx en cada registro (campo cédula), pasando como parámetro por el QueryString el Id de cada registro (parámetro

Key=ID), además pasar los filtros que se tienen para no perderlos cuando se retorne de la página de actualizar.

- Actualizar:
 - Se debe crear la página ClientesActualizar.aspx
 - Recibir el parámetro Key con el ID del cliente. Si este no se encuentra, establecer el ID en 0 y tratar el registro como nuevo.
 - Si el registro existe, cargar la información de este permitiendo modificar todos los datos excepto la cédula.
 - Al guardar el registro del cliente ya existente, se debe validar que se hayan ingresado los campos obligatorios. Una vez guardado, se debe redireccionar a la página del listado conservando los filtros que se tenían.
 - Si el registro es nuevo, entonces se deben mostrar habilitados todos los campos, y cuando el usuario guarde, se debe validar que todos los datos obligatorios hayan sido ingresados, y que no exista otro registro de clientes con el mismo número de cédula ingresado. Una vez se guarde el registro, se debe redireccionar a página del listado.

5.5.2.2.2 Administrar parámetros

Ilustración 11. Administrar parámetros.



Fuente: Elaboración propia.

CU Administrar parámetros implementado completamente

Lineamientos de implementación:

- Listado
 - Se debe crear la página Parametros.aspx

- Crear el acceso a datos de las tablas tbParametros, tbParametrosInternos
 - Se deben listar los registros de la tabla tbParametros, con la descripción y el valor de cada uno de los registros.
 - Se debe crear un link a la pagina ParametrosActualizar.aspx en cada descripción del listado, pasando como parámetro por el QueryString el Id de cada registro (parámetro Key=ID), además pasar los filtros que se tienen para no perderlos cuando se retorne de la página de actualizar.
 - Se debe mostrar el UCBuscador y permitir buscar por la descripción del parámetro
 - Se debe mostrar el UCPaginador si la cantidad de registros es mayor a la especificada en el parámetro interno “Cantidad de registros para el listado”.
-
- Administración
 - Crear la página ParametrosActualizar.aspx
 - Recibir el parámetro Key del QueryString y buscar con ese ID en la tabla tbParametros. En caso de no recibir el parámetro, entonces lanzar una excepción como error.
 - Validar según el parámetro que se actualiza, que el valor esté en los rangos aceptados, y que sea del tipo de dato que debe recibir, esto es, si el parámetro es un número entero, entonces que esté entre 1 y 255, que si es Si o No, el valor sea 1 u otro diferente, y que si es texto, sea diferente de vacío.

5.5.2.2.3 Administrar ítems

CU Administrar parámetros implementado completamente

Lineamientos de implementación:

- Listado:
 - Se debe crear la página Items.aspx
 - Crear el acceso a datos de las tablas tbItems, tbPropiedades, tbMarcas y tbCategorías.
 - Se deben listar los registros de la tabla tbitems donde el tipo sea = enumTipoItem.Item, como se muestra en el prototipo, adicionando un link en el listado en la columna código que lleve a la página ItemsActualizar, pasando como parámetro en el QueryString el Id del registro (Parámetro Key = ID).
 - Se debe mostrar el UCBuscador y permitir buscar por el nombre, la categoría, la marca y el código de los ítems
 - Se debe mostrar el UCPaginador si la cantidad de registros es mayor a la especificada en el parámetro interno “Cantidad de registros para el listado”.

- Administración:
 - Se debe crear la página ItemsActualizar.aspx
 - Recibir el parámetro Key con el ID del ítem. Si este no se encuentra, establecer el ID en 0 y tratar el registro como nuevo.

- Si el registro existe, listar el encabezado de este (Código, categoría, marca, nombre, valor y cantidad) de manera sololectura y listar las propiedades asociadas al código del ítem listado, este listado debe tener paginador en caso de que el número de registros sea mayor al parámetro interno “Cantidad de registros para el listado”.
- Adicionar en el listado de propiedades un link en el nombre, que por medio del RowCommand abra un popup que permita modificar la propiedad seleccionada, además, cuando se presione el botón de nueva propiedad, se abra este mismo popup y permita crear una nueva propiedad asociada al ítem. El popup debe ser Modal.
- Al guardar el registro del ítem ya existente, se debe validar que se hayan ingresado los campos obligatorios y que el valor sea mayor que 0. Una vez guardado, se debe redireccionar a la página del listado conservando los filtros que se tenían.
- Si el registro es nuevo, entonces se deben mostrar habilitados los campos del encabezado, y cuando el usuario guarde, se debe validar que todos los datos obligatorios hayan sido ingresados, que el valor sea mayor que 0 (excepto el valor del porcentaje de IVA), y que no exista otro registro de ítems con el mismo código ingresado. Una vez se guarde el registro, se debe redireccionar a la misma página con el nuevo ID como parámetro del QueryString para que el registro abra con la posibilidad de ingresar nuevas propiedades.

5.5.2.2.4 Administrar servicios

CU Administrar parámetros implementado completamente

Lineamientos de implementación:

- Listado:
 - Se debe crear la página Servicios.aspx
 - Se deben listar los servicios registrados de la tabla ítems donde el tipo sea = enumTipoItem.Servicio
 - Adicionar un link a cada registro y manejarlo por el evento de RowCommand abriendo un popup que permita modificar el servicio seleccionado, al presionar el botón nuevo, se abrirá el mismo popup para permitir crear un nuevo servicio en la aplicación
 - Al crear un nuevo servicio, se debe establecer los datos de código (autonumérico para los servicios), nombre y valor mayor a 0, los demás datos se deben establecer en 0.
 - El código del servicio será controlado por un parámetro interno “Consecutivo códigos de servicios”, este consecutivo se debe tomar y aumentar en 1 en una transacción, cuando se cree el registro, se actualiza el parámetro.

