

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

CARRERA: INGENIERÍA DE SISTEMAS

**Trabajo de titulación previo a la obtención de los títulos de:
INGENIEROS DE SISTEMAS**

**TEMA:
ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UN
SISTEMA QUE CONTROLE ACTIVOS FIJOS, SUMINISTROS, BIENES
CONTROLADOS, DEPRECIACIÓN DE BIENES PARA LA
SUPERINTENDENCIA DE CONTROL DEL PODER DE MERCADO**

**AUTORES:
LUIS ANDRES NÚÑEZ FLORES
FERNANDO DANIEL LINCANGO BAQUERO**

**TUTOR:
WASHINGTON RAÚL PADILLA ARIAS**

Quito, abril de 2016

Cesión de derechos de autor

Nosotros Luis Andres Núñez Flores, con documento de identificación N° 1724613706 y Fernando Daniel Lincango Baquero con documento de identificación N°1723674899, manifestamos nuestra voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA QUE CONTROLE ACTIVOS FIJOS, SUMINISTROS, BIENES CONTROLADOS, DEPRECIACIÓN DE BIENES PARA LA SUPERINTENDENCIA DE CONTROL DEL PODER DE MERCADO, mismo que ha sido desarrollado para optar por el título de: Ingenieros de Sistemas, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



Nombre: Luis Andres Núñez Flores

Cédula: 1724613706

Fecha: abril de 2016



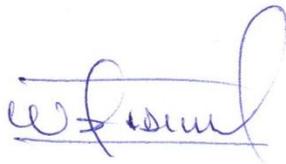
Nombre: Fernando Daniel Lincango Baquero

Cédula: 1723674899

Declaratoria de coautoría del docente tutor

Yo declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación ANÁLISIS, DISEÑO, DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA QUE CONTROLE ACTIVOS FIJOS, SUMINISTROS, BIENES CONTROLADOS, DEPRECIACIÓN DE BIENES PARA LA SUPERINTENDENCIA DE CONTROL DEL PODER DEL MERCADO realizado por Luis Andrés Núñez Flores y Fernando Daniel Lincango Baquero, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerados como trabajo final de titulación.

Quito, abril 2015



Washington Raúl Padilla Arias

CI: 1707492888

RESUMEN

El presente proyecto busca solventar una de las necesidades fundamentales de la Superintendencia de Control del poder de mercado(SCPM) mejorando la administración de los bienes controlados y los suministros mediante la implementación de un sistema informático web , el cual permitirá el ingreso y la salida de los suministros y bienes controlados a bodega.

También permitirá realizar pedidos siguiendo el proceso que se estableció en este mismo proyecto, en donde existe el administrador quien será la persona que aprueba dicho pedido y posteriormente el pedido pasa a bodega para su despacho.

El presente documento, describe todo el proceso que se realizó durante el desarrollo del proyecto. Aquí se podrá observar las distintas etapas que se efectuaron en el transcurso del mencionado trabajo, desde la recolección de requerimientos hasta la realización de las pruebas. El contenido del documento se lo dividió en tres partes fundamentales:

Capitulo uno, se encuentra el marco teórico donde se definen los conceptos básicos que se va a tratar, también se dará detalle sobre las herramientas que se utilizaran a lo largo del desarrollo del proyecto.

Capitulo dos en donde se procede con la realización del proceso con que se va solventar la necesidad de la SCPM en cuanto a la gestión de suministros y bienes controlados. Desde la base de este proceso se hará la recolección de los requerimientos con los que se elaborara el sistema.

Capitulo tres donde se procede con el desarrollo de la aplicación y las pruebas en base a los requerimientos solicitados.

ABSTRACT

This project seeks to solve one of the basic needs of the Superintendencia de Control del Poder de Mercado (SCPM) improving the management of controlled goods and supplies by implementing a Web computer system, which will allow the entry and exit supplies and goods controlled cellar.

It will also enable orders following the process established in this project, where there is a manager who will be the person who approves such a request and then passes the request for release warehouse.

This document describes the process that took place during the project. Here you can see the different stages that were made in the course of that work, from gathering requirements to conducting the tests. The content of the document is divided it into three main parts:

Chapter one is the theoretical framework where the basic concepts to be treated are defined, also give detail about the tools that will be used throughout the development of the project.

Chapter Two where proceed with the completion of the process that will address the need of SCPM regarding supply management and controlled goods. From the base of this process the collection of requirements that the system will be developed.

Chapter Three where we proceed with the application development and testing based on the requirements requested.

ÍNDICE

INTRODUCCIÓN	1
Antecedentes.....	1
Justificación.....	3
Alcance	4
Objetivos	5
Objetivo general	5
Objetivos específicos.....	5
Marco Metodológico.....	6
Metodología XP.....	7
Fases de la metodología XP	10
Planificación del Proyecto	10
Diseño 10	
Codificación.....	11
Pruebas 11	
Leguaje de modelado UML.....	12
Diagrama de casos de uso.....	12
Diagrama de clases	12
Diagrama de relaciones de entidad	15
CAPÍTULO 1	18
ESTADO DEL ARTE.....	18
1.1 Marco Teórico	18

1.1.1	ERP	19
1.1.2	Lenguajes de Desarrollo	19
1.1.3	Herramientas de Modelado	23
CAPÍTULO 2		25
ANÁLISIS Y DISEÑO		25
2.1	Definición del proceso que se va a automatizar	25
1.1.2	Explicación del Diagrama de procesos	27
2.2	Definición de requerimientos	30
2.3	Diagrama de clases	35
2.3.1	Desarrollo de tarjetas CRC	36
CAPÍTULO 3		43
CONSTRUCCION Y PRUEBAS		43
3.1	Herramientas para el desarrollo	43
3.2	Diagrama de entidad relación	43
3.2.1	Descripción de la base de datos	44
3.2.2	Diccionario de Datos	47
3.3	Construcción de la Estructura MVC	57
3.3.1	Capa Modelo	57
3.3.2	Capa de Vista	58
3.3.3	Capa de controlador	59
3.4	Construcción de la aplicación web.	60
3.4.1	Acceso a datos	61

3.4.2 Estructura JavaScript.	64
3.4.3 Llamadas AJAX.....	66
3.4.4 Interfaz.....	68
3.4.5 Alertas y validación.....	69
3.4.6 Librerías.....	70
3.5 Pruebas	71
3.5.1 Pruebas solicitud de suministros.....	71
3.5.2 Pruebas estado de pedidos.....	73
3.5.3 Pruebas revisión de pedidos.....	74
3.5.4 Pruebas administración de suministros.....	75
3.5.5 Pruebas administración de proveedores.....	77
3.5.6 Pruebas entrega de pedido.....	78
3.5.7 Pruebas reportes suministros	79
3.5.8 Pruebas administración de categorías.....	80
3.5.9 Pruebas administración de bienes asignados.....	81
CONCLUSIONES	83
TRABAJOS FUTUROS	85
REFERENCIAS	86

ÍNDICE DE FIGURAS

Figura 1. Estructura ERP SCPM.	5
Figura 2. Cuadro de características de metodología XP.	8
Figura 3. Ejemplo de diagrama de clases.	13
Figura 4. Ejemplo de representación de una clase en UML.	13
Figura 5. Ejemplo de generalización o herencia entre las clases.	14
Figura 6. Ejemplo de Asociación entre clases según UML.	15
Figura 7. Ejemplo de Acumulación entre clases según UML.	15
Figura 8. Ejemplo de Composición entre clases según UML.	15
Figura 9. Ejemplo de diagrama entidad relación según UML.	16
Figura 10. Ejemplo de representación de entidad según UML.	17
Figura 11. Diagrama del proceso que realizara el sistema.	26
Figura 12. Diagrama de clases.	36
Figura 13. Diagrama entidad relación de la Base de Datos.	44
Figura 14. Contenido del proyecto base.	58
Figura 15. Contenido del proyecto Procesos Ambiental.	59
Figura 16. Contenido del webForm "AsignarBienControlado".	59
Figura 17. Logo de Visual Studio. Fuente: (Network M. D., 2015)	60
Figura 18. Logo de Microsoft SQLServer. Fuente: (Network M. D., 2015)	60
Figura 19. Código ejemplo de inserción de datos mediante SP.	62
Figura 20. Código ejemplo búsqueda de datos mediante SP.	63
Figura 21. Código ejemplo actualización de datos mediante SP.	63
Figura 22. Código ejemplo para creación de un SP.	64
Figura 23. Estructura de archivos JavaScript de la aplicación.	65
Figura 24. Ejemplo método AJAX.	66

Figura 25. Ejemplo método web Controlador.....	67
Figura 26. Ejemplo método de acceso a datos.	67
Figura 27. Interfaz ventana modal de Bootstrap.	69
Figura 28. Alerta mensaje de validación.....	70
Figura 29. Alerta de acción exitosa.....	70
Figura 30. Validaciones módulo solicitud de suministros.	72
Figura 31. Ventana modal de confirmación módulo solicitud de suministros.....	73
Figura 32. Ventana Modal del estado de los pedidos.....	74
Figura 33. Ventana revisión de pedido solicitado.....	75
Figura 34. Ingreso de nuevos suministros.....	76
Figura 35. Administración de proveedores.	77
Figura 36. Calendario de entrega de pedidos.	78
Figura 37. Reporte obtenido bajo parámetros de búsqueda.	79
Figura 38. Administración de una sub categoría.....	81

ÍNDICE DE TABLAS

Tabla 1. Comparativa de metodologías ágiles vs tradicionales.	7
Tabla 2. Ejemplo historia de usuario. (es.slideshare.net, 2015).....	10
Tabla 3. Ejemplo tarjeta C.R.C	11
Tabla 4. Descripción de las actividades de los actores dentro del proceso.....	27
Tabla 5. Continuación de la tabla 4.....	28
Tabla 6. Continuación de la tabla 4.....	29
Tabla 7. Continuación de la tabla 4.....	30
Tabla 8. Tabla de rangos de importancia.	31
Tabla 9. Historia de usuario 1.	31
Tabla 10. Historia de usuario 2.	31
Tabla 11. Historia del usuario 3.	32
Tabla 12. Historia del usuario 4.	33
Tabla 13. Historia del usuario 5.	33
Tabla 14. Historia del usuario 6.	34
Tabla 15. Historia del usuario 7.	34
Tabla 16. Historia del usuario 8.	35
Tabla 17. Tabla CRC Clase Bienes.....	36
Tabla 18. Tabla CRC clase Casos.....	37
Tabla 19. Tabla CRC clase Categoría.	37
Tabla 20. Tabla CRC clase Condición.....	37
Tabla 21. Tabla CRC clase Documentos.	38
Tabla 22. Tabla CRC clase Entrega Pedidos.	38
Tabla 23. Tabla CRC Estado Pedidos.....	38
Tabla 24. Tabla CRC clase Formas de Adquisición.	39

Tabla 25. Tabla CRC clase Kardex.....	39
Tabla 26. Tabla CRC clase Marca.	39
Tabla 27. Tabla CRC clase Materiales.....	40
Tabla 28. Tabla CRC clase Pedido.	40
Tabla 29. Tabla CRC clase Proveedor.	40
Tabla 30. Tabla CRC clase Revisión Pedido.	41
Tabla 31. Tabla CRC clase Subcategoría.....	41
Tabla 32. Tabla CRC clase Suministros.	41
Tabla 33. Tabla CRC clase Tipo Bien.	42
Tabla 34. Tabla CRC clase Utilitario Suministro.	42
Tabla 35. Descripción de las tablas de la base de datos.....	45
Tabla 36. Continuación de la tabla 35.....	46
Tabla 37. Diccionario de datos BIENES_CONTROLADOS.....	47
Tabla 38. Diccionario de datos de BIENES_FUNCIONARIO.	48
Tabla 39. Diccionario de datos de CASOS.....	48
Tabla 40. Diccionario de datos de CATEGORIA.....	49
Tabla 41. Diccionario de datos de CONDICION.	49
Tabla 42. Diccionario de datos de FORMA_ADQUISICION.	49
Tabla 43. Diccionario de datos de INDICES.....	50
Tabla 44. Diccionario de datos de INGRESO DOCUMENTOS.....	50
Tabla 45. Continuación tabla 44.	51
Tabla 46. Diccionario de datos de KARDEX.	51
Tabla 47. Diccionario de datos de MARCA.	52
Tabla 48. Diccionario de datos de MATERIALES.	52
Tabla 49. Diccionario de datos de PEDIDOS.....	52

Tabla 50. Continuación tabla 49.	53
Tabla 51. Diccionario de datos de PROVEEDOR.....	54
Tabla 52. Diccionario de datos de SCPM_PERSONAL.	55
Tabla 53. Diccionario de datos de SUB_CATEGORIA.....	56
Tabla 54. Diccionario de datos de SUMINISTROS.	56
Tabla 55. Continuación de la tabla 54.....	57
Tabla 56. Diccionario de datos de TIPO BIENES.....	57

ANEXOS

Anexo 1. Glosario de términos.....	89
Anexo 2. Formulario para solicitud de suministros.....	93

INTRODUCCIÓN

El proyecto se presenta como requisito para obtener el título de Ingeniería en Sistemas en la Universidad Politécnica Salesiana.

La posibilidad de integrar el equipo de ingenieros en sistemas en el Ecuador ha generado la motivación para contribuir como estudiantes de la citada especialización en la consolidación y mejoramiento de la tecnología informática, a través de la gentil y honrosa intervención mediante Convenio, entre La Superintendencia de Control del Poder de Mercado, para la presentación de proyectos y la Universidad Politécnica Salesiana, en la formulación de transferencia de tecnologías apropiadas para las organizaciones gubernamentales del Estado Ecuatoriano, ha permitido intervenir y participar en la posibilidad de aplicar dentro de un sistema integral, diversificado y efectivo en el desarrollo de software que reduzca considerablemente la compra de estos paquetes informáticos, que tienen un alto costo dentro de la oferta privada.

Antecedentes

Las organizaciones inteligentes se enmarcan en los parámetros de; oportunidad, eficaz y estratégicas facilitando respuestas inmediatas a la necesidad permanente de adaptación a los cambios de su entorno. La importancia y velocidad de estos cambios son aspectos que definen el escenario estratégico donde se desenvuelve. Cambios que se simbolizan por medio de amenazas o expresión de factores negativos que ponen en dificultad su eficiencia, supervivencia y oportunidades, como factores positivos que permiten aprovechar las situaciones externas. Ante esta perspectiva la organización debe contar con unas fortalezas tecnológicas para alcanzar la adaptación perseguida. La actitud constante de actualización tecnológica persigue una constante adaptación de la organización con el entorno inestable, creyendo que el futuro puede ser mejorado a

través del desarrollo de paquetes informáticos acordes con las demandas existentes. La alta dirección de la organización elige no sólo la fortaleza interna, sino que también obliga a desarrollar software que cumpla con las exigencias del entorno a través de la fijación de maneras de competir.

La administración moderna demanda de las organizaciones inteligentes mayor efectividad en la ejecución de sus procesos, la Superintendencia de Control del Poder de Mercado desde ahora referenciada con las siglas “SCPM” cumple un rol determinante en el direccionamiento estratégico de las demandas y ofertas del mercado a nivel nacional, razón suficiente para desarrollar soluciones efectivas que simplifiqué la complejidad de sus procedimientos, utilizando el potencial tecnológico con el que cuenta actualmente la institución.

Los procesos segmentados, dispersos y difusos conducen a la desorganización, establecen barreras entre las diferentes áreas de trabajo en cuanto a la obtención y distribución de los suministros y bienes controlados, obstaculizando la adecuada interrelación entre los procesos involucrados, para existe la necesidad de implementar un sistema automatizado moderno para unificar, ordenar y armonizar los procesos mencionados y consecuentemente mejorar de forma sustancial el desempeño de la asignación de los bienes y la distribución de los suministros en las todas las áreas de la institución.

En consideración a esta realidad expuesta se puede citar que; una de las debilidades detectadas en la SCPM es en los procesos de distribución del contenido de bodega entre las diferentes áreas, para lo cual se puede utilizar la tecnología de punta en el área informática, creando un software que solucione esta necesidad.

La base legal y técnica, para determinar esta alternativa se sustenta en el decreto ejecutivo 1004, dictado en el 2008. En él se promueve el uso del software libre para lograr la independencia económica en cuanto al uso de tecnología, el decreto busca que en todas las instituciones públicas usen el software libre como plataforma para todos los desarrollos posteriores a esa, también se busca la migración de tecnologías y sistemas que se encuentran en plataformas paganas a plataformas libres. (PUBLICA, 2015) Pese a esto la SCPM trabaja sobre una plataforma Microsoft visual Studio 2008 usando una base de datos SQL Server 2008, las cuales nos son de uso libre sino licenciado.

Justificación

Ante la ineficacia de los modos de software convencionales para adaptar la organización a un entorno cada vez más hostil, nace la necesidad de una respuesta efectiva y que se ajuste a las demandas que exige la organización para la cual se desarrolla el software a la medida, que permita que la unidad económica pueda sobrevivir o incrementar su eficacia optimizando procesos muchos de los cuales se los realiza de manera errónea o de forma manual.

Debido a lo expuesto anteriormente y mediante el convenio entre la SCPM y la Universidad Politécnica Salesiana se planteó que la institución pública va a proveer de proyectos de desarrollo con el fin de levantar un sistema eficiente y con lo cual exista un beneficio entre estudiantes de la UPS los cuales desarrollaran los diferentes proyectos para poder obtener el título académico y la SCPM que será la beneficiaria de cada uno de estos proyectos que formaran parte de un ERP.

Este sistema se desarrollara en conjunto con SCPM la cual requiere un software que permita gestionar y distribuir tanto suministros como bienes controlados que se

encuentran registrados en esta institución con el cual se optimice el proceso que se realiza actualmente de forma manual y con el cual se quiere eliminar en lo más posible el uso del papel, el sistema también permitirá la realización de los pedidos al área de bodega pasando por un filtro administrativo el cual autorizara el pedido, para luego ser despachado, todos los reportes que se hagan sobre el estado del proceso del pedido así como el reporte de entrega recepción se manejaran de manera digital eliminando el uso del papel.

Alcance

Con el proyecto “analizar diseñar y construir un sistema que permita el control de suministros y bienes controlados de la SCPM” busca optimizar los procesos de solicitud de pedidos y la administración del contenido en bodega de manera estratégica economizando el tiempo de trabajo del personal, además permitirá ajustar la estrategia en cuanto a la productividad y eficacia la cual se espera sea mejorada ya que el sistema no solo va dirigido al área de bodega sino que será utilizado por todas las áreas al momento de realizar los pedidos de suministros.

El sistema que se va a desarrollar será anclado de manera estratégica al sistema actual de la SCPM, el cual maneja diferentes aspectos de la institución como ejemplo: la gestión de talento humano y de vehículos de la institución. El sistema será parte de este ERP y estará el encargado de la gestión de suministros y activos fijos, tal y como lo muestra la Figura1.

Diagrama ERP SCPM

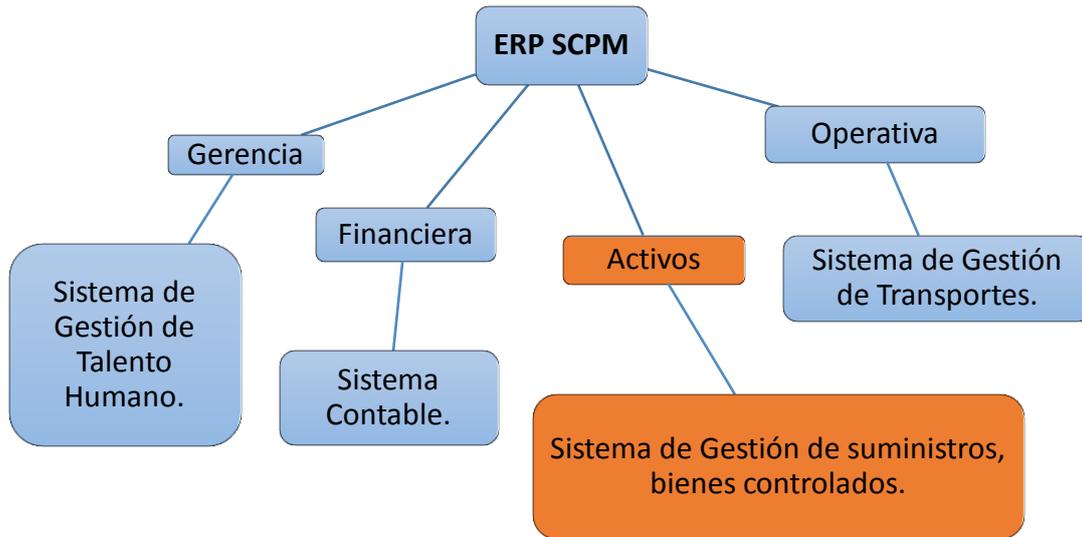


Figura 1. Estructura ERP SCPM.

Elaborado por: Luis Núñez & Daniel Lincango

Objetivos

Objetivo general

Optimizar las actividades de administración y despacho de suministros y bienes controlados ubicados en las bodegas de la SCPM la cual actualmente se lleva de manera manual, mediante el desarrollo e implementación de un sistema informático que se realizará en conjunto con un encargado del área de informática de la institución, en base a las plataformas tecnológicas existente en la SCPM.

Objetivos específicos

- Establecer el proceso requerido por SCPM sobre el cual se va a desarrollar el sistema.

- Revisar los parámetros y herramientas que utiliza la SCPM para la realización e implementación de un sistema.
- Analizar el tipo de lenguajes y plataformas son las que usa la SCPM para realizar sus desarrollos.
- Evaluar el sistema en conjunto con empleados de la SCPM creando un ambiente de evaluación ideal.
- Entregar la documentación necesaria que solvente que el software se entregó según los requerimientos así como también entregar los respectivos manuales de usuario.

Marco Metodológico

Debido a la importancia en el ámbito de desarrollo de software, en la actualidad se cuentan con diferentes metodologías ágiles que han desplazado a las metodologías de desarrollo convencional ya que estas permiten un desarrollo dinámico y a la par con los requerimientos que necesita el usuario según como se muestra en la (Tabla 1), estas metodologías son variadas y todas ellas se enfocan a la satisfacción del cliente mediante la entrega un software que cumpla con un desarrollo basado en las buenas prácticas y el control de calidad, como por ejemplo: XP, RUP, SCRUM.

Tabla 1. Comparativa de metodologías ágiles vs tradicionales.

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo de desarrollo)	Impuestas externamente
Proceso menos controlados, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Nota: (Bolivariana, 2015).

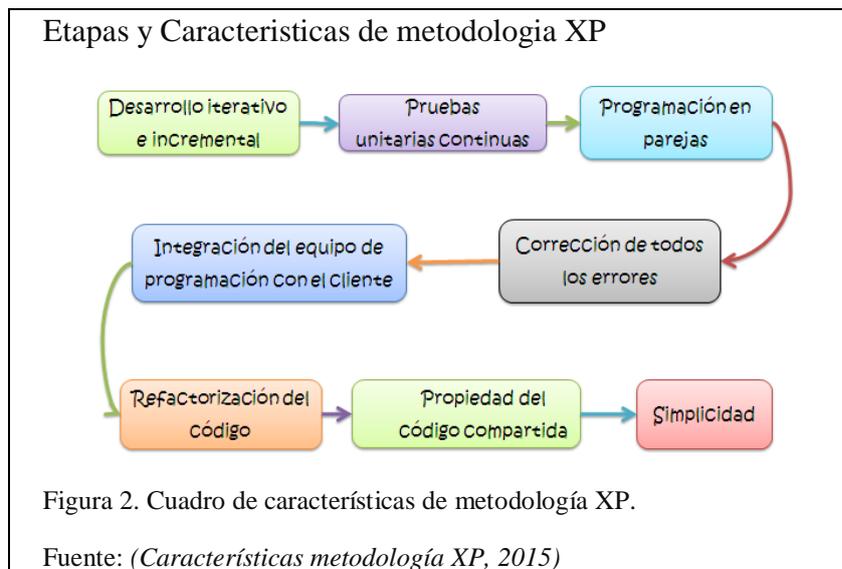
De entre de las diferentes metodologías de desarrollo ágil, la que se va aplicar la metodología XP (Xtreme Programming), para el desarrollo del sistema y UML como un lenguaje de modelado el cual permitirá realizar diferentes gráficas sobre cómo va estructurado el sistema.

Metodología XP

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como

especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. (Universidad Union Bolivariana, 2015)

La principal ventaja que ofrece XP frente a las otras metodologías de desarrollo ágiles es que está orientada a un grupo de desarrollo pequeño, para especificar se promueve el desarrollo por parejas y al contar este proyecto con la misma cantidad de equipo es ideal para cumplir con las expectativas de calidad. Para complementar las ventajas el siguiente gráfico explicará las diferentes características que posee esta metodología.



Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.

Pruebas unitarias continuas: frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.

Programación en parejas: supone la mayor calidad del código escrito de esta manera ya que el código es revisado y discutido mientras se escribe lo cual es más importante que la posible pérdida de productividad inmediata.

Integración del equipo de programación con el cliente: se recomienda que un representante del cliente forme parte del equipo de trabajo.

Corrección de todos los errores: antes de añadir nueva funcionalidad. Hacer entregas frecuentes.

Refactorización del código: es decir, describir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.

Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.

Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo (Características metodología XP, 2015).

Fases de la metodología XP

Planificación del Proyecto

La parte principal de esta fase de desarrollo es la llamada historia de usuario que es el primer paso de cualquier proyecto que siga la metodología XP. Ya que se va a utilizar UML para los diagramas, los casos de uso se reemplazaran por estas historias de usuario, las cuales son escritas por el cliente de manera general sin el uso de lenguaje técnico. Estas son usadas para estimas tiempos de desarrollo y posteriormente se utiliza en la fase de prueba como una verificación del cumplimiento de los requerimientos del usuario. Una vez entregada la historia de usuario se pasa a la fase de diseño.

Tabla 2. Ejemplo historia de usuario.

Historia:	Nombre de la clase o proceso
ID:	Número de historia de usuario
Descripción:	Se enuncia la funcionalidad a codificar y las restricciones
Importancia:	Se asigna un número en un rango predefinido que indica que tan crítico es el proceso
Como probarlo / Pruebas de aceptación:	El cliente detalla los criterios bajo los cuales se evaluará el módulo y se determinará si está o no listo.

Nota: (es.slideshare.net, 2015)

Diseño

La premisa de XP es hacer un diseño sencillo que sea entendible y amigable al usuario sin complicarse mucho en el desarrollo y la implementación del mismo, uno de los artefactos para la parte de diseño son las tarjetas CRC.

El uso de las tarjetas C.R.C (Class, Responsibilities and Collaboration) permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica.

Las tarjetas C.R.C representan objetos; la clase a la que pertenece el objeto se puede escribir en la parte de arriba de la tarjeta, en una columna a la izquierda se pueden

escribir las responsabilidades u objetivos que debe cumplir el objeto y a la derecha, las clases que colaboran con cada responsabilidad.

Tabla 3. Ejemplo tarjeta C.R.C

Nombre de la clase	Se describe el nombre de la clase
Las responsabilidades de la clase.	Que actividades es la que realiza la clase
Las clases con las que va a colaborar.	Con que clases va a colaborar
Autor	Quien realizo o edito dicha clase
Fecha	Fecha de creación de la clase

Nota: (Jummp, 2012)

Codificación

Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada.

La codificación debe hacerse ateniendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.

Crear test que prueben el funcionamiento de los distintos códigos implementados nos ayudará a desarrollar dicho código. Crear los test permitirá saber qué es lo que tiene que hacer el sistema una vez implementado

Pruebas

Uno de los pilares fundamentales de XP es la parte de pruebas para esto se debe, realizar u test que sea independiente del código y se deben desarrollar test de forma específica los cuales se irán aplicando según se vaya implementando el sistema (Fases de la Programación Extrema, 2015).

Leguaje de modelado UML

El lenguaje unificado de diagrama o notación (UML) sirve para especificar, visualizar y documentar esquemas de sistemas de software orientado a objetos. UML no es un método de desarrollo, lo que significa que no sirve para determinar qué hacer en primer lugar o cómo diseñar el sistema, sino que simplemente le ayuda a visualizar el diseño y a hacerlo más accesible para otros. UML está controlado por el grupo de administración de objetos (OMG) y es el estándar de descripción de esquemas de software.

UML está diseñado para su uso con software orientado a objetos, y tiene un uso limitado en otro tipo de cuestiones de programación.

UML se compone de muchos elementos de esquematización que representan las diferentes partes de un sistema de software. Los elementos UML se utilizan para crear diagramas, que representa alguna parte o punto de vista del sistema (Acerca de UML, 2015).

Dentro de la serie de diagramas soportados por UML se presentara a continuación los más relevantes, alguno de los cuales se desarrollaran en este proyecto.

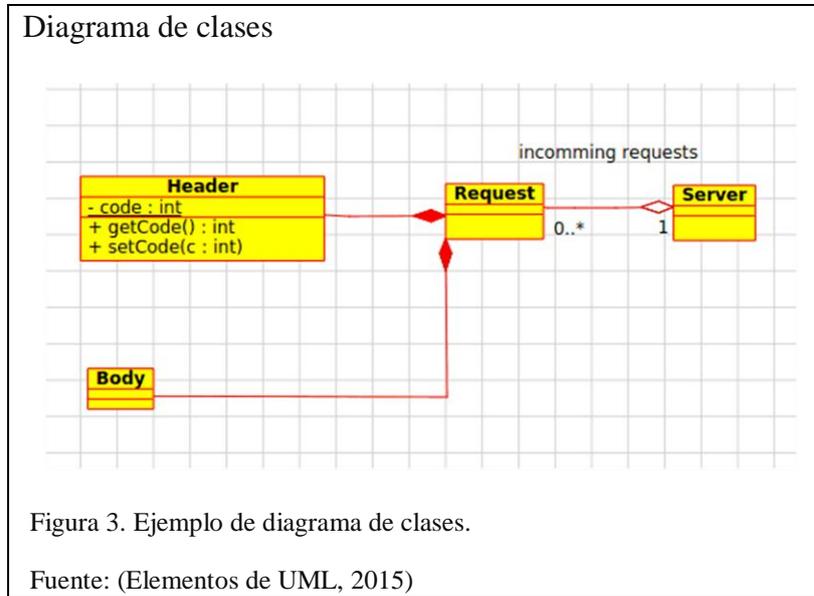
Diagrama de casos de uso

Muestran a los actores (otros usuarios del sistema), los casos de uso (las situaciones que se producen cuando utilizan el sistema) y sus relaciones.

Diagrama de clases

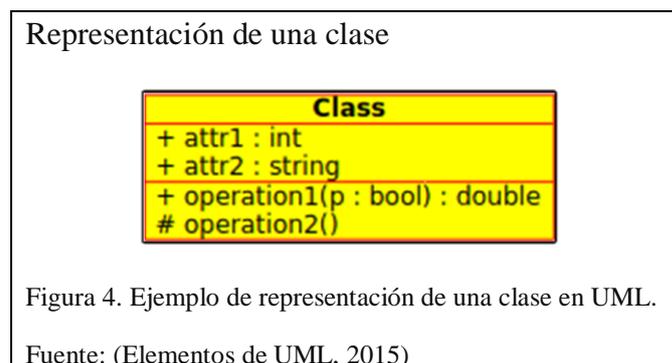
Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que los diagramas de clases son diagramas «estáticos» porque muestran las clases, junto con sus métodos y atributos, así como las

relaciones estáticas entre ellas: qué clases «conocen» a qué otras clases o qué clases «son parte» de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.



Dentro de este tipo de diagrama existen los diferentes elementos que la componen como son:

Clases: Una clase define los atributos y los métodos de una serie de objetos. Todos los objetos de esta clase (instancias de esa clase) tienen el mismo comportamiento y el mismo conjunto de atributos (cada objeto tiene el suyo propio). Las clases se representan por rectángulos con el nombre de la clase, los atributos que esta contiene y los métodos que son utilizados.