5.5.2.2.5 Administrar parámetros funcionales

Ilustración 12. Administrar parámetros funcionales.

NOTA: los registros se mostrarán por defecto ordenados por esta columna en forma descendente

CONSULTAR PARÁMETROS FUNCIONALES

Buscar: Descripción [v] Contiene (like) [v] Escriba Texto [input] **Buscar** **Cancelar**

Descripción	Valor
Diferencia en altura superior inferior aceptable (mm)	15
Diferencia mínima respecto a altura del panel (mm)	10
Tiempo para enviar notificación sobre las FTM incompletas (días)	2
Tiempo máximo para generar plan de acción (días)	5
Tolerancia mínima aceptable en longitud (mm)	12

MODIFICAR PARÁMETROS FUNCIONALES

Descripción: *

Valor: *

Guardar **Cancelar**

Fuente: Elaboración propia

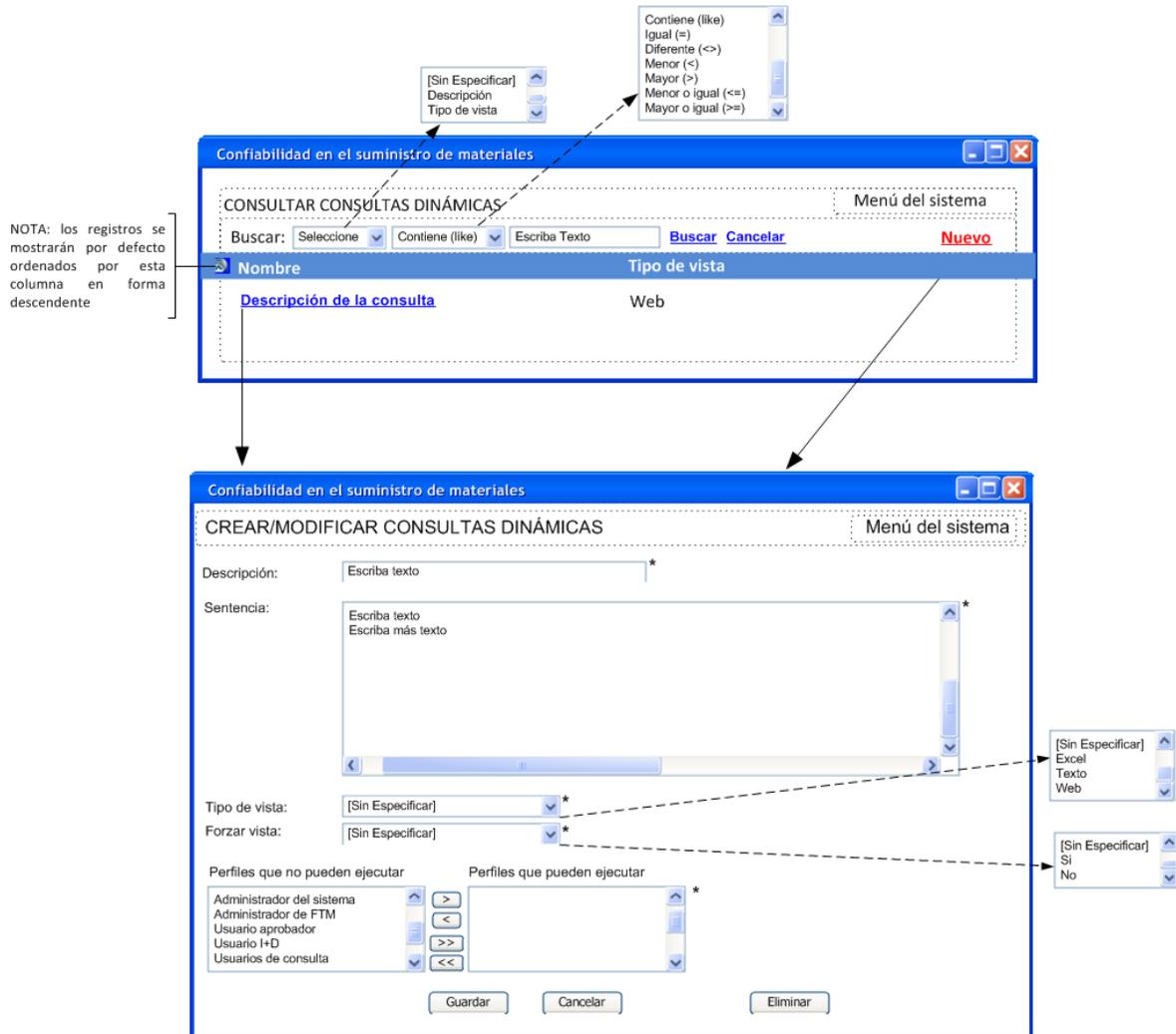
CU 0041 - Administrar parámetros funcionales implementado completamente

Lineamientos de implementación:

- Adecuar la configuración general de administración de parámetros funcionales configurada en la plantilla.
- Se deben agregar en la tabla `tbParametrosFuncionales` los registros:
- Se debe crear la enumeración *parametrosFuncionales* con los siguientes miembros:
- Validaciones:
 - El nombre del parámetro deberá ser único.
 - Cantidad mínima de ítems por sección de la plantilla: debe ser un valor numérico entero de máximo 3 dígitos
 - Tiempo para eliminar plantillas que no se pudieron publicar (días): debe ser un valor numérico entero de máximo 3 dígitos
 - Cantidad mínima de ítems por sección de la FTM: debe ser un valor numérico entero de máximo 3 dígitos
 - Tiempo máximo para publicar nuevas FTM (días): debe ser un valor numérico entero de máximo 3 dígitos

5.5.2.2.6 Administrar consultas dinámicas

Ilustración 13. Administrar consultas dinámicas.