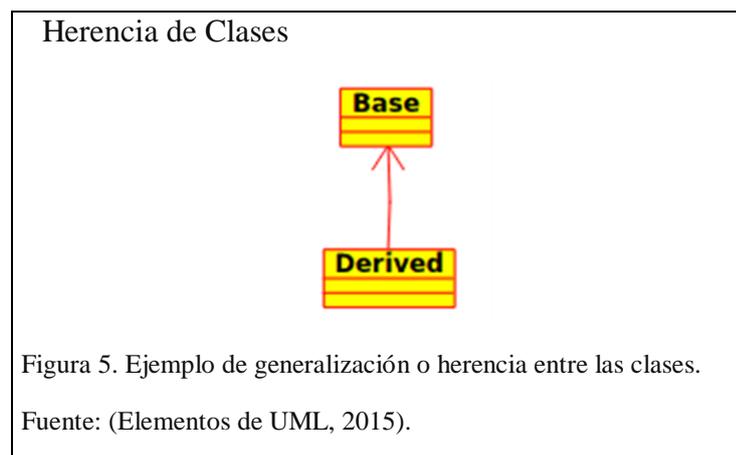


Atributos: Los atributos se muestran al menos con su nombre, y también pueden mostrar su tipo, valor inicial y otras propiedades, los atributos también pueden ser del tipo público (+), protegido (#), privado (-).

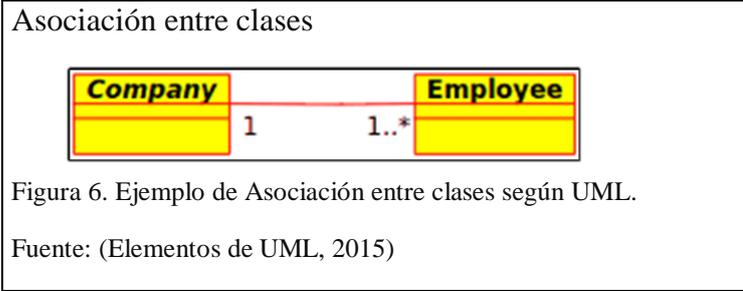
Operaciones: También se representan al menos con el nombre, si es que devuelven algún dato esto debe estar indicado también, como los atributos también pueden ser del tipo público (+), protegido (#), privado (-).

Asociación de clases: las clases se pueden asociar de diferentes maneras según se requiera en el desarrollo.

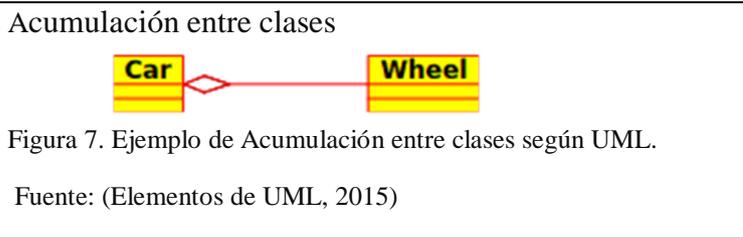
Generalización: en UML la generalización se expresa como una herencia entre las 2 clases en la cual se heredan valores de la clase base, aquí se representa mediante una línea con una flecha apuntando a la clase base.



Asociación: representa la relación entre 2 clases las cuales una depende de la otra ya que permite la comunicación mutua entre los objetos.



Acumulación: representa una relación donde los elementos y atributos de una clase componen una clase mayor.



Composición: es una relación similar a la de acumulación pero de manera más fuerte ya que cada parte de esta relación no puede existir por si misma sino como parte de un todo.

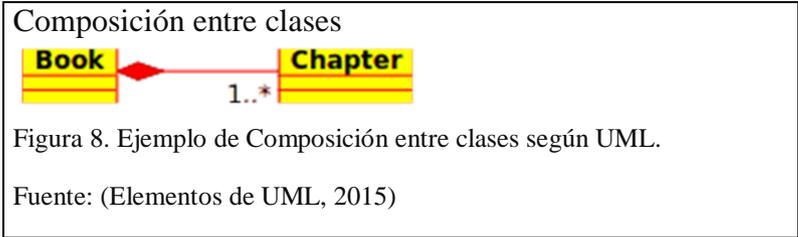


Diagrama de relaciones de entidad

Los diagramas de relaciones de entidad (diagramas ER) muestran el diseño conceptual de las aplicaciones de bases de datos. Representan varias entidades (conceptos) en el sistema de información y las relaciones y restricciones existentes entre ellas. Una extensión de los diagramas de relaciones de entidad llamado «diagramas de relaciones

de entidad extendida» o «diagramas de relaciones de entidad mejoradas» (EER), se utiliza para incorporar las técnicas de diseño orientadas a objetos en los diagramas ER.

Diagrama entidad relación

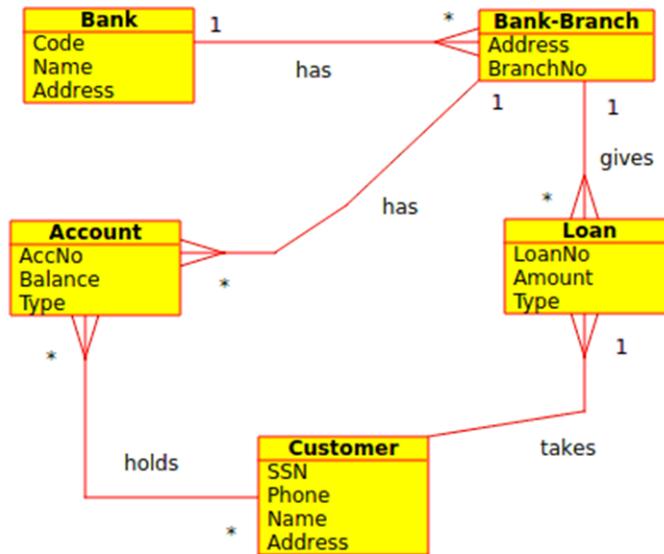
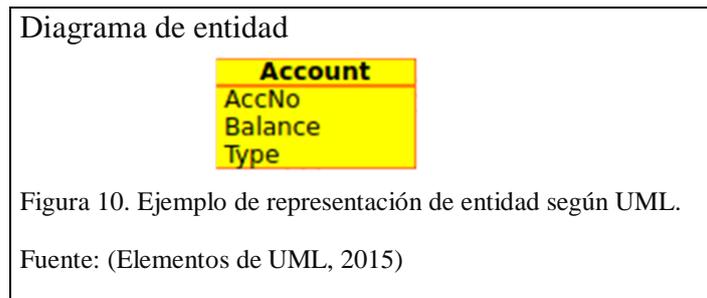


Figura 9. Ejemplo de diagrama entidad relación según UML.

Fuente: (Elementos de UML, 2015)

Dentro de este tipo de diagrama existen los diferentes elementos que la componen como son:

Entidad: Una Entidad es cualquier concepto del mundo real con una existencia independiente. Puede ser un objeto con una existencia física (ejemplo, máquina, robot) o puede ser un objeto con una existencia conceptual (p. ej.: Curso de universidad). Cada entidad tiene un conjunto de atributos que describen las propiedades de la entidad.



Atributos: Los atributos de la entidad se muestran bajo el nombre de la misma.

Restricciones: Las restricciones en los diagramas ER especifican las restricciones de los datos en el esquema de información.

Existen cuatro tipos de restricciones soportadas por este diagrama:

Clave Primaria: es única para la entidad. Solo puede haber una clave primaria en una entidad y ninguno de los atributos que la componen puede ser NULL.

Clave Única: El conjunto de atributos declarados como *única* son únicos para la entidad. Puede haber muchas restricciones únicas en una entidad. Los atributos que lo componen pueden tener el valor NULL. Las claves únicas y primarias identifican de forma única una fila de una tabla (entidad).

Clave externa: referencia también clave foránea es una referencia entre dos tabla donde la clave primaria de una tabla es la clave externa o foránea de otra.

Restricción de comprobación: Una restricción de comprobación (también conocida como restricción de comprobación de tabla) es una condición que define los datos válidos cuando se añaden o actualizan datos en una tabla de la base de datos relacional. Se aplicará una restricción a cada fila de la tabla. La restricción debe ser un predicado. Puede referirse a una o varias columnas de la tabla. Ejemplo: precio \geq 0 (Elementos de UML, 2015).

CAPÍTULO 1

ESTADO DEL ARTE

1.1 Marco Teórico

La necesidad por ser más productivo en el ámbito laboral ha permitido a la tecnología estar a la punta siempre en busca de nuevos desarrollos que permitan realizar cualquier actividad ya sea de trabajo o personal de manera más fácil.

Es por este motivo que las empresas buscan nuevos software para poder automatizar sus procesos y tener un mejor rendimiento para poder solventar las necesidades diarias de sus clientes y ser una empresa que cuente con un alto nivel de competitividad.

Para este proyecto se busca la implementación de sistema que permita el manejo de bodega de la SCPM de manera ágil y eficiente.

En internet existen diferentes programas que realizan esta actividad, muchos de ellos son gratuitos pero todos tienen en común que ofrecen un manejo eficiente de lo existente en bodega.

Algunos de los sistemas de bodega son:

- Mi bodega
- StockControl
- Easy WMS
- Bcase
- VinoTec

Todos estos estos son sistemas ERP

1.1.1 ERP

Una definición sencilla de qué es un ERP (Enterprise Resource Planning – Planificación de Recursos Empresariales) es un conjunto de sistemas de información que permite la integración de ciertas operaciones de una empresa, especialmente las que tienen que ver con la producción, la logística, el inventario, los envíos y la contabilidad.

El ERP funciona como un sistema integrado. Aunque pueda tener menús modulares, es un todo. Es decir, es un único programa con acceso a una base de datos centralizada. Un ejemplo claro se tiene en PROWIN ERP, que además de ser un programa de gestión, está integrado con el programa de contabilidad WINCONTA FINANCIALS, el programa de calidad QUALYPRO,... Los datos se dan de alta sólo una vez y son consistentes, completos y comunes (Informáticos, 2015).

El propósito de un software ERP es apoyar a los clientes de la empresa, dar tiempos rápidos de respuesta a sus problemas, así como un eficiente manejo de información que permita la toma de decisiones minimizando los costes.

1.1.2 Lenguajes de Desarrollo

Visual Studio .Net: es un conjunto completo de herramientas de desarrollo para la construcción de aplicaciones Web ASP, servicios Web XML, aplicaciones para escritorio y aplicaciones móviles. Visual Basic .NET, Visual C++ .NET, Visual C# .NET y Visual J# .NET utilizan el mismo entorno de desarrollo integrado (IDE), que les permite compartir herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes aprovechan las funciones de .NET Framework, que ofrece acceso a tecnologías clave para simplificar el desarrollo de aplicaciones Web ASP y servicios Web XML (Network M. D., 2015).

Microsoft Visual Studio .Net provee adicionalmente versiones de **Framework** para facilitar y simplificar el desarrollo de software. El Framework es un entorno de ejecución administrado que da diversos servicios a aplicaciones en ejecución. Cuenta con un motor de ejecución (Common Language Runtime CLR) que vigila las aplicaciones en que se están ejecutando, y una biblioteca de clases de .NET Framework que proporciona librerías de código garantizado y reutilizable (Network M. D., 2015).

Microsoft SQL Server: es un sistema de administración y análisis de bases de datos relacionales de Microsoft para soluciones de comercio electrónico, línea de negocio y almacenamiento de datos (Network M. D., 2015).

JavaScript: JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, JavaScript es una marca registrada de la empresa Sun Microsystems (LIBROSWEB, 2015).

Jquery: jQuery es una biblioteca gratuita de Javascript, cuyo objetivo principal es simplificar las tareas de creación de páginas web responsivas, acordes a lo estipulado en la Web 2.0, la cual funciona en todos los navegadores modernos. Por otro lado, se dice que jQuery ayuda a concentrarse en gran manera en el diseño del sitio, al abstraer por completo todas las características específicas de cada uno de los navegadores. Otra de las grandes ventajas de jQuery es que se enfoca en simplificar los scripts y en acceder/modificar el contenido de una página web. Finalmente, jQuery agrega una cantidad impresionante de efectos nuevos a Javascript, los cuales podrán ser utilizados en cualquier sitio web a desarrollar (mexired, 2015).

Bootstrap: Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo (ARWEB, 2015).

CSS3: son las siglas de “Cascading Style Sheets”, es decir, “Hojas de estilo en cascada” y es un lenguaje para definir el estilo o la apariencia de las páginas web, escritas con HTML o de los documentos XML, separando el contenido de la presentación. A su vez, permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas

CSS3 es el último estándar de CSS y viene abrazado por HTML5. En un intento de reducir el uso de código Javascript y para estandarizar funciones populares, CSS3 no solo cubre diseño y estilos web sino también forma y movimiento. Desde esquinas

redondeadas y sombras hasta transformaciones y reposicionamiento de elementos ya presentado en pantalla, cada posible efecto aplicado previamente con Javascript fue cubierto. Esto convierte a CSS3 en una tecnología prácticamente inédita comparada con versiones anteriores (Bernal, 2016).

Ajax: Son las siglas de (Asynchronous JavaScript And XML). No es un lenguaje de programación sino un conjunto de tecnologías (HTML-JavaScript-CSS-DHTML-PHP/ASP.NET/JSP-XML) que nos permiten hacer páginas de internet más interactivas.

La característica fundamental de AJAX es permitir actualizar parte de una página con información que se encuentra en el servidor sin tener que refrescar completamente la página. De modo similar se puede enviar información al servidor. (AJAXYA, 2015)

JSON: Son las siglas de (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otro.

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias (Introducción a JSON, 2015).

Arquitectura Modelo Vista Controlador (MVC): Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

- El **Modelo** que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.
- La **Vista**, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste.
- El **Controlador**, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno (Alicante, 2016).

1.1.3 Herramientas de Modelado

Sybase Power Designer: Es la solución de procesos de negocios líder en la industria. Es un software de modelado de datos y gestión de metadatos para la arquitectura de datos, arquitectura de información y arquitectura empresarial (Power Designer, 2015).

Esta herramienta permite realizar diferentes tipos de diagramas los cuales se usan para el diseño de un sistema basados siempre en el mejor rendimiento de la empresa, en esta herramienta se pueden realizar diagramas desde base de datos hasta los diagramas de casos de uso, es muy útil para ver el flujo de la información dentro de un proceso empresarial.

Bizagi Modeler: Es un poderoso modelador de procesos de negocio compatible con el estándar BPMN 2.0, diseñado para mapear, modelar y diagramar todo (bizagi, 2016).

CAPÍTULO 2

ANÁLISIS Y DISEÑO

2.1 Definición del proceso que se va a automatizar

Para poder desarrollar un sistema eficiente que cumpla con las llamadas buenas prácticas se debe partir de los procesos, los cuales están orientados a mostrar el funcionamiento y las actividades necesarias para solventar un problema o una necesidad.

Para este efecto en ayuda con la SCPM se elaboró el siguiente esquema de procesos, con el cual se planea solventar el sistema de una manera estratégica, el cual será la base para llevar a cabo las demás actividades de análisis y diseño. En este esquema se puede observar las diferentes etapas por las que pasa un pedido y actividades que realizan cada uno de los actores que estarán involucrados en el sistema independientemente si tienen relación o no en el proceso de realizar un pedido.