Fuente: Elaboración propia.

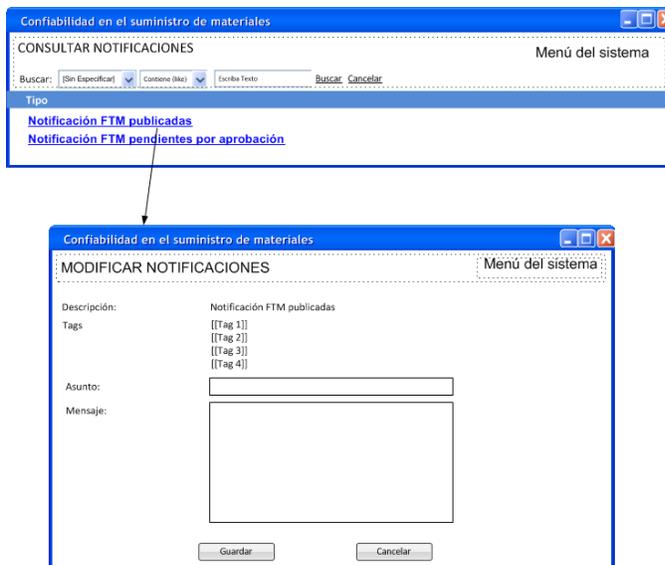
CU 0042 - Administrar consultas dinámicas implementando completamente

Lineamientos de implementación:

- Configuración general de administración de consultas dinámicas configurada en la plantilla.
- Se habilita la selección de los perfiles que pueden ejecutar la consulta.
- Los mismos parámetros deben ser verificados en la ejecución de las consultas dinámicas.
- Revisar ejemplo en plantilla.
- Validaciones:
 - El nombre de la consulta deberá ser único.

5.5.2.2.7 Administrar notificaciones

Ilustración 14. Administrar notificaciones.

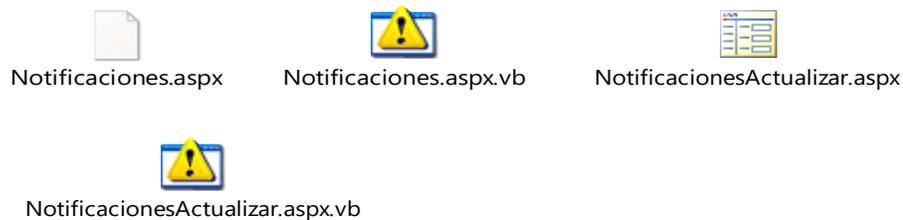


Fuente: Elaboración propia.

Lineamientos de implementación:

- Se debe generar la tabla tbNotificaciones y agregar todas las capas y el SP.
- El listado deberá mostrarse inicialmente ordenado de forma ascendente por el nombre de la notificación.
- Se deben tomar los siguiente archivos y hacer las modificaciones necesarias:

5.5.2.2.7.1 Código fuente de notificaciones.



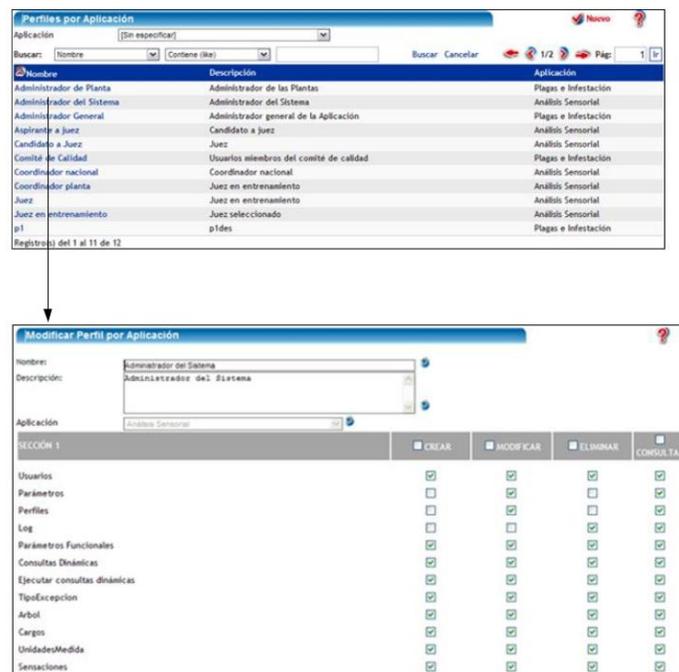
Fuente: Elaboración propia en Visual Basic.

- Se debe quitar el control ddlListada y el botón eliminar de los archivos base pasados anteriormente.
- Las notificaciones no pueden ser creadas, solo modificadas.
- Es sistema lista los datos de la notificación donde se enumeran los tags disponibles que el usuario puede utilizar para la redacción.
- El usuario puede cambiar el contenido de la modificación escribiendo el asunto y el mensaje, en ambos casos puede utilizar los tags disponibles.
- El mensaje de la notificación es un campo de texto enriquecido que soporta contenido HTML (Ver ejemplo en la plantilla)

- Crear un método que permita realizar el remplazo de los tags por los datos específicos de la notificación.
- Revisar el correcto funcionamiento al momento de adecuar los archivos de ejemplo suministrados.

5.5.2.2.8 Administrar perfiles por aplicación

Ilustración 15. Administrar perfiles por aplicación.



Fuente: Elaboración propia

CU 0039 - Administrar perfiles por aplicación implementado completamente

Lineamientos de implementación:

- Se deben listar todos los recursos de la tabla tbRecursos, con 4 checkbox que corresponden a las acciones de crear, modificar, eliminar y consultar, según se seleccionen las casillas para cada recurso, se darán los permisos para realizar acciones a los usuarios de cada perfil que ingresen al sistema.