Diagrama del proceso

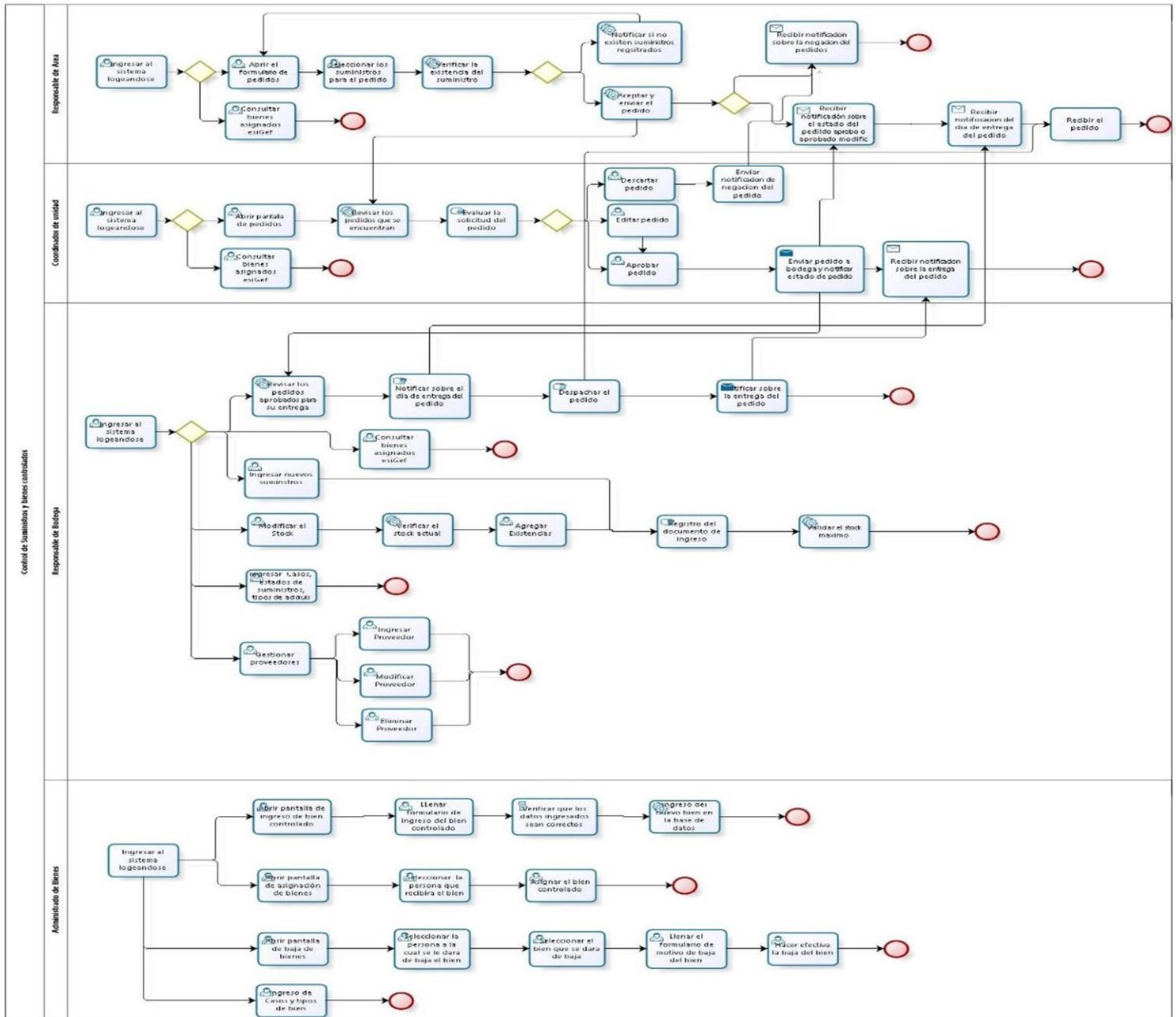


Figura 11. Diagrama del proceso que realizara el sistema.

Elaborado por: Luis Núñez & Daniel Lincango

Con el diagrama expuesto anteriormente se puede realizar la explicación respectiva y así dar forma a la idea principal sobre el funcionamiento del sistema y poder obtener ya la definición de los requerimientos.

1.1.2 Explicación del Diagrama de procesos

Como primer punto en la explicación de la ilustración anterior, se deben de identificar los actores dentro del proceso y posteriormente explicar cuál es su rol en cada uno de ellos y las tareas a las que están asignadas. Para la explicación de la siguiente tabla se presentaran los actores en el orden en que se realizara el proceso de pedido.

Tabla 4. Descripción de las actividades de los actores dentro del proceso.

Actor	Descripción	Tareas dentro del proceso
<p>Responsable del área</p>	<p>Es la persona encargada del área al momento de hacer los pedidos, esta persona puede ser el jefe de área o un delegado.</p>	<p>Abrir y llenar el formulario para realizar un pedido, dicho formulario de pedido fue entregado por la SCPM el cual se encuentra en la sección de ANEXOS (Anexo 2).</p> <p>Una vez hecho el pedido esta persona deberá esperar a que el coordinador de unidad de un pronunciamiento sobre el estado del pedido.</p> <p>Una vez notificado sobre el estado del pedido procederá a esperar a que el responsable de bodega emita una fecha de entrega la cual deberá ser revisada en el correo.</p>

		Recibir y verificar que el pedido este correcto.
--	--	--

Nota: Luis Núñez & Daniel Lincango

Tabla 5. Continuación de la tabla 4.

Actor	Descripción	Tareas dentro del proceso
Coordinador de unidad	Es la persona de más alto rango dentro del proceso, es el encargado de aprobada o negar los pedidos.	<p>Abrir la pantalla de revisión de pedidos y revisar los pedidos especialmente si lo solicitado se ajusta al stock actual.</p> <p>Evaluar la solicitud y emitir una notificación de (Aprobado, Negado o Modificado), al responsable del área que realizo el pedido.</p> <p>Si el pedido esta como aprobado o modificado envía un notificación para el despacho al responsable de bodega.</p> <p>Una vez hecha la notificación a bodega deberá esperar a que bodega emita una respuesta vía correo indicando la fecha del despacho del pedido.</p>

Nota: Luis Núñez & Daniel Lincango

Tabla 6. Continuación de la tabla 4.

Actor	Descripción	Tareas dentro del proceso
Responsable de bodega	Es la persona encargada del área de bodega la cual puede ser el jefe del área o un delegado, ya que este actor realiza más actividades que los anteriores, pueden ser los responsables o encargados 2 o más personas.	<p>En el paso de los pedidos las tareas para este responsable son:</p> <p>Entrar en la pantalla donde revisara los pedidos que se encuentran aprobados para su despacho.</p> <p>Asignar un fecha de despacho y notificar tanto al responsable del área que realizo el pedido como al coordinador de unidad.</p> <p>Supervisar la entrega del pedido al área solicitante.</p> <p>Dentro del proceso las diferentes tareas que debe realizar este se responsable se encuentran de la siguiente manera:</p> <p>Ingresar nuevos suministros.</p> <p>Actualizar el stock de los suministros.</p> <p>Realizar el registro de los suministros con el documento respectivo.</p> <p>Informar al responsable de adquisiciones en caso de que haya escasez de un suministro.</p>

Nota: Luis Núñez & Daniel Lincango

Tabla 7. Continuación de la tabla 4.

Actor	Descripción	Tareas dentro del proceso
Administrador de bienes Controlados	Esta persona es un administrador asignado por administrador del sistema en sí, esta persona puede ser el mismo responsable de bodega o puede ser el coordinador de unidad.	Ingresar un nuevo bien controlado. Asignar el bien controlado al personal de toda la SCPM En caso de devolución del bien controlado se encargara de dar la baja en el sistema y de registrar el motivo de la devolución.

Nota: Luis Núñez & Daniel Lincango

2.2 Definición de requerimientos

Posteriormente a la definición del proceso que va a solventar el sistema mediante la ilustración 12 y posteriormente la explicación de dicho diagrama se da el siguiente paso que es la recolección de los requerimientos basándose en la explicación del diagrama de procesos el cual fue aprobado por la SCPM.

Los requerimientos que se obtuvieron se los realizo mediante las historias de usuario y la información que fue proporcionada por la SCPM, mediante las siguientes historias de usuario se definen los requerimientos en los que se basó el sistema. Cabe mencionar que para esta etapa las historias de usuario reemplazan a los casos de uso en el modelo UML por lo que no es necesario la utilización de dichos diagramas.

Para estas historias se establece un rango de importancia como se muestra en la siguiente figura.

Tabla 8. Tabla de rangos de importancia.

Valor	Prioridad
1-3	Baja
4-7	Media
8-10	Alta

Nota: Luis Núñez & Daniel Lincango

Tabla 9. Historia de usuario 1.

Historia:	Desarrollo en base al modelo MVC
ID:	R1
Descripción/Funcionalidad:	El sistema de debe ser desarrollado en base a la estructura que actualmente maneja la SCPM que es una arquitectura por capas, ya que de no hacerlo así el sistema no podrá ser implementado
Importancia:	10
Como probarlo / Pruebas de aceptación:	La persona de la SCPM encargada de hacer la implementación revisara que la arquitectura concuerde de acuerdo con este requerimiento.

Nota: Luis Núñez & Daniel Lincango

Tabla 10. Historia de usuario 2.

Historia:	Autenticación de usuarios según un rol.
ID:	R2
Descripción/Funcionalidad:	<p>El sistema debe permitir autenticación de usuarios basados en un rol asignado.</p> <p>Los tres tipos de usuarios serán: Administrador, Gestor de pedidos y Supervisor.</p> <p>El sistema permitirá el acceso a las páginas designadas según su rol.</p>
Importancia:	2

Como probarlo / Pruebas de aceptación:	Al estar correcto el logeo se re direccionará a la página principal asignada según el rol de usuario. Este requerimiento una alta importancia ya que la SCPM ya cuenta con una autenticación propia.
--	---

Nota: Luis Núñez & Daniel Lincango

Tabla 11. Historia del usuario 3.

Historia:	Generación de Pedido
ID:	R3
Descripción/Funcionalidad:	<p>El sistema debe permitir crear una interfaz completamente dinámica, creando pestañas inicializadas con el tipo de categoría de los suministros existentes en la tabla SUMINISTROS.</p> <p>En la interfaz en la que el usuario realiza los pedidos se encontraran un botón donde se redireccionara a la pantalla de estado suministro.</p> <p>Los suministros serán visualizados listados con un checkbox y una caja de texto asociados,</p>
Importancia:	10
Como probarlo / Pruebas de aceptación:	Se mostrar la lista de los suministros con diferentes categorías y ahí se procederá a realizar un pedido para verificar que el stock se actualiza y se guarda en la base dicho pedido.

Nota: Luis Núñez & Daniel Lincango

Tabla 12. Historia del usuario 4.

Historia:	Revisión de los estados de pedidos
ID:	R4
Descripción/Funcionalidad:	<p>Aquí se desplegara todos los pedidos que se han realizado por distintos usuarios del sistema.</p> <p>Tendrá un botón de detalle del pedido que se desplegará en una ventana modal.</p>
Importancia:	8
Como probarlo / Pruebas de aceptación:	<p>Todos los pedidos realizados se deben mostrar en una pantalla, también se deben mostrar los pedidos que se han aprobado, negado o modificado alguno de sus items</p>

Nota: Luis Núñez & Daniel Lincango

Tabla 13. Historia del usuario 5.

Historia:	Aprobación de Pedidos
ID:	R5
Descripción/Funcionalidad:	<p>De este procesos se encarga una persona de administración esta persona revisara el pedido y vera si el stock que se tiene es el suficiente para el despacho, aquí también se podrá modificar la cantidad de un suministro solicitado y negar un suministro o todo el pedido. En caso de negar o modificar se puede poner un mensaje del motivo de esta operación.</p>
Importancia:	10
Como probarlo / Pruebas de aceptación:	<p>Los pedidos se mostraran en una pantalla con un botón donde al dar clic se dará un detalle del pedido</p>

	en una moda. En la modal se procederá a aprobar todo el pedido, modificar y negar uno o varios suministros.
--	---

Nota: Luis Núñez & Daniel Lincango

Tabla 14. Historia del usuario 6.

Historia:	Gestión de inventario
ID:	R6
Descripción/Funcionalidad:	Este requerimiento debe realizar la generación de reportes a manera de kardex , Ingreso de nuevos suministros, registro de documento de ingreso, ingreso de nuevos proveedores, categorías, subcategorías, consulta de los pedidos que se van a despachar
Importancia:	10
Como probarlo / Pruebas de aceptación:	Todos los mantenedores de categorías, subcategorías deben registrar datos y modificar datos.

Nota: Luis Núñez & Daniel Lincango

Tabla 15. Historia del usuario 7.

Historia:	Gestión de bienes controlados
ID:	R7
Descripción/Funcionalidad:	El requerimiento es similar a la gestión de inventario pero aquí el administrador del sistema registrar un nuevo bien controlado, podrá también asignar bienes controlados al resto del personal y dar de baja el bien en caso de que fuera necesario.
Importancia:	10
Como probarlo / Pruebas de aceptación:	Se debe probar que se ingrese un nuevo bien controlado, que se pueda asignar a cualquier persona y

	que posteriormente se pueda retirar este bien a esa persona, debe contar con los respectivos mensajes de advertencia en caso de que no existan bienes controlados.
--	--

Nota: Luis Núñez & Daniel Lincango

Tabla 16. Historia del usuario 8.

Historia:	Despacho de Pedidos
ID:	R8
Descripción/Funcionalidad:	Este requerimiento ira dentro del módulo de administración de bodega aquí al administrador de bodega le llegara una notificación indicando los pedidos que ya se encuentran aprobados para el despacho, en una pantalla se mostraran los pedidos por despachar y despachados, para poder despachar los pedidos se deberá ingresar una fecha y esta se guardara en la base de datos
Importancia:	10
Como probarlo / Pruebas de aceptación:	Que se muestren los pedidos que se encuentran despachados y por despachar en un calendario, poder asignar una fecha de despacho a un pedido aprobado.

Nota: Luis Núñez & Daniel Lincango

2.3 Diagrama de clases

Este diagrama explica las relaciones entre las clases del sistema y las variables que se utiliza en cada clase junto con los métodos que cada una contiene.

Autores	Daniel Lincango, Luis Núñez
Fecha	10 de octubre del 2015

Nota: Luis Núñez & Daniel Lincango

Tabla 18. Tabla CRC clase Casos.

Nombre de la clase	clsCasos
Las responsabilidades de la clase.	Administrar los métodos necesarios para la gestión de los casos por los que se registra un nuevo bien o suministro.
Las clases con las que va a colaborar.	ClsBienes, ClsSuministro
Autores	Daniel Lincango, Luis Núñez
Fecha	13 de septiembre de 2015

Nota: Luis Núñez & Daniel Lincango

Tabla 19. Tabla CRC clase Categoría.

Nombre de la clase	clsCategoría
Las responsabilidades de la clase.	Administrar los métodos necesarios para la gestión de las categorías para el registro de un nuevo bien o suministro.
Las clases con las que va a colaborar.	ClsSubCategoría
Autores	Daniel Lincango, Luis Núñez
Fecha	23 de julio de 2015

Nota: Luis Núñez & Daniel Lincango

Tabla 20. Tabla CRC clase Condición.

Nombre de la clase	clsCondicion
Las responsabilidades de la clase.	Administrar los métodos necesarios para la gestión de las condiciones físicas en las que se encuentra un bien o suministro.
Las clases con las que va a colaborar.	ClsBienes, ClsSuministro
Autores	Daniel Lincango, Luis Núñez

Fecha	23 de julio de 2015
-------	---------------------

Nota: Luis Núñez & Daniel Lincango

Tabla 21. Tabla CRC clase Documentos.

Nombre de la clase	clsDocumentos
Las responsabilidades de la clase.	Administrar los métodos necesarios para la el registro de los documentos con los que se ingresó uno o varios suministros.
Las clases con las que va a colaborar.	ClsPedido
Autores	Daniel Lincango, Luis Núñez
Fecha	10 de octubre de 2015

Nota: Luis Núñez & Daniel Lincango

Tabla 22. Tabla CRC clase Entrega Pedidos.

Nombre de la clase	clsEntregaPedidos
Las responsabilidades de la clase.	Revisión de pedidos aprobados por estado y usuario, asignación de las fechas de entrega.
Las clases con las que va a colaborar.	clsKardex
Autores	Daniel Lincango, Luis Núñez
Fecha	4 de octubre de 2015

Nota: Luis Núñez & Daniel Lincango

Tabla 23. Tabla CRC Estado Pedidos.

Nombre de la clase	clsEstadoPedidos
Las responsabilidades de la clase.	Revisión del estado de los pedidos realizados.
Las clases con las que va a colaborar.	clsPedido
Autores	Daniel Lincango, Luis Núñez
Fecha	28 de septiembre de 2015

Nota: Luis Núñez & Daniel Lincango

Tabla 24. Tabla CRC clase Formas de Adquisición.

Nombre de la clase	clsFormasAdquisicion
Las responsabilidades de la clase.	Adminstras las formas en las que se adquirió el suministro o bien controlado
Las clases con las que va a colaborar.	clsBienes,ClsSuministro
Autores	Daniel Lincango, Luis Núñez
Fecha	4 de octubre de 2015

Nota: Luis Núñez & Daniel Lincango

Tabla 25. Tabla CRC clase Kardex.

Nombre de la clase	clsKardex
Las responsabilidades de la clase.	Administra los ingresos y egresos del suministro
Las clases con las que va a colaborar.	Ninguno
Autores	Daniel Lincango, Luis Núñez
Fecha	10 de octubre de 2015

Nota: Luis Núñez & Daniel Lincango

Tabla 26. Tabla CRC clase Marca.

Nombre de la clase	clsMarcas
Las responsabilidades de la clase.	Adminstras las marcas de los suministros que se registraran junto con otras características.
Las clases con las que va a colaborar.	clsSubcategoria
Autores	Daniel Lincango, Luis Núñez
Fecha	19 de octubre de 2015

Nota: Luis Núñez & Daniel Lincango

Tabla 27. Tabla CRC clase Materiales.

Nombre de la clase	clsMateriales
Las responsabilidades de la clase.	Administrar los materiales de los que está compuesto un suministro.
Las clases con las que va a colaborar.	clsSubCategoria
Autores	Daniel Lincango, Luis Núñez
Fecha	20 de octubre de 2015

Nota: Luis Núñez & Daniel Lincango

Tabla 28. Tabla CRC clase Pedido.

Nombre de la clase	clsPedido
Las responsabilidades de la clase.	Realizar los pedidos necesarios.
Las clases con las que va a colaborar.	clsKardex
Autores	Daniel Lincango, Luis Núñez
Fecha	27 de julio de 2015

Nota: Luis Núñez & Daniel Lincango

Tabla 29. Tabla CRC clase Proveedor.

Nombre de la clase	clsProveedor
Las responsabilidades de la clase.	Administrar la información de los proveedores, ingresar o dar de baja un proveedor
Las clases con las que va a colaborar.	clsKardex
Autores	Daniel Lincango, Luis Núñez
Fecha	29 de julio de 2015

Nota: Luis Núñez & Daniel Lincango

Tabla 30. Tabla CRC clase Revisión Pedido.

Nombre de la clase	clsRevisionPedido
Las responsabilidades de la clase.	Revisar los pedidos para ser aprobados, modificados o negados.
Las clases con las que va a colaborar.	clsKardex
Autores	Daniel Lincango, Luis Núñez
Fecha	4 de octubre de 2015

Nota: Luis Núñez & Daniel Lincango

Tabla 31. Tabla CRC clase Subcategoría.

Nombre de la clase	clsSubCategoria
Las responsabilidades de la clase.	Administrar las subcategorías en las que se encuentran los suministros.
Las clases con las que va a colaborar.	clsBienes, ClsSuministro
Autores	Daniel Lincango, Luis Núñez
Fecha	20 de octubre de 2015

Nota: Luis Núñez & Daniel Lincango

Tabla 32. Tabla CRC clase Suministros.

Nombre de la clase	clsSuministros
Las responsabilidades de la clase.	Gestionar los suministros
Las clases con las que va a colaborar.	clsKardex
Autores	Daniel Lincango, Luis Núñez
Fecha	23 de julio del 2015

Nota: Luis Núñez & Daniel Lincango

Tabla 33. Tabla CRC clase Tipo Bien.

Nombre de la clase	clsTipoBien
Las responsabilidades de la clase.	Adminstras los tipos de bienes con que serán registrados los suministros o bienes controlados
Las clases con las que va a colaborar.	clsCategoria
Autores	Daniel Lincango, Luis Núñez
Fecha	23 de octubre de 2015

Nota: Luis Núñez & Daniel Lincango

Tabla 34. Tabla CRC clase Utilitario Suministro.

Nombre de la clase	clsUtilitarioSuministro
Las responsabilidades de la clase.	Administración de métodos genéricos utilizados por otras clases
Las clases con las que va a colaborar.	Todas las clases
Autores	Daniel Lincango, Luis Núñez
Fecha	23 de octubre de 2015

Nota: Luis Núñez & Daniel Lincango

CAPÍTULO 3

CONSTRUCCION Y PRUEBAS

3.1 Herramientas para el desarrollo

Las herramientas usadas en la construcción de este sistema fueron establecidas por la SCPM y se detallan a continuación.

- SQLServer R2 2008 como motor de base de base de datos
- Como lenguaje principal para el desarrollo principal C#
- Microsoft Visual Studio como IDE principal en la elaboración del sistema.

Herramientas extras:

- Javascript como modelador complementario en el desarrollo del lado del cliente
- Bootstrap en versión 3.3.6 para el diseño interfaz web.

3.2 Diagrama de entidad relación

El diagrama entidad relación presentado a continuación presenta el esquema de la base de datos que actualmente se encuentra funcional para este proyecto, en este esquema la tabla de personal será sustituida en la SCPM una vez implementado el sistema, por la tabla de personal que posea la SCPM.

Cabe mencionar que esta base es totalmente independiente de la base global que maneja la SCPM, el único nexo entre la base actual de la SCPM y la base de este proyecto es la tabla “SCPM_PERSONAL” anteriormente mencionada.

Tabla 35. Descripción de las tablas de la base de datos.

TABLA	DESCRIPCIÓN
BIENES_CONTROLADOS	Almacena toda la información y datos respecto a los bienes controlados que tiene la SCPM
BIENES_FUNCIONARIO	Almacena un registro de los funcionarios de la SCPM y los bienes controlados que se encuentran a su cargo
CASO	Guarda los motivos por los que se ingresa un nuevo suministro o bien controlado
CATEGORIA	Almacena las categorías en las que se encuentran catalogados los suministros
CONDICION	Almacena la condición física en que se encuentra un suministro o un bien controlado.
FORMA_ADQUISICION	Almacena la forma en cómo se adquirieron los suministros o bienes controlados, esto puede ser comprados, donados, etc.
INDICES	Es una tabla independiente que almacena el índice para el registro de un nuevo bien controlado o un suministro.
INGRESOS_DOCUMENTOS	Registra los detalles del documento con el que se hizo el ingreso de nuevos suministros o la actualización de los mismos.
KARDEX	Almacena los datos de las transacciones de los suministros ingresos y egresos.

Nota: Luis Núñez & Daniel Lincango

Tabla 36. Continuación de la tabla 35.

TABLA	DESCRIPCIÓN
MARCA	Almacena detalles y características de las marcas de los suministros.
MATERIALES	Almacena detalles y características de los materiales de los suministros.
PEDIDO	Contiene los detalles de los pedidos que se han realizado.
PROVEEDOR	Almacena la información pertinente sobre el proveedor de los suministros.
SCPM_PERSONAL	Esta tabla contiene los datos personales de los funcionarios de la institución (esta tabla será reemplaza por la de la SCPM).
SUB_CATEGORIAS	Almacena las subcategorías en las que se encuentran los suministros.
SUMINISTROS	Guarda la información detallada de los suministros.
TIPO_BIEN	Almacena si la información sobre si lo que se va a ingresar es un suministro o bien controlado.

Nota: Luis Núñez & Daniel Lincango

3.2.2 Diccionario de Datos

Se procede a detallar los campos de cada de una de las tablas de la base de datos.

Tabla 37. Diccionario de datos BIENES_CONTROLADOS.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID_BIEN_CONTROLADO	Int (not null)	Clave primaria de BIENES_CONTROLADOS.
ID_CONDICION	Int (not null)	Clave foránea de la tabla CONDICION.
ID_SUB_CATEGORIA	Int (not null)	Clave foránea de SUB_CATEGORIA.
ID_FORMA_ADQUISICION	Int (not null)	Clave foránea de la tabla FORMA_ADQUISICION.
ID_CASO	Int (not null)	Clave foránea de la tabla de CASO.
COD_BIEN	Varchar (100,not null)	Es el código que pone la SCPM a bien controlado.
COD_BIEN_ANTERIOR	Varchar (100,not null)	Es el código anterior con el cual contaba el bien controlado.
SERIE	Varchar (100,not null)	Es la serie del bien controlado.
MODELO	Varchar (100,not null)	Es modelo del bien controlado.
ESTADO	Int(not null)	Indica si bien controlado se encuentra en estado 19(activado), 399(desactivado).
FECHA_INGRESO	Date(not null)	Es la fecha con la que se ingresó el bien controlado.
OBSERVACION	Varchar (250, not null)	Guarda una observación para el bien controlado si fuera necesario.
VIDA_UTIL	Int (not null)	Indica la vida útil del bien controlado.

LUGAR	Varchar (100, not null)	Indica el lugar del bien controlado.
--------------	-------------------------	--------------------------------------

Nota: Luis Núñez & Daniel Lincango

Tabla 38. Diccionario de datos de BIENES_FUNCIONARIO.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID_BIEN_FUNCIONARIO	Int (not null)	Clave primaria de BIENES_FUNCIONARIO.
ID_BIEN_CONTROLADO	Int (not null)	Clave foránea de la tabla BIENES_CONTROLADOS.
ID_FUNCIONARIO	Int (not null)	Clave foránea de la tabla de SCPM_PERSONAL.
ESTADO	Int (not null)	Indica si bien controlado se encuentra en estado. 19(activado), 399(desactivado)
FECHA_ENTREGA	Date (not null)	Fecha en la que se le asigna el bien al funcionario.
FECHA_DEVOLUCION	Date	Fecha en la que se entrega el bien por parte del funcionario.

Nota: Luis Núñez & Daniel Lincango

Tabla 39. Diccionario de datos de CASOS.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID_CASO	Int (not null)	Clave primaria de CASOS.
NOM_CASO	Varchar (50 not null)	Nombre del caso ingresado

Nota: Luis Núñez & Daniel Lincango

Tabla 40. Diccionario de datos de CATEGORIA.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID_CATEGORIA	Int (not null)	Clave primaria de CATEGORIA.
ID_BIEN	Int (not null)	Clave foránea de la tabla TIPO_BIENES.
NOM_CATEGORIA	Varchar (250, not null)	Nombre de la categoría ingresada.

Nota: Luis Núñez & Daniel Lincango

Tabla 41. Diccionario de datos de CONDICION.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID_CONDICION	Int (not null)	Clave primaria de CONDICION.
NOM_CONDICION	Varchar (50, not null)	Nombre de la condición.

Nota: Luis Núñez & Daniel Lincango

Tabla 42. Diccionario de datos de FORMA_ADQUISICION.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID_FORMA_ADQUISICION	Int (not null)	Clave primaria de FORMA_ADQUISICION.
NOM_ADQUISICION	Varchar (50, not null)	Nombre la forma de adquisición.

Nota: Luis Núñez & Daniel Lincango

Tabla 43. Diccionario de datos de INDICES.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID_INDICE	Int (not null)	Clave primaria de INDICES.
NOM_INDICE	Varchar (50, not null)	Indica si el índice es para suministros o bienes controlados.
VALOR	Int (not null)	Indica en que valor se encuentra el índice.
ESTADO	Int (not null)	Indica el estado del índice 19(activado), 399 (desactivado).

Nota: Luis Núñez & Daniel Lincango

Tabla 44. Diccionario de datos de INGRESO DOCUMENTOS.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID_INGRESO	Int (not null)	Clave primaria de INGRESO_DOCUMENTOS.
ID_SUMINISTROS	Int (not null)	Clave foránea de la tabla SUMINISTROS.
ID_PROVEEDOR	Int (not null)	Clave foránea de la tabla de PROVEEDOR.
FECHA_INGRESO	Date(not null)	Fecha de ingreso del registro.
OBSERVACION	Varchar(250)	Guarda una observación al registro en caso de que sea necesaria
NUM_FACTURA	Varchar (15)	Guarda el número de la factura con que se registra.
CANTIDAD_INGRESO	Int (not null)	Guarda la cantidad del suministro registrado.

Nota: Luis Núñez & Daniel Lincango

Tabla 45. Continuación tabla 44.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
COSTO_FACTURA	Float (not null)	Saca el costo total del suministro registrado.

Nota: Luis Núñez & Daniel Lincango

Tabla 46. Diccionario de datos de KARDEX.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID_KARDEX	Int (not null)	Clave primaria de KARDEX.
ID_SUMINISTRO	Int (not null)	Clave foránea de la tabla SUMINISTRO.
FECHA_KARDEX	date (not null)	Fecha de registro del kardex.
TIPO_TRANS_KARDEX	Varchar (15,not null)	Indica si la transacción es de ingreso o egreso.
SALDO_KARDEX	Int (not null)	Indica cual es el saldo actual del suministro.
COSTO_KARDEX	Float (not null)	Indica el costo actual de ese suministro.
CANTIDAD	Int (not null)	Indica la cantidad que se ingresó o egreso del suministro.
MOTIVO	Varchar (250,not null)	Motivo por el cual se hizo la transacción.

Nota: Luis Núñez & Daniel Lincango

Tabla 47. Diccionario de datos de MARCA.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID_MARCA	Int (not null)	Clave primaria de MARCA.
NOM_MARCA	Varchar (50,not null)	Nombre de la marca.
COLOR	Varchar (50,not null)	Color producto.
DIMENSIONES	Varchar (100,not null)	Dimensiones producto.

Nota: Luis Núñez & Daniel Lincango

Tabla 48. Diccionario de datos de MATERIALES.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID_MATERIAL	Int (not null)	Clave primaria de MATERIALES.
NOM_MATERIAL	Varchar (50,not null)	Nombre del material.

Nota: Luis Núñez & Daniel Lincango

Tabla 49. Diccionario de datos de PEDIDOS.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID_PEDIDO	Int (not null)	Clave primaria de PEDIDOS.
COD_PEDIDO	BigInt (not null)	Codigo del pedido.
ID_SUMINISTRO	Int (not null)	Clave foránea de la tabla de SUMINISTROS
PER_COD	Int (not null)	Clave foránea de la tabla SCPM_PERSONAL. Es la clave de la persona que realizo el pedido.
CANTIDAD_PEDIDO	Int (not null)	Cantidad que se solicitó de un suministro.

Nota: Luis Núñez & Daniel Lincango

Tabla 50. Continuación tabla 49.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ESTADO_PEDIDO	Varchar (100,not null)	Estado en que se encuentra el pedido: pendiente, aprobado, negado.
FECHA_EMISION	Date (not null)	Fecha de emisión del pedido.
FECHA_REVISION	Date	Fecha en la que se revisa si el pedido es válido.
FECHA_ENTREGA	Date	Fecha de entrega del pedido.
PER_COD_APRUEBA	Int	Clave del funcionario que revisa si el pedido se aprueba, modifica o niega
MENSAJE	Varchar (300)	Observación que se hace al pedido de ser necesario.
VISTO	Varchar (2)	Indica si el pedido ya ha sido revisado por la persona que hizo el pedido.
ESTADO_SUMINISTRO	Varchar (15)	Indica si está aprobado el suministro ya que la aprobación se hace por ítem.
PER_COD_DESPACHO	Int	Clave de la persona que despacha el pedido.
CANTIDAD_MODIFICADA	Int	En caso de modificado la cantidad solicitada.