5.5.2.3 *Módulo de cuentas por cobrar*

5.5.2.3.1 Administrar créditos

CU Administrar Créditos Implementado parcialmente

CU Administrar Cuotas Implementado Parcialmente

Lineamientos de implementación

- Listado
 - Crear el acceso a datos de las tablas tbCreditos y tbCuotas
 - Crear la página Creditos.aspx
 - Se deben listar los créditos que están en estado enumEstadoCredito.EnCurso, o los créditos cuyo estado sea diferente de este, y la fecha de actualización esté entre la fecha actual y la fecha actual menos un mes. Utilizar el siguiente query base.

Ilustración 16. Query base "Estado crédito en curso".

```
DECLARE @nidEstadoCuotaPagada as integer
DECLARE @nidEstadoCreditoEnCurso as integer

SELECT tb2.sCedula, tb2.sNombre, tb2.PrimerApellido, tb2.SegundoApellido,
tb1.NumeroFactura, tb1.Valor, tb1.NumeroCuotas,
(SELECT COUNT(nidCuota) FROM tbCuotas
WHERE nidCredito = tb1.nidCredito AND nEstado = @nidEstadoCuotaPagada) as CuotasPagadas
, tb3.sDescripcionEstado
FROM tbCreditos as tb1
INNER JOIN tbClientes as tb2 ON tb1.Cedula = tb2.Cedula
INNER JOIN tbEstadoCredito as tb3 ON tb1.nidEstado = tb3.nidEstado
WHERE tb1.nidEstado = @nidEstadoCreditoEnCurso
OR (tb1.nidEstado <> @nidEstadoCreditoEnCurso AND tb1.dFechaModificacion
BETWEEN FECHA_ACTUAL AND FECHA_ANTERIOR)
```

Fuente: Elaboración propia.

- Crear un Link a la página CreditosActualizar.aspx que pase como parámetro el ID del crédito. El link estará en la columna de cédula del cliente.
 - Se debe mostrar el UCBuscador y permitir buscar por los nombres de la persona, o por la cédula, o por el estado del crédito.
 - Se debe mostrar el UCPaginador si la cantidad de registros es mayor a la especificada en el parámetro interno “Cantidad de registros para el listado”.
- Administración:
 - Crear la página CreditosActualizar.aspx
 - Se debe recibir en el parámetro Key del QueryString el ID del crédito
 - Se debe listar la información del crédito mostrando un link en el campo de la cédula, uno en el campo de la cédula del fiador1, y otro en el campo del fiador2, estos 3 deben abrir un Popup con la información del cliente relacionada con cada cédula.

- Se debe mostrar el listado de cuotas, asociadas al crédito que se recibió en el QueryString, con un semáforo en la primera columna con 2 colores que indique el estado de la cuota de la siguiente manera:
 - Primer color Verde: Indica que la cuota ya se canceló-
 - Primer color negro: Indica que la cuota no se ha cancelado aún
 - Segundo color Verde: Indica que la fecha de cancelación de la cuota es menor que la fecha en la que se vencía, si no se ha cancelado aún, entonces se toma la fecha actual
 - Segundo color Amarillo: Indica que la fecha de cancelación de la cuota está entre la fecha fecha en la que se vencía y la fecha de vencimiento menos el parámetro “Dias alerta vencimiento de cuotas”, si no se ha cancelado aún, entonces se toma la fecha actual
 - Segundo color Rojo: Indica que la fecha de cancelación de la cuota es mayor o igual que la fecha en la que se vencía, si no se ha cancelado aún, entonces se toma la fecha actual

- La opción de registrar pago abre un PopUp que por defecto muestra la fecha actual sin la posibilidad de modificarla y el valor de la cuota, este valor se puede modificar. Al guardar, se debe validar que se haya ingresado un valor mayor que 0 y lanzar un evento RegistrarPago (ValorCuota)
 - La página debe recibir el evento del popup que ingresa el valor a cancelar y realizar lo siguiente:

Ilustración 17. Proceso para valor de cancelación.

```
Dim nValorPagado as integer = 'Valor Pagado

dim oProximaCuota as Objetos.Cuotas = RNCuotas.BuscarProxima(ID_CREDITO)
Try

    While nValorPagado > (ProximaCuota.Valor - oProximaCuota.Abonos + oProximaCuota.InteresesMora) Do
        nValorPagado = nValorPagado - (ProximaCuota.Valor - oProximaCuota.Abonos + oProximaCuota.InteresesMora)
        oProximaCuota.Estado = enumEstadoCuotas.Pagada
        oProximaCuota.FechaPago = FechaActual
        RNCuotas.Actualizar(oProximaCuota)
    End While
    If nValorPagado > 0 Then
        oProximaCuota as Objetos.Cuotas = RNCuotas.BuscarProxima(ID_CREDITO)
        oProximaCuota.Abonos = nValorPagado
        RNCuotas.Actualizar(oProximaCuota)
    End If
Catch Exception ex
    ' Capturar cuando no existan mas cuotas
End Try
```

Fuente: Elaboración propia.

- Crear el método RNCuotas.BuscarSiguiente, que busca con el ID del crédito una cuota en estado Pendiente y con menor fecha de pago programado, en caso de no encontrarla, debe lanzar una excepción que indique que el cliente no tiene más cuotas pendientes por pagar.

5.5.2.4 Módulo de facturación

5.5.2.4.1 Administrar facturas

CU Administrar Facturas Implementado completamente

CU Imprimir Detalle de Factura Implementado completamente

CU Aplicar descuentos en ítems de facturas Implementado completamente

CU Separar IVA del valor de las facturas Implementado completamente

CU Agregar servicios a las facturas Implementado completamente

Lineamientos de implementación:

- Listado
 - Se debe crear el acceso a datos de las tablas tbFacturas y tbItemsFacturables
 - Se debe crear la página Facturas.aspx
 - Los filtros de fechas son obligatorios y por defecto se cargarán en fecha final la fecha actual y en fecha inicial la fecha actual menos 1 mes
 - Buscar las facturas que cumplan con los filtros de búsqueda ingresados por el usuario.
 - Crear un link en el número de la factura que lleve a la página FacturasActualizar.aspx, pasando como parámetro en el QueryString key = ID Factura, si es nuevo registro, se debe pasar el parámetro en 0
 - Permitir exportar el listado que se consulta a Excel, al exportarlo, mostrar en el encabezado los filtros utilizados para consultar el listado de facturas.
 - Se debe mostrar el UCBuscador y permitir buscar por los nombres de la persona, o por la cédula o por el tipo de pago
 - Se debe mostrar el UCPaginador si la cantidad de registros es mayor a la especificada en el parámetro interno “Cantidad de registros para el listado”.

- Administración:

- Se debe crear la página FacturasActualizar.aspx
- Se debe buscar el ID Factura en el QueryString, y si es diferente de 0, entonces se debe buscar el registro y la información de ítems asociados a la factura en la tabla tbItemsFacturables.
- Si la factura es diferente de 0
 - La página debe abrir en modo solo lectura, ya que no se puede modificar la información de una factura creada.
 - Se debe mostrar el botón de “Anular Factura”, el cual marcará la factura en estado EnumEstadoFacutra.Anulada.
 - Se debe mostrar la opción de “Consultar Crédito”, en caso de que la factura no esté asociada a ninguno, entonces se debe mostrar deshabilitada esta opción. Este botón debe redireccionar a la página CreditosActualizar.aspx pasando como parámetro por el QueryString el ID del crédito asociado a la factura.
 - Se debe mostrar la opción de exportar a Exel el detalle de la factura con los ítems seleccionados.
- Si la factura es 0,
 - Se debe permitir ingresar la información asociada a esta, incluyendo la información relacionada con el crédito asociado en caso de que se seleccione el tipo de pago EnumTipoPago.Crédito.
 - Se deben permitir asociar nuevos ítems a la factura, para esto, cuando el usuario presione el botón “Nuevo ítem”, el sistema lanzará un popup en el cual solicite el tipo de ítem que desea asociar (ítem o servicio), además el código asociado a este, se deberá consultar la descripción del ítem en la tabla de tbItems o

tbServicios dependiendo del tipo seleccionado, además el valor asociado a esta selección.

- Se deberá solicitar el valor en porcentaje o en precio, con cualquiera de los 2 ingresos, se deberá calcular el otro (Ej: Si el usuario ingresa un precio de descuento, se calcula el porcentaje de descuento correspondiente a ese precio, si selecciona un porcentaje, entonces se deberá calcular el precio del descuento), y por último, se debe mostrar el valor total con descuento del ítem o servicio seleccionado.
- Se debe separar el IVA de cada ítem seleccionado. Al adicionar el registro y consultar el ítem, se debe consultar también este porcentaje y discriminarlo en el valor de la factura. Si se selecciona un servicio, se debe establecer el valor del IVA en 0.
- Al guardar, se deben validar que los valores ingresados sean diferentes de 0, y que toda la información obligatoria haya sido diligenciada.
- Generar el código de la factura tomando de la tabla de tbRangosFacturacion el rango activo en el sistema, sumando uno al contador de facturas y guardando de nuevo el registro, todo dentro de una misma transacción.
- Validar que el número de factura que se genera no exista en el sistema, si existe, lanzar una excepción informando la situación y no permitir crear la factura.
- Una vez se guarde el registro, debe redireccionar a la misma página con el ID de la factura que recién se creó

- Como en una factura lo más común es que no se tengan muchos ítems, y debido a la importancia de tener toda la información a la vista, en este listado de ítems no habrá paginador.

5.5.2.4.2 Controlar rangos de facturación

CU Controlar rangos de facturación Implementado completamente.

Lineamientos de implementación:

- Listado:
 - Crear el acceso a datos de la tabla tbRangosFacturación
 - Crear la pagina Rangos.aspx
 - Se deberán listar los rangos de facturación creados en el sistema
 - Crear un link en rango de la facturación, pasando como parámetro en el QueryString key = ID rango, si es nuevo registro, se debe pasar el parámetro en 0, el link debe redireccionar a la pagina RangosActualizar.aspx
 - Permitir exportar el listado que se consulta a Excel, al exportarlo, mostrar en el encabezado los filtros utilizados para consultar el listado de facturas.
 - Se debe mostrar el UCPaginador si la cantidad de registros es mayor a la especificada en el parámetro interno “Cantidad de registros para el listado”.

- Administración

- Se debe crear la página RangosActualizar.aspx
- Se debe recibir el parámetro del QueryString con el ID del rango de facturación
- Si el parámetro es diferente de 0
 - Se debe consultar el rango con el ID, si el rango está inactivo, entonces se debe cargar la información a modo lectura, en caso contrario, se permitirá modificar la información de rango y contador.
- Si el parámetro es 0
 - Se debe solicitar la información del rango y contador, por defecto el contador debe estar en 0
 - Al guardar, se debe validar que no haya otro rango con valor menor al ingresado, ya que se pueden duplicar los números de facturas.
 - Una vez guardado este rango, se debe establecer el anterior como inactivo y dejar este como único activo en la aplicación.

5.5.2.4.3 Administrar servicios adicionales

CU Administrar servicios adicionales Implementado completamente.