Nota: Luis Núñez & Daniel Lincango

Tabla 51. Diccionario de datos de PROVEEDOR.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID_PROVEEDOR	Int (not null)	Clave primaria de PROVEEDOR
RUC	Int (not null)	Clave foránea de la tabla CONDICION.
NOMBRE	Int (not null)	Clave foránea de la tabla de SUB_CATEGORIA.
DIRECCION	Int (not null)	Clave foránea de la tabla FORMA_ADQUISICION.
ACTIVIDAD_ECONOMICA	Int (not null)	Clave foránea de la tabla de CASO.
PROPIETARIO	Varchar (100,not null)	Es el código que pone la SCPM a bien controlado.
CORREO	Varchar (100,not null)	Es el código anterior con el cual contaba el bien controlado.
CONTACTOS	Varchar (100,not null)	Es la serie del bien controlado.
ESTADO_PROVEEDOR	Varchar (100,not null)	Es modelo del bien controlado.

Nota: Luis Núñez & Daniel Lincango

Aunque esta tabla es provisional hasta el momento de la implementación en la SCPM ya que esta tiene una propia, también es fundamental hacer el diccionario de datos por la importancia que representa para este proyecto.

Tabla 52. Diccionario de datos de SCPM_PERSONAL.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
PER_COD	Int (not null)	Clave primaria de SCPM_PERSONAL.
NOMB_PERSONAL	Varchar (30,not null)	Nombres del funcionario.
APELLIDO_PERSONAL	Varchar (30,not null)	Apellidos del funcionario.
CARGO_PERSONAL	Varchar (30,not null)	Cargo que desempeña en la institución.
AREA_DEPENDENCIA	Varchar (30,not null)	Área a la que pertenece el funcionario.
ROL_PERSONAL	Varchar (15,not null)	Rol del funcionario que desempeña dentro del área.
PASSWORD	Varchar (25not null)	Password para el login del funcionario.
MAIL	Varchar (30,not null)	Mail para contacto del funcionario.
USUARIO	Varchar (30)	Usuario para el login del funcionario.

Nota: Luis Núñez & Daniel Lincango

Tabla 53. Diccionario de datos de SUB_CATEGORIA.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID_SUB_CATEGORIA	Int (not null)	Clave primaria de SUB_CATEGORIA.
ID_CATEGORIA	Int (not null)	Clave foránea de la tabla CATEGORIA.
ID_MARCA	Int (not null)	Clave foránea de la tabla de MARCA.
ID_MATERIAL	Int (not null)	Clave foránea de la tabla MATERIAL.
NOM_SUB_CATEGORIA	Varchar (225,not null)	Nombre de la subcategoría ingresada.

Nota: Luis Núñez & Daniel Lincango

Tabla 54. Diccionario de datos de SUMINISTROS.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID_SUMINISTRO	Int (not null)	Clave primaria de SUMINISTROS.
ID_SUB_CATEGORIA	Int (not null)	Clave foránea de la tabla SUB_CATEGORIA.
ID_ESTADO_SUMINISTRO	Int (not null)	Clave foránea de la tabla de CONDICION.
DESCRIPCION_SUMINISTRO	Varchar (255,not null)	Descripción del suministro.
PRESENTACION	Varchar (25,not null)	Forma de presentación del suministro, ejemplo: unidades, cajas, litros, galones.
EXT_MINIMA	Int (not null)	Existencia mínima del suministro.
EXT_MAXIMA	Int (not null)	Existencia máxima del suministro.

Nota: Luis Núñez & Daniel Lincango

Tabla 55. Continuación de la tabla 54.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
COSTO_ACTUAL	Float (not null)	Costo del suministro actualmente.
SALDO_ACTUAL	Int (not null)	Saldo en que se encuentra el suministro.
ESTADO	Int (not null)	Indica el estado del suministro 19(activado), 399(desactivado).

Nota: Luis Núñez & Daniel Lincango

Tabla 56. Diccionario de datos de TIPO BIENES.

NOMBRE COLUMNA	TIPO DE DATO	DESCRIPCIÓN
ID_BIEN	Int (not null)	Clave primaria de TIPO_BIENES.
TIPO_BIEN	Varchar (100,not null)	Nombre del tipo de bien.
ESTADO	Varchar (10,not null)	Indica si el suministro se encuentra en estado 1(activado), 399(desactivado).
PROGRAMACION	Varchar (50,not null)	Indica si el tipo de bien pertenece a suministros o bienes controlados.

Nota: Luis Núñez & Daniel Lincango

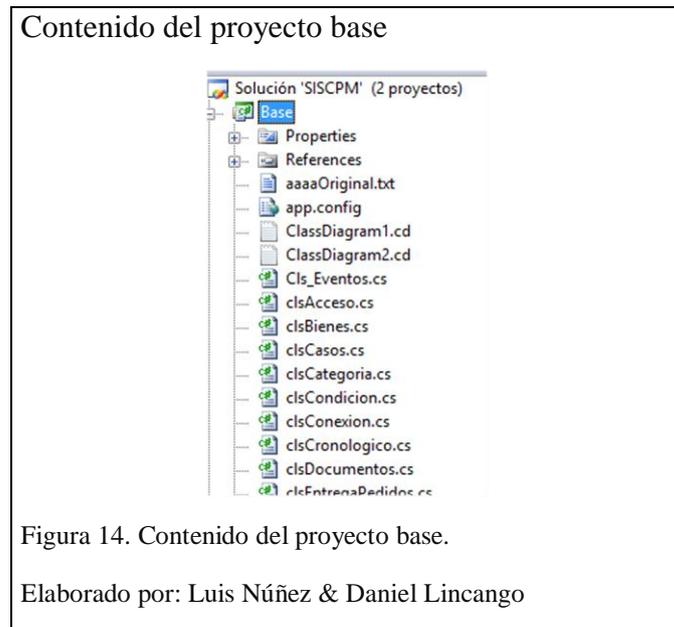
3.3 Construcción de la Estructura MVC

Como se planteó en la sección **1.1.2 Lenguajes de Desarrollo y Estructura del Marco Teórico**, la estructura MVC permite crear un sistema de manera ordenada y eficiente, siguiendo las buenas practicas ya que divide al software en 3 capas esenciales.

3.3.1 Capa Modelo

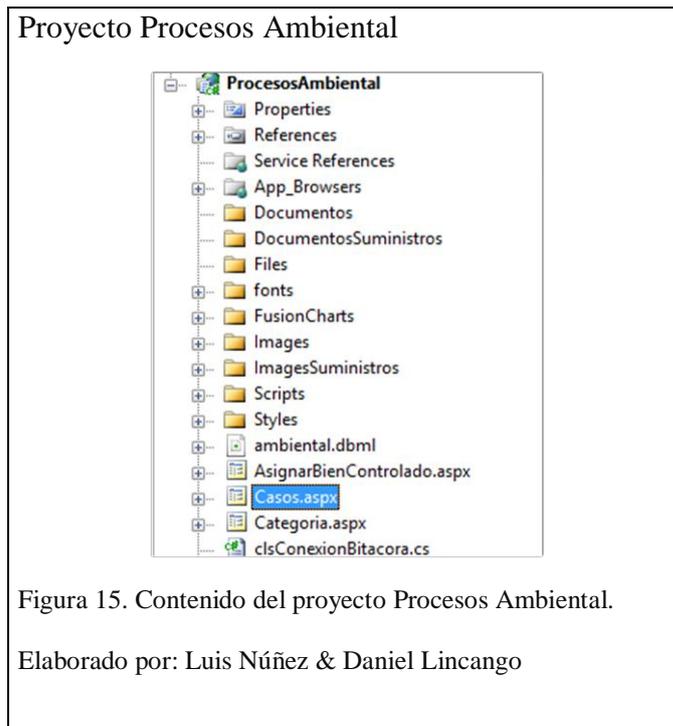
Dentro la arquitectura MVC, la capa modelo es en la cual se tiene acceso a datos, mediante procedimientos almacenados previamente creados. Por tal motivo se creó el

proyecto de nombre Base el cual está constituido de archivos con la extensión .cs los cuales representan clases, que serán utilizadas posteriormente por el controlador.



3.3.2 Capa de Vista

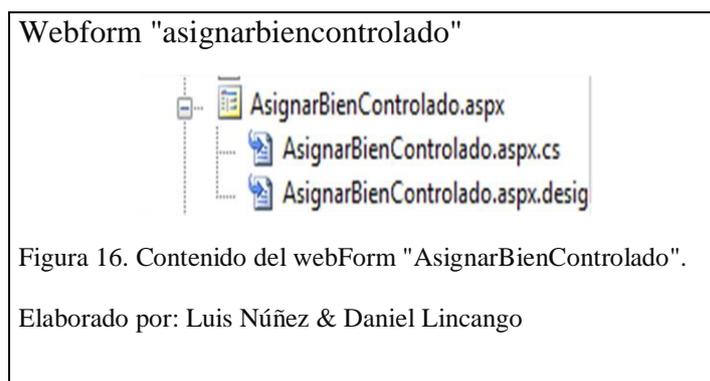
Es la capa de presentación es decir la que va a interactuar con el cliente, la misma que se encuentra separada del modelo por tal motivo esta creado un proyecto llamado “Procesos Ambiental” en donde se encuentran los archivos de extensión .aspx.



3.3.3 Capa de controlador

Como se observó en la sección **1.1.2 Lenguajes de Desarrollo y Estructura del Marco Teórico** indica que la capa de controlador es la que interactúa entre los datos y la presentación dentro de la aplicación, esta capa está relacionada con las reglas del negocio, por ende en esta capa es donde existe el código de mayor importancia.

Para dar un ejemplo de la estructura del controlador se tomara como ejemplo el webForm “AsignarBienControlado”.



Como se puede observar el webForm contiene 2 archivos. El archivo “AsignarBienControlado.aspx.designer.cs” contiene todos componentes del formulario, en otras palabras es la conexión la vista dentro de la arquitectura MVC. El otro archivo “AsignarBienControlado.aspx.cs” hace referencia y utiliza los componentes del formulación para obtener los datos necesarios para insertar la BDD, también permite usar los componentes como contenedor de datos provenientes de la BDD, por tal motivo esta la conexión con el modelo dentro de la arquitectura MVC.

3.4 Construcción de la aplicación web.

La principal necesidad de una solución informática por parte del personal que administra actualmente la bodega de la SCPM, es la base fundamental para la construcción de una aplicación web en donde se optó por construir por medio de las herramientas Microsoft Visual Studio 2008 como entorno de desarrollo (Figura 17) y Microsoft SQL Server 2008 R2 como motor de base de datos (Figura 18).

<p>Logo de Visual Studio</p>  <p>Microsoft Visual Studio</p> <p>Figura 17. Logo de Visual Studio. Fuente: (Network M. D., 2015)</p>	<p>Logo de SQLServer</p>  <p>Microsoft SQL Server 2008 R2</p> <p>Figura 18. Logo de Microsoft SQLServer. Fuente: (Network M. D., 2015)</p>
---	---

La aplicación se encuentra constituida por nueve módulos que son:

- Administración de bienes asignados.

- Administración de categorías.
- Administración de proveedores.
- Administración de suministros.
- Entrega pedidos.
- Estado de pedidos.
- Reportes de suministros.
- Revisión Pedidos.
- Solicitud de suministros.

3.4.1 Acceso a datos

Una vez instalados los programas principales de desarrollo, dentro de la aplicación web el acceso a datos de encuentra definido por procedimientos almacenados como requerimiento del usuario.

Los procedimientos almacenados fueron creados directamente en la base de datos bajo scripts los cuales son llamados por métodos de inserción (Figura 19), búsqueda (Figura 20) y actualización (Figura 21) creados dentro de las clases existentes en el proyecto “Base” como parte del modelo.

Método para inserción de datos

```
//INGRESO NUEVO PEDIDO
public static void
    ingresarPedido(int codPedido, int idSuministro, int perCod, int cantidad)
{
    SqlConnection conn = null;
    var cad = new clsConexion();
    conn = new SqlConnection(cad.conexion());
    conn.Open();
    DateTime fechaActual = DateTime.Today;
    SqlCommand cmd = new SqlCommand("ingresarNuevoPedido", conn);
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Parameters.Add("@idPedido", SqlDbType.Int).Value =
        Base.clsUtilitarioSuministros.getUltimoID("obtenerUltimoIDPedido");
    cmd.Parameters.Add("@codPedido", SqlDbType.Int).Value = codPedido;
    cmd.Parameters.Add("@idSuministro", SqlDbType.Int).Value = idSuministro;
    cmd.Parameters.Add("@perCod", SqlDbType.Int).Value = perCod;
    cmd.Parameters.Add("@cantidad", SqlDbType.Int).Value = cantidad;
    cmd.Parameters.Add("@fechaEmision", SqlDbType.Date).Value = fechaActual;
    cmd.ExecuteNonQuery();
    conn.Close();
}
```

Figura 19. Código ejemplo de inserción de datos mediante SP.

Elaborado por: Luis Núñez & Daniel Lincango

Método para obtener datos

```
//METODO PRA MOSTRAR DATOS DE PERSONAL
public static List<Object> getDatosPersonal ()
{
    List<Object> persona = new List<Object>();
    SqlConnection conn = null;
    var cad = new clsConexion();
    conn = new SqlConnection(cad.conexion());
    conn.Open();
    SqlCommand cmd = new SqlCommand("obtenerDatosPersonal", conn);
    cmd.CommandType = CommandType.StoredProcedure;
    SqlDataReader reader = cmd.ExecuteReader();

    if (reader.HasRows)
    {
        while (reader.Read())
        {
            persona.Add(new personal ()
            {
                perCod = Convert.ToString(reader[0]),
                nombre = Convert.ToString(reader[1]),
                apellido = Convert.ToString(reader[2])
            });
        }
    }
    conn.Close();
    return persona;
}
```

Figura 20. Código ejemplo búsqueda de datos mediante SP.

Elaborado por: Luis Núñez & Daniel Lincango

Método para actualización de datos

```
//METODO PARA ACTUALIZAR EL ESTADO DEL BIEN CONTROLADO
public static void actualizarEstadoBien(int idBien, int estado)
{
    SqlConnection conn = null;
    var cad = new clsConexion();
    conn = new SqlConnection(cad.conexion());
    conn.Open();
    SqlCommand cmd = new SqlCommand("acualizarEstadoBien", conn);
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Parameters.Add("@ID_BIEN_CONTROLADO", SqlDbType.Int).Value = idBien;
    cmd.Parameters.Add("@ESTADO", SqlDbType.Int).Value = estado;
    cmd.ExecuteNonQuery();
    conn.Close();
}
```

Figura 21. Código ejemplo actualización de datos mediante SP.

Elaborado por: Luis Núñez & Daniel Lincango

En la figura 19, 20 y 21 se establecen los pasos para la conexión con la base de datos mediante procedimientos almacenados, estos procedimientos reciben los parámetros y posteriormente realizan las sentencias INSERT, SELECT, UPDATE según sea lo requerido. La estructura del procedimiento almacenado (SP) se define según la (Figura 22).

Estructura para crear un SP

```
create procedure actualizarEstadoCondicionBienes
@ID_BIEN_CONTROLADO int,
@ID_CONDICION int,
@ID_CASO int,
@ESTADO int,
@OBSERVACION varchar(250)
as
UPDATE BIENES_CONTROLADOS SET ESTADO=@ESTADO, OBSERVACION=@OBSERVACION, ID_CONDICION=@ID_CONDICION,
ID_CASO=@ID_CASO
WHERE ID_BIEN_CONTROLADO=@ID_BIEN_CONTROLADO
```

Figura 22. Código ejemplo para creación de un SP.

Elaborado por: Luis Núñez & Daniel Lincango

3.4.2 Estructura JavaScript.

Para la aplicación web se creó una estructura de archivos de tipo JavaScript como lo muestra la (Figura 23) para el control de componentes que interactúan directamente del lado del cliente, obteniendo páginas completamente dinámicas y altamente amigables con el usuario final.