Lineamientos de implementación:

- Listado:
 - Crear el acceso a datos de la tabla tbServicios
 - Crear la pagina Servicios.aspx

- Listar los registros de la tabla tbServicios y adicionar un link en la descripción que redireccione a la pagina tbServiciosActualizar pasando como parámetro en el QueryString key = ID Servicio. Si se selecciona la opción de nuevo registro, se debe redireccionar con el parámetro 0
- Se debe mostrar el buscador permitiendo filtrar los registros por el código del servicio o por la descripción.
- Se debe mostrar el UCPaginador si la cantidad de registros es mayor a la especificada en el parámetro interno “Cantidad de registros para el listado”.
- Administración:
 - Se debe crear la pagina ServiciosActualizar.aspx
 - Se debe recibir el parámetro del QueryString con el ID
 - Si el ID Servicio es diferente de 0
 - Se debe buscar el registro y permitir modificar la información excepto el código del servicio
 - Si el ID es 0
 - Se debe solicitar toda la información, adicionalmente, se debe generar el código del servicio de la siguiente manera:
 - Se inicia una transacción de la base de datos.
 - Se consulta el parámetro “Consecutivo código de servicio”
 - Se le suma 1 al consecutivo
 - Se actualiza el consecutivo en la base de datos y se finaliza la transacción.
 - Se muestra el consecutivo como sololectura en la página.

- Al guardar, se debe validar que el consecutivo no exista, si existe, entonces se debe actualizar el consecutivo nuevamente y guardar.
- Se debe validar que el valor del servicio sea mayor que 0 que la descripción no esté vacía.
- Después de guardar, se debe redireccionar a la página de listado.

5.5.2.5 *Servicio de Windows*

5.5.2.5.1 Aplicar intereses por mora

CU Aplicar intereses por mora Implementado Completamente

CU Informar cuotas atrasadas Implementado Completamente

Lineamientos de implementación:

- Esta tarea se debe ejecutar todos los días de la semana a las 7 am.
- Se deben listar todas las cuotas sin la columna de intereses por mora con las siguientes condiciones:
 - Que la fecha de pago sea menor que la fecha actual
 - Que el estado de la cuota sea Pendiente
 - Que el estado del crédito sea En curso.
- Adicionar al DataTable resultante la columna de intereses por mora calculados

- Por cada cuota realizar lo siguiente:
 - Determinar el número de días pasados entre el día actual -1 y la fecha de vencimiento del pago de la cuota.
 - Consultar el porcentaje del interés por mora del parámetro “Porcentaje Intereses por mora”
 - Calcular los intereses con la siguiente fórmula:

Ecuación 2. Intereses.

$$\text{Intereses} = \text{Porcentaje} * \text{ValorCuota} * \text{DíasAtraso}$$

Fuente: Elaboración propia.

- Guardar el valor en la nueva columna del DataTable
- Crear una transacción y actualizar cada cuota del DataTable
- Generar un DataTable temporal (dtInforme) con la función `DataTable.DefaultView.ToTable(True, “Columna_Cedula”, “Columna_NombreCompleto)`
- Crear una estructura de tabla recorriendo los datos como se indica mas adelante, la tabla debe ser de la siguiente manera:

Tabla 42. Tabla para recorrer datos de intereses por mora.

Nombre				
Cédula				
Cuota	Valor	Fecha Programada	Días atraso	Intereses Mora

Fuente: Elaboración propia.

- Por cada registro en el dtInforme, realizar lo siguiente
 - Hacer una subconsulta en el DataTable de las cuotas vencidas filtrando por la cédula del registro actual.
 - Enviar el correo tanto al tomador del crédito, como a cada fiador, y con copia al usuario del almacén, informando que tiene esas cuotas vencidas y que estas están generando intereses por mora.

5.5.2.5.2 Recordar pagos de cuotas

CU Recordar pagos de cuotas Implementado Completamente

Lineamientos de implementación:

- Esta tarea se debe ejecutar todos los días de la semana a las 5 pm.
- Crear la tabla tbCuotasNotificadas
 - nidCuota Integer PK
 - dFechaNotificacion SmallDateTime
- Se deben listar las cuotas con estado enumEstadoCuotas.Pendiente y la fecha programada esté entre la fecha actual +1 y la fecha actual + parámetro “Dias alerta vencimiento de cuotas”
- Generar un DataTable temporal (dtInforme) con la función `DataTable.DefaultView.ToTable(True, “Columna_Cedula”, “Columna_NombreCompleto)`

- Crear una estructura de tabla recorriendo los datos como se indica mas adelante, la tabla debe ser de la siguiente manera:

Tabla 43. Tabla para recorrido de datos de recordatorio de pago de cuotas.

Nombre		
Cédula		
Cuota	Valor	Fecha Programada

Fuente: Elaboración propia.

- Por cada registro en el dtInforme, realizar lo siguiente
 - Hacer una subconsulta en el DataTable de las cuotas próximas a vencer filtrando por la cédula del registro actual.
 - Si el cliente tiene activo el envío de correo, entonces enviar el correo al tomador del crédito, informando que tiene esas cuotas próximas a vencer.
 - Si el cliente no tiene activo el envío de correo, entonces validar en la tabla tbCuotasInformadas si el ID de la cuota no existe, si es así enviar el correo al empleado del almacén informando que el cliente tiene inactivo el servicio de correo, y almacenar el registro en la tabla de tbCuotasNotificadas