Contenido de la carpeta Scripts

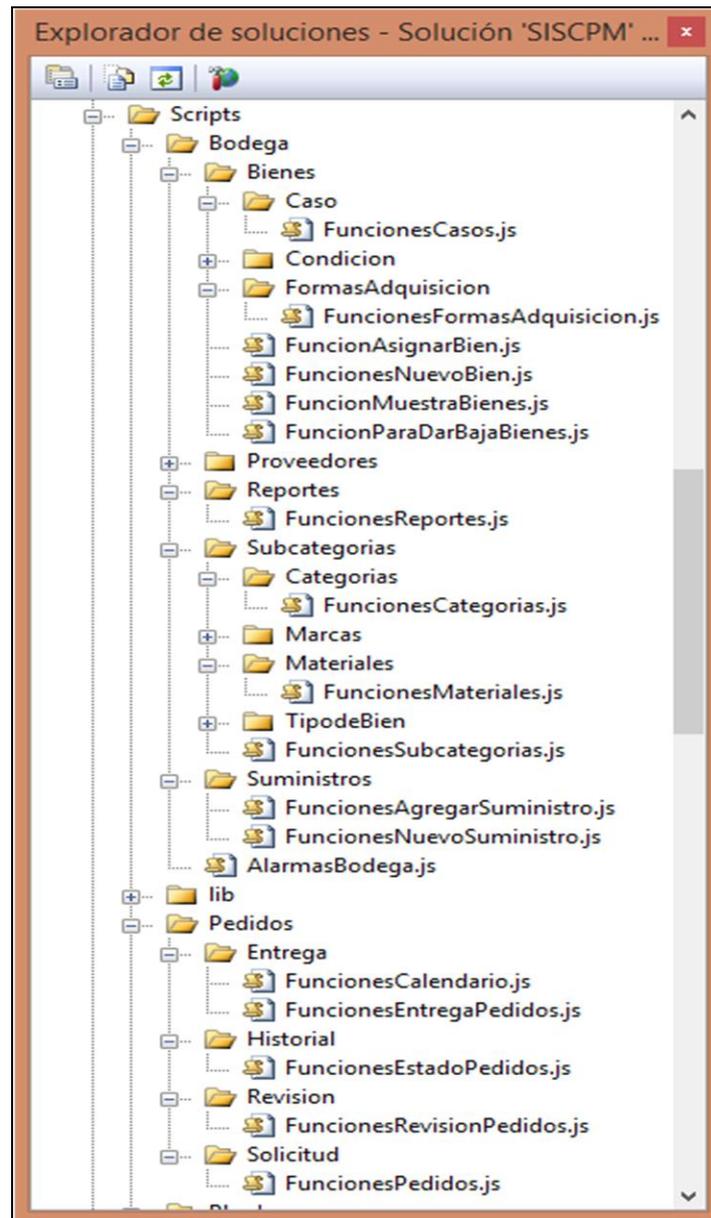


Figura 23. Estructura de archivos JavaScript de la aplicación.

Elaborado por: Luis Núñez & Daniel Lincango

Cada archivo dentro de esta estructura contiene métodos que crean la interfaz del usuario mediante los controladores de cada página, basándose en llamadas AJAX y obteniendo datos en formato JSON y mostrándolos en pantalla.

3.4.3 Llamadas AJAX.

La aplicación cuenta casi en su totalidad con funciones asíncronas mediante el uso de llamadas AJAX utilizando la librería JQuery, este tipo de llamadas permiten realizar acciones transaccionales o no transaccionales, permitiendo cargar datos desde el servidor sin interferir con la visualización ni el comportamiento de cada página, obteniendo respuestas del lado del servidor como segundo plano sin tener que recargar la página.

El uso de AJAX dentro de la aplicación permite que el usuario tenga completa interactividad con las funciones que quiere realizar, una de las funciones AJAX que se describe en la (Figura 24), realiza una llamada a un método web (Figura 25) creado en el controlador de la vista .aspx, el cual a su vez llama a un método creado en una clase específica (Figura 26) obteniendo un resultado asíncrono como muestra la (Figura 24).

Llamada AJAX

Método controlador 

```
$.ajax(  
  {  
    type: "POST",  
    url: "SubCategoria.aspx/validarSubcat",  
    contentType: "application/json; charset=utf-8",  
    dataType: "json",  
    cache: false,  
    async: false,  
    data: '{"nombre":""}',  
    success: dibujarListaSubCategorias,  
    error: function (result) {  
      mensajeriaSuministros(false,  
        '#FA5858','remove','tag','Error al obtener subcategorias!',  
        'Consulte con el administrador...');  
    }  
  }  
);
```

Método resultante 

Figura 24. Ejemplo método AJAX.

Elaborado por: Luis Núñez & Daniel Lincango

Método web Controlador

```
//METODO PARA VALIDAR SI YA EXISTE CATEGORIA
[WebMethod]
public static List<Object> validarSubcategoriaExiste(string nombre)
{
    List<Object> subCategoria = null;
    subCategoria = Base.clsSubcategoria.getSubCategoriaValidacion(nombre);
    return subCategoria;
}
```

Figura 25. Ejemplo método web Controlador.

Elaborado por: Luis Núñez & Daniel Lincango

Método de acceso a los datos

```
//METODO PARA OBTENER LAS CATEGORIAS PARA VALIDACION DE EXISTENCIAS
public static List<Object> getSubCategoriaValidacion(string nombre)
{
    List<Object> bien = new List<Object>();
    SqlConnection conn = null;
    var cad = new clsConexion();
    conn = new SqlConnection(cad.conexion());
    conn.Open();
    SqlCommand cmd = new SqlCommand("validarSubcategoria", conn);
    cmd.Parameters.Add("@nombre", SqlDbType.VarChar).Value = nombre;
    cmd.CommandType = CommandType.StoredProcedure;
    SqlDataReader reader = cmd.ExecuteReader();
    if (reader.HasRows)
    {
        while (reader.Read())
        {
            bien.Add(new validacion()
            {
                nombre = Convert.ToString(reader[0])
            });
        }
    }
    conn.Close();
    return bien;
}
```

Figura 26. Ejemplo método de acceso a datos.

Elaborado por: Luis Núñez & Daniel Lincango

3.4.4 Interfaz.

La interfaz de la aplicación está desarrollada en Bootstrap en su versión 3.3.6. Se optó por utilizar este framework por la extensa capacidad visual que nos ofrece a la hora de crear la interfaz visual de la aplicación mediante sus componentes como se muestra en la (Figura 27).

La aplicación cuenta con diferentes componentes listados a continuación:

- Alertas de validación e informativas.
- Barras de navegación.
- Calendarios dinámicos.
- Paginación, ordenamiento y búsqueda para listas.
- Ventanas modales.

Ventana modal de bootstrap.

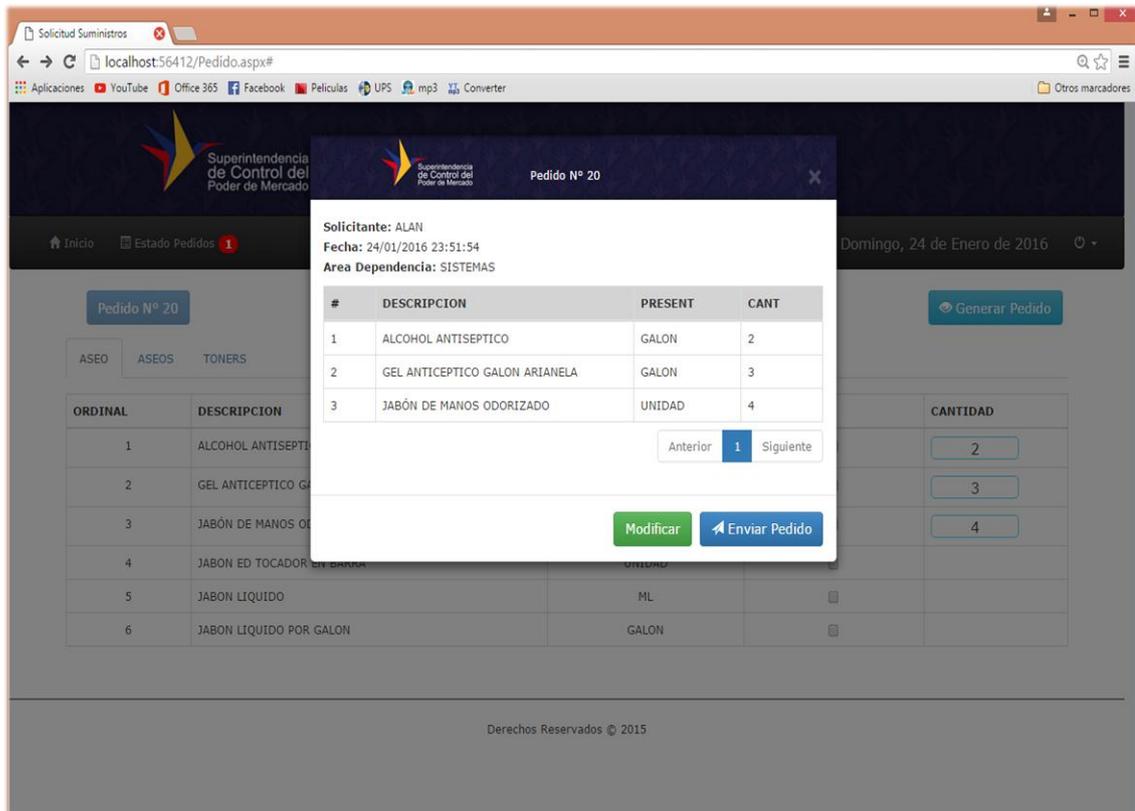


Figura 27. Interfaz ventana modal de Bootstrap.

Elaborado por: Luis Núñez & Daniel Lincango

3.4.5 Alertas y validación.

Dentro de la aplicación se creó un archivo JavaScript llamado MensajeríaSuministros.js, este archivo fue creado con el fin de mostrar alertas de peligro, precaución e informativas como componente genérico, es decir para ser utilizada por todo el sistema haciendo referencia a un método que se instancia de la siguiente manera:

```
mensajeríaSuministros(<recargar pagina true o false>,<color>,<icono principal>,<icono secundario>,<mensaje1>,<mensaje2>);
```

Mensaje de Alerta

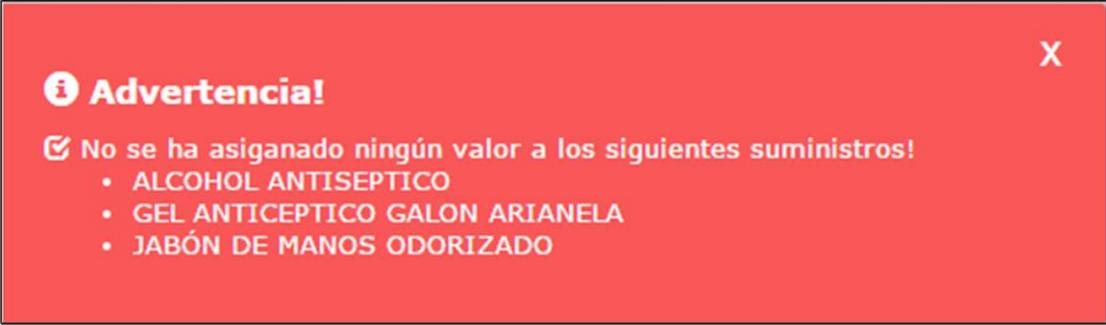


Figura 28. Alerta mensaje de validación.

Elaborado por: Luis Núñez & Daniel Lincango

The image shows a red alert box with a white 'X' in the top right corner. The text inside the box reads: 'Advertencia!' followed by 'No se ha asignado ningún valor a los siguientes suministros!' and a bulleted list: 'ALCOHOL ANTISEPTICO', 'GEL ANTICEPTICO GALON ARIANELA', and 'JABÓN DE MANOS ODORIZADO'.

Mensaje de alerta exitoso

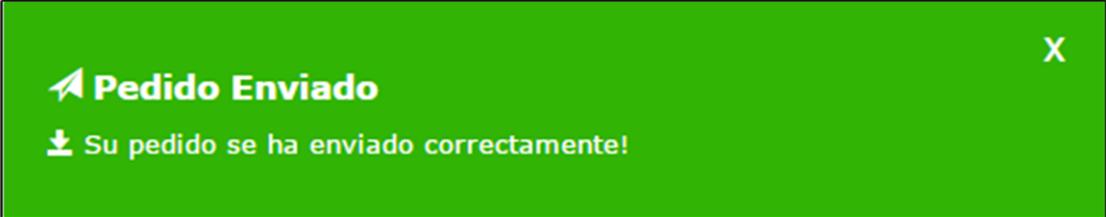


Figura 29. Alerta de acción exitosa.

Elaborado por: Luis Núñez & Daniel Lincango

The image shows a green alert box with a white 'X' in the top right corner. The text inside the box reads: 'Pedido Enviado' followed by 'Su pedido se ha enviado correctamente!'.

Para la validación se creó un archivo JavaScript llamado ValidacionesSuministros.js, este archivo contiene todos los métodos de validación de la aplicación desarrollada.

3.4.6 Librerías.

Para satisfacer la exigente demanda del usuario se optó por importar componentes de visualización de tipo JavaScript mediante algunas librerías libres descritas a continuación:

- **Bootstrap:** librería usada para obtener sus componentes visuales utilizados en toda la aplicación, ejemplo: botones, tabs, paneles, etc (Bootstrap, 2015).
- **DataTable:** librería usada para paginar, ordenar y buscar en una lista de datos específica (Javascript Source Data, 2015).

- **DatePicker**: librería usada para generar un calendario y obtener la fecha seleccionada (bootstrap-datepicker sandbox, 2015).
- **FullCalendar**: librería usada para generar agenda y visualización de acciones por fecha (Full Calendar 1.6 Released, 2015).
- **Jquery**: librería que sirve para facilitar acciones de código JavaScript usada en toda la aplicación, principalmente su función AJAX (jQuery, Downloading jQuery, 2015).
- **MaskMoney**: librería usada para controlar ingreso de monedas (jquery-maskMonkey, 2015).
- **DialogBox**: variante de ventana modal usada en comentarios de la aplicación (jQuery, Dialog, 2015).

Las formas de uso de las librerías así como link de descarga se encuentran en las referencias antes mencionadas.

3.5 Pruebas

Las pruebas de la aplicación fueron realizadas en cada una de los módulos que la conforman, que se detallan a continuación del documento.

3.5.1 Pruebas solicitud de suministros.

Dentro del módulo de solicitud de suministros se detalla el comportamiento de la aplicación en acciones de control y rendimiento.

Este módulo cuenta con validaciones como:

- Ingreso de solamente valores numéricos en cajas de texto de cantidad.
- Control de valores vacíos.
- Control de valores seleccionados.

Validaciones para pedidos

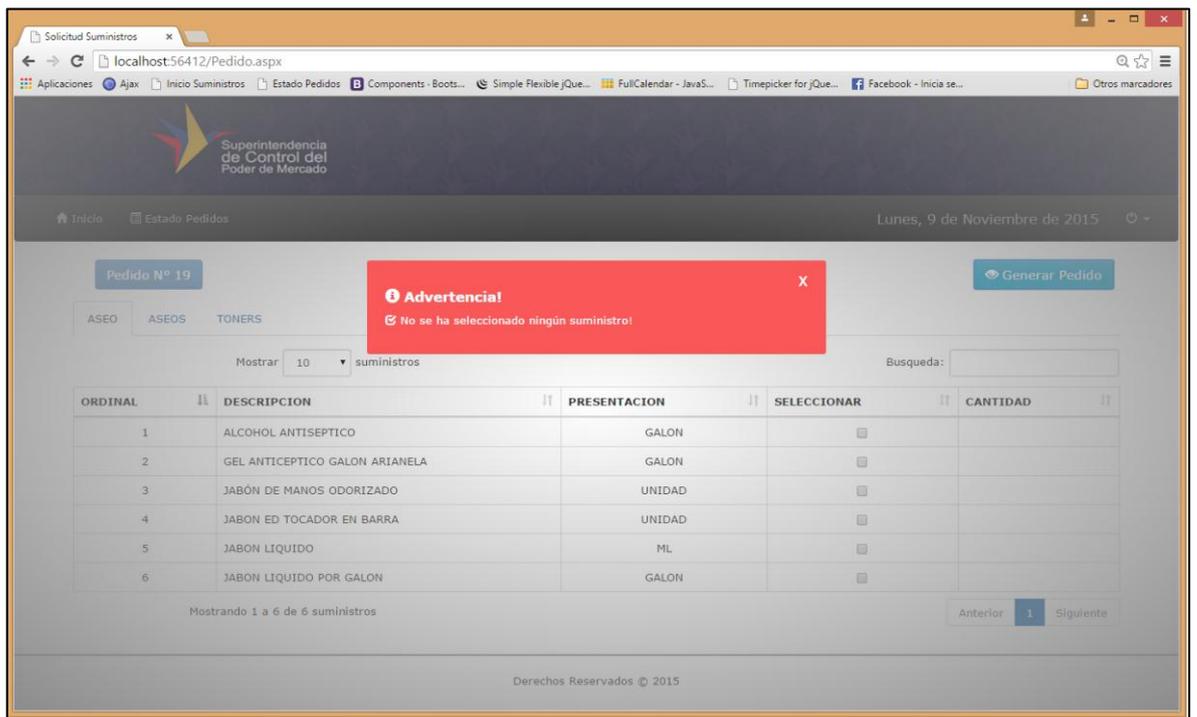


Figura 30. Validaciones módulo solicitud de suministros.