5.5.2.6 *Módulo de inventario*

5.5.2.6.1 Generar Código Único

5.5.2.6.2 Consultar Stock de artículos

5.5.2.6.3 Registrar lotes de entrada

5.5.2.6.4 Registrar salida de artículo

5.5.2.6.5 Descontar ítems del inventario

5.5.2.7 *Módulo de cotizaciones*

5.5.2.7.1 Generar cotización

5.5.2.7.2 Imprimir cotización

5.5.2.7.3 Exportar cotización

5.5.2.8 *Módulo de cuentas por pagar*

5.5.2.8.1 Administrar facturas por pagar

5.5.2.8.2 Alertar vencimiento de facturas por pagar

5.5.2.9 *Módulo de Reportes.*

5.5.2.9.1 Listar lotes de entrada en un rango de fechas

5.5.2.9.2 Listar clientes con cuotas atrasadas

5.5.2.9.3 Exportar listado de clientes con cuotas atrasadas

5.5.2.9.4 Listar crédito de cliente

5.5.2.9.5 Listar Historial de pagos de cliente

5.5.2.9.6 Exportar crédito de cliente

5.5.2.9.7 Listar cuentas por pagar

5.5.2.9.8 Exportar listado de cuentas por pagar

5.6 Soluciones Similares a las necesidades

5.6.1 World office

Este es un sistema de gestión completo, que permite manejar los módulos de manera separada, con diversa cantidad de informes según los módulos que se seleccionen, también permite integrar el manejo de la aplicación con los impuestos que se generan y consultar el informe de manera entregable a las autoridades del país. (World Office, 2015)

Los módulos que se ajustan a las necesidades descritas son:

- Contabilidad
- Impuestos
- Inventarios
- Ventas
- Cartera y Cuentas por Pagar
- Nomina

El software, con los módulos seleccionados, tiene un valor aproximado de 5.000.000, la variación depende de si se escoge a crédito o en pago inmediato.

Este sistema se descarta, porque el almacén no está dispuesto a pagar un valor tan elevado para resolver una situación informática a medias, este software, aunque es demasiado

parametrizable, no es un desarrollo a la medida de las necesidades de un almacén que se ha dedicado más que a vender artículos, a generar una cartera con un pobre manejo, no muy soportada en el software evaluado.

5.6.2 Affinity

Este sistema es el que actualmente apoya el manejo del almacén, es un sistema bastante robusto actualizado con las leyes de Colombia. Por ser genérico, su utilidad se da de acuerdo a las configuraciones que se tengan. Adicionalmente, el sistema es en línea, por lo que facilita el ingreso y consulta de los datos.

Sin embargo, para parametrizar este software se requiere de experiencia en contabilidad, al ser una herramienta tan configurable, se pueden presentar errores en declaraciones de impuestos o balances del almacén, además, no tiene módulo de cartera, por lo que no es una solución completa. (Asp Affinity, 2016)

6 Conclusiones

Luego de la fase de análisis de los requisitos, se realizó una búsqueda de sistemas similares genéricos que permitieran satisfacer las necesidades del cliente, se encontraron 2 soluciones que no contemplan todas las necesidades, y se decidió continuar con las demás fases del proyecto para dar solución a estas necesidades.

Durante la aplicación de la metodología seleccionada, se pudo observar la importancia de las fases de análisis de los requisitos y diseño de la solución, ya que son las bases fundamentales para la implementación del sistema.

Aunque todavía hay trabajo futuro, la base del sistema que está en construcción **está bien** definida y se espera no tenga que ser modificada, ya que todos los requisitos están documentados y modelados en las entidades correspondientes,

Realizando una comparación de la metodología tradicional versus la metodología ágil, se observa que en la tradicional, el usuario final tiene un trabajo fuerte únicamente al inicio del proyecto, pero solo observa resultados luego de la fase de implementación, mientras que en la metodología ágil, se observan resultados desde el principio, sin embargo, el trabajo fuerte del usuario final se mantiene durante todo el proceso de implementación.

7 Bibliografía

Abad Londoño, J. H. (6 de 4 de 2005). *Ing-sw*. Obtenido de Tipos de pruebas de software:

<http://ing-sw.blogspot.com.co/2005/04/tipos-de-pruebas-de-software.html>

Ageraldo. (15 de 12 de 2005). *Diseño y modelación de un proyecto de software utilizando el*

lenguaje UML. Obtenido de Monografias:

<http://www.monografias.com/trabajos28/proyecto-uml/proyecto-uml.shtml>

Asp Affinity. (2016). *Asp Affinity*. Recuperado el 2016, de <https://www.aspaffinity.biz/>

Buenas tareas. (26 de 4 de 2016). *Buenas tareas*. Obtenido de Las 50 preguntas que se deben

hacer para obtener requerimientos: <http://www.buenastareas.com/ensayos/Las-50->

[Preguntas-Que-Se-Deben/1367813.html](http://www.buenastareas.com/ensayos/Las-50-Preguntas-Que-Se-Deben/1367813.html)

Canarias, I. (7 de 2 de 2012). *Buenas prácticas para la construcción de software*. Obtenido de

Slide share: [http://es.slideshare.net/ikercanarias/buenas-prcticas-para-la-construccin-de-](http://es.slideshare.net/ikercanarias/buenas-prcticas-para-la-construccin-de-software)

[software](http://es.slideshare.net/ikercanarias/buenas-prcticas-para-la-construccin-de-software)

Cervantes, H. (24 de 2 de 2010). *Arquitectura de Software*. *SG Buzz - Conocimiento para crear*

software grandiosos, [http://sg.com.mx/revista/27/arquitectura-](http://sg.com.mx/revista/27/arquitectura-software#.V13Mg_nhDNN)

[software#.V13Mg_nhDNN](http://sg.com.mx/revista/27/arquitectura-software#.V13Mg_nhDNN). Obtenido de SG Buzz - Conocimiento para crear software

grandiosos: http://sg.com.mx/revista/27/arquitectura-software#.V13Mg_nhDNN

Cobian, C. (20 de 11 de 2013). *9 Herramientas para crear maquetas interactivas*. Obtenido de

Cobian Media: [http://www.cobianmedia.com/2013/11/20/9-herramientas-para-crear-](http://www.cobianmedia.com/2013/11/20/9-herramientas-para-crear-maquetas-interactivas-2/)

[maquetas-interactivas-2/](http://www.cobianmedia.com/2013/11/20/9-herramientas-para-crear-maquetas-interactivas-2/)

Fernández, X. (2 de 10 de 2013). *Proyectos de guerrilla sobre sistemas, operaciones y la cadena de suministro*. Obtenido de Las cinco etapas de ingeniería del software:

<http://proyectosguerrilla.com/blog/2013/02/las-cinco-etapas-en-la-ingenieria-del-software/>

García, C. A. (1 de 9 de 2007). *Software Guru*. Obtenido de Software Guru:

http://sg.com.mx/revista/17/obtencion-requerimientos-tecnicas-y-estrategia#.V2DI_P19670

Hastie, S., & Wojewoda, S. (4 de 10 de 2015). *InfoQ*. Obtenido de Standish group 2015 chaos report - Q&A with Jennifer Lynch: <https://www.infoq.com/articles/standish-chaos-2015>

Herrera, L. J. (12 de 4 de 2016). *Monografías*. Obtenido de Ingeniería de requerimientos ingeniería de software: <http://www.monografias.com/trabajos6/resof/resof2.shtml>

International requirements engineering boards. (11 de 6 de 2016). Obtenido de CPRE by IREB - Your personal certificate: <https://www.ireb.org/en>

International software testing qualification board. (10 de 6 de 2016). *Foundation Level material for download*. Obtenido de International software testing qualification board:

<http://www.istqb.org/certification-path-root/foundation-level.html>

ISO 12207-1. (14 de 06 de 2013). *Ingeniería de Software*. Obtenido de web archive org:

<http://web.archive.org/>

Junta de andalucia. (24 de 4 de 2016). *Junta de andalucia*. Obtenido de Guía para la redacción de casos de uso: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/416>

Maxim, B. (24 de 3 de 2014). *Universidad nacional abierta y a distancia*. Obtenido de

Introducción a la ingeniería de sistemas:

<https://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=10&ved=0ahUKEwiHxoP0k6HNAhWGbSYKHeoPCjkQFghmMAk&url=http%3A%2F%2Fgroups.eengin.umd.umich.edu%2FCIS%2Fcourse.des%2Fcis376%2Fppt%2Flec8c.ppt&usg=AFQjCNESYQMnL-Jb4yA9hSwoKg44pqscKA&sig2=D4oMvvXx>

Metodología gestión de requerimientos. (20 de 5 de 2016). *Metodología gestión de*

requerimientos. Obtenido de Capítulo 2 Técnicas para identificar requisitos funcionales y no funcionales: <https://sites.google.com/site/metodologiareq/capitulo-ii/tecnicas-para-identificar-requisitos-funcionales-y-no-funcionales>

Microsoft. (2 de 4 de 2016). *Microsoft*. Obtenido de Microsoft Certified Solutions Developer

(MCS D): <https://www.microsoft.com/en-us/learning/mcsd-certification.aspx>

Oracle Certified Professional, Java SE 7 Programmer. (15 de 2 de 2016). Obtenido de Oracle:

https://education.oracle.com/pls/web_prod-plq-dad/db_pages.getpage?page_id=458&get_params=p_track_id:OCPJSE7

Ortiz, J. (13 de 5 de 2016). *Prezi*. Obtenido de Diseño y desarrollo de sistemas:

<https://prezi.com/jq2537byjcyb/disen-y-desarrollo-de-sistemas/>

Pérez Hernández, M. A. (29 de 5 de 2013). *Ingsoftmkh*. Obtenido de Fases de prueba de

Software: <http://ingsoftmkh.blogspot.com.co/2013/05/fases-de-prueba-de-software.html>

Plugin Portal. (02 de 05 de 2016). Obtenido de Netbeans:

<http://plugins.netbeans.org/PluginPortal/>

Sánchez Rios, S. (2 de 5 de 2016). *Slide share*. Obtenido de Ingeniería de Software - Unidad 1
Análisis de Requerimientos: <http://es.slideshare.net/SergioRios/unidad-13-analisis-de-requerimientos>

Solorio, M. (16 de 4 de 2013). *Metodología en cascada*. Obtenido de Metodología en cascada:
<http://metodologiaencascada.blogspot.com.co/>

Sommerville, I. (10 de 12 de 2014). *software-engineering-book*. Obtenido de Slide Share:
<http://www.slideshare.net/software-engineering-book/ch1-introduction-42645973>

Tedeschi, N. (23 de 5 de 2016). *¿Qué es un Patrón de Diseño?* Obtenido de Microsoft MSDN:
<https://msdn.microsoft.com/es-es/library/bb972240.aspx>

Universidad de Chile. (4 de 3 de 2016). *Universidad de Chile*. Obtenido de Casos de uso (use case): <http://users.dcc.uchile.cl/~psalinas/uml/casosuso.html>

Velázquez Cruz, A. D. (2012). Mejores prácticas para el desarrollo de software. *SG Buzz - Conocimiento para crear software grandioso*, <http://sg.com.mx/revista/mejores-pr%C3%A1cticas-para-el-desarrollo-software#.V11d6JF95hE>.

Visual studio gallery. (25 de 5 de 2016). Obtenido de Productos y extensiones para Visual Studio: <https://visualstudiogallery.msdn.microsoft.com/>

Wikipedia. (12 de 06 de 2016). *Ingeniería de software*. Obtenido de Wikipedia:
https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software#Modelo_en_cascada_o_cil.C3.A1sico

World Office. (2015). *World office*. Recuperado el 2015, de <http://www.worldoffice.com.co/>

Yuml. (2 de 4 de 2016). *Yuml*. Obtenido de Yuml: <http://yuml.me/>

8 Anexos

Anexo 1 Código fuente

[Código fuente](#)

Anexo 2 Instaladores de aplicación

[Instaladores de aplicación](#)

Anexo 3 Base datos

[Bases de datos](#)