Elaborado por: Luis Núñez & Daniel Lincango.

Estas validaciones cuentan con una alerta personalizada, indicando la acción realizada. Ese tipo de control permite que los datos registrados por la aplicación no sean inconsistentes. Dentro del módulo de solicitud también contiene una ventana modal de confirmación como muestra la (Figura 31), la cual confirma el tipo de transacción se va a realizar.

Las pruebas realizadas sobre este módulo obtuvieron como resultado el registro exitoso de una nueva solicitud de suministro.

Confirmacion de Pedido

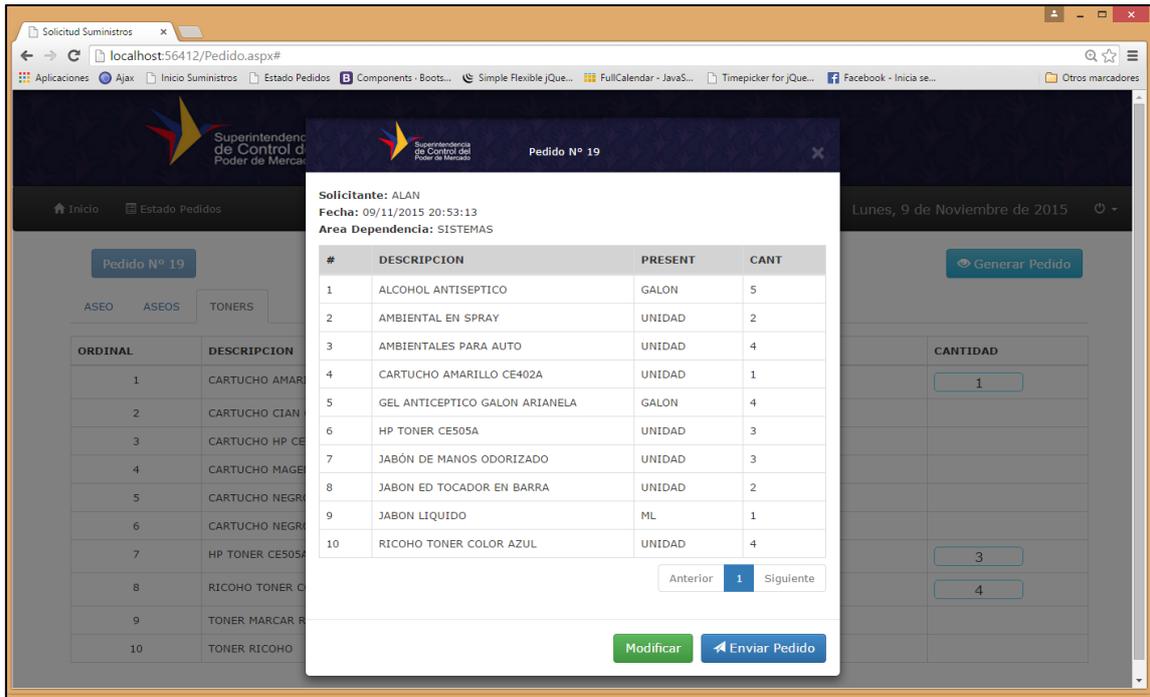


Figura 31. Ventana modal de confirmación módulo solicitud de suministros.

Elaborado por: Luis Núñez & Daniel Lincango.

3.5.2 Pruebas estado de pedidos.

El módulo de pedido cuenta con una barra de navegación indicando los pedidos realizados, aprobados, negados y pendientes de revisión. Este módulo cuenta con una ventana modal la cual muestra el detalle de un pedido seleccionado permitiendo visualizar el estado en que se encuentra cada suministro solicitado como muestra la (Figura 32), obteniendo como resultado la acción que se ha efectuado sobre el pedido enviado.

El resultado obtenido en este módulo es el de obtener información sobre el estado en que se encuentra el pedido realizado, mostrando detalladamente la transacción realizada por cada suministro solicitado.

Estado de los Pedidos

Revisado por: DANIEL LINCANGO
Fecha Emisión: 07/11/2015
Fecha Revisión: 07/11/2015
Fecha Entrega: 08/11/2015

SE ACEPTA CORRECTAMENTE

#	DESCRIPCION	PRESENTACION	CANTIDAD
11	CEPILLO PARA SANITARIO CON BASE PLASTICA	UNIDAD	6 → 2
12	CERA CREMA PARA AUTO	UNIDAD	✓ 7
13	CERA LIQUIDA PARA AUTO	UNIDAD	2 → 1
14	CERA PARA PISO ANTIDESLIZANTE	CENECA	5 → 1
15	CLORO LIQUIDO 5%	GALON	✓ 2
16	CLORO LÍQUIDO AL 5%	GALON	✓ 1
17	CONTENEDOR DE BASURA METALICO	UNIDAD	✓ 3
18	DELANTAL INDUSTRIAL	UNIDAD	21 → 1
19	DESINFECTANTE LIQUIDO VARIAS FRAGANCIAS	GALON	✓ 1
20	DESINFECTANTE PARA TANQUE INODOROS ZAPOLLO	UNIDAD	✓ 3

Anterior 1 2 Siguiente

Figura 32. Ventana Modal del estado de los pedidos.

Elaborado por: Luis Núñez & Daniel Lincango

3.5.3 Pruebas revisión de pedidos.

El módulo de revisión cuenta con una barra de navegación mostrando los pedidos pendientes de revisar, pedidos que ya han sido revisados y aprobados como también los pedidos que han sido revisados y negados por el usuario registrado. Este módulo cuenta también con una ventana modal la cual permite aprobar, negar o modificar el pedido recibido, la ventana cuenta con validaciones como stock mínimo, ingreso de valores numéricos y control de valores vacíos. Como notificación permite agregar un comentario a la transacción realizada por pedido seleccionado como muestra la (Figura 33).

El resultado obtenido de este módulo fue la revisión exitosa del pedido recibido, agregando una notificación con la información acerca de la transacción efectuada el mismo.

Revisión del Pedido Solicitado

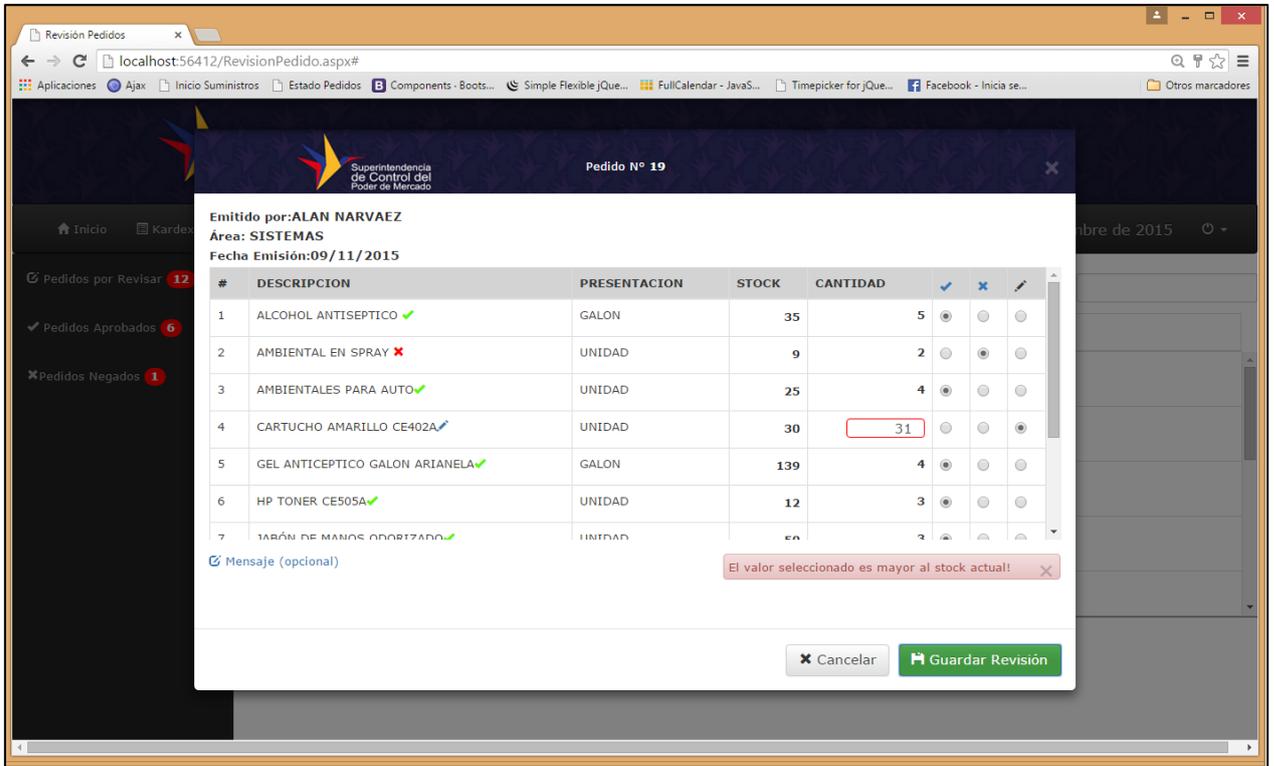


Figura 33. Ventana revisión de pedido solicitado.

Elaborado por: Luis Núñez & Daniel Lincango.

3.5.4 Pruebas administración de suministros.

El módulo de administración de suministros permite realizar ingresos de suministros nuevos y existentes, en los que los suministros nuevos permiten agregar un archivo adjunto, el módulo cuenta con validación de campos y mensajes de satisfacción como muestra la (Figura 34).

Este módulo contiene funcionalidades del módulo de administración de categorías, las cuales consisten en cargar información que depende de opciones seleccionadas, obteniendo como resultado ingresar nuevos suministros con características únicas.

El resultado de la prueba se obtuvo el ingreso exitoso de nuevos suministros.

Ingreso de nuevos suministros

REGISTRO DE NUEVO SUMINISTRO			
CÓDIGO NUEVO SUMINISTRO:	SCPM_SUM_27	FECHA INGRESO:	Lunes, 9 de Noviembre de 2015
NÚMERO DE DOCUMENTO:	968778969	PROVEEDOR:	PLASTILIMPIO <input type="button" value="Nuevo"/>
OBSERVACIONES:	AGREGAR OBSERVACIÓN...		
DESCRIPCION SUMINISTRO:	MAQUINA SECADORA DE MANOS		
TIPO DE BIEN:	UTILES DE LIMPIEZA <input type="button" value="Nuevo"/>	CATEGORIA:	ASEOS <input type="button" value="Nuevo"/>
SUB CATEGORIA:	SELECCIONAR SUB CATEGORÍ <input type="button" value="Nuevo"/>	ESTADO:	SELECCIONAR CONDICION
PRESENTACIÓN:	EJ: UNIDAD, LITROS, GALONES...	COSTO UNITARIO:	\$
EXISTENCIA MÍNIMA:	0	EXISTENCIA MÁXIMA:	0
CANTIDAD:	0	<input type="button" value="Ingresar"/>	

Figura 34. Ingreso de nuevos suministros.

Elaborado por: Luis Núñez & Daniel Lincango

3.5.5 Pruebas administración de proveedores.

El módulo de administración de proveedores se compone del ingreso de nuevo proveedor, edición de proveedores registrados, activar e inactivar proveedores. Este módulo contiene controles de validación de tipos de datos ingresados como muestra la (Figura 35).

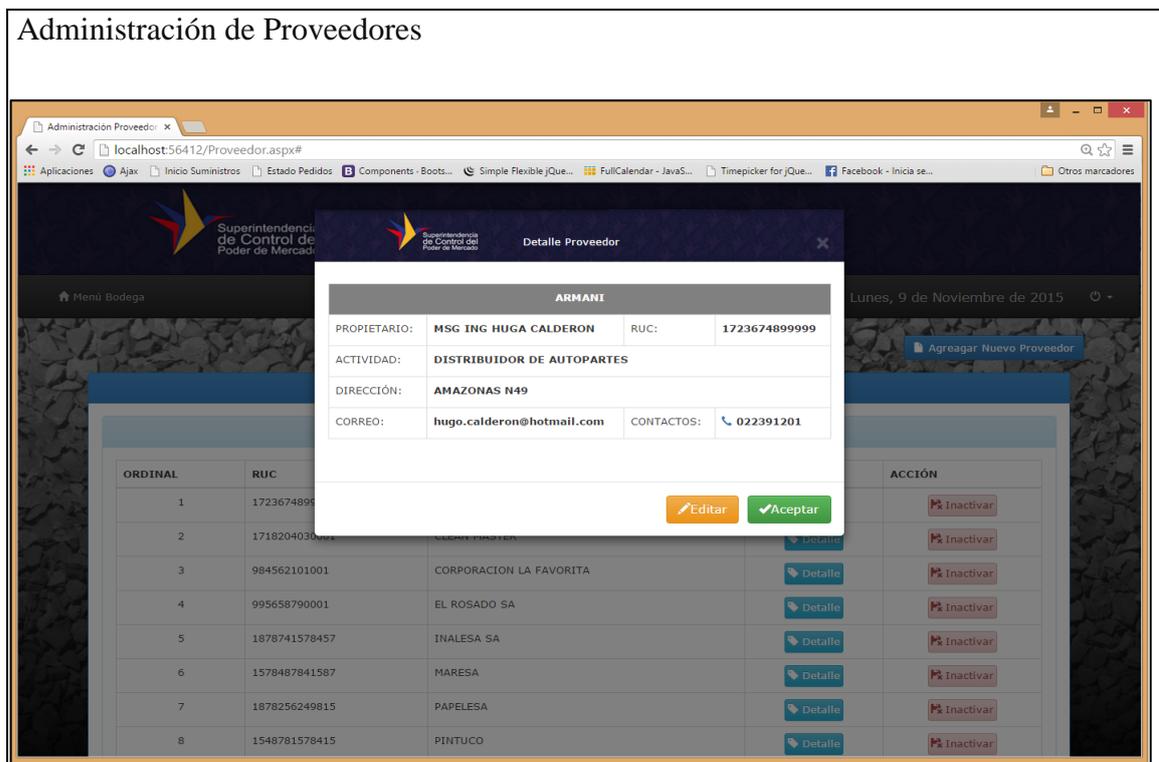


Figura 35. Administración de proveedores.

Elaborado por: Luis Núñez & Daniel Lincango

El resultado que se obtuvo fue el ingreso exitoso de nuevos proveedores, como también la edición de proveedores registrados, también se obtuvo que correcta activación e inactivación de proveedores registrados.

3.5.6 Pruebas entrega de pedido.

EL módulo entrega de pedido está constituido por una barra de navegación por estados de los pedidos que han sido aprobados y entregados. Este módulo cuenta con una ventana modal que permite la asignación de una fecha de entrega los pedidos recibidos, para después poder consultar en un calendario las acciones pendientes a realizarse por fecha como muestra la (Figura 36).

Fechas para entrega de Pedidos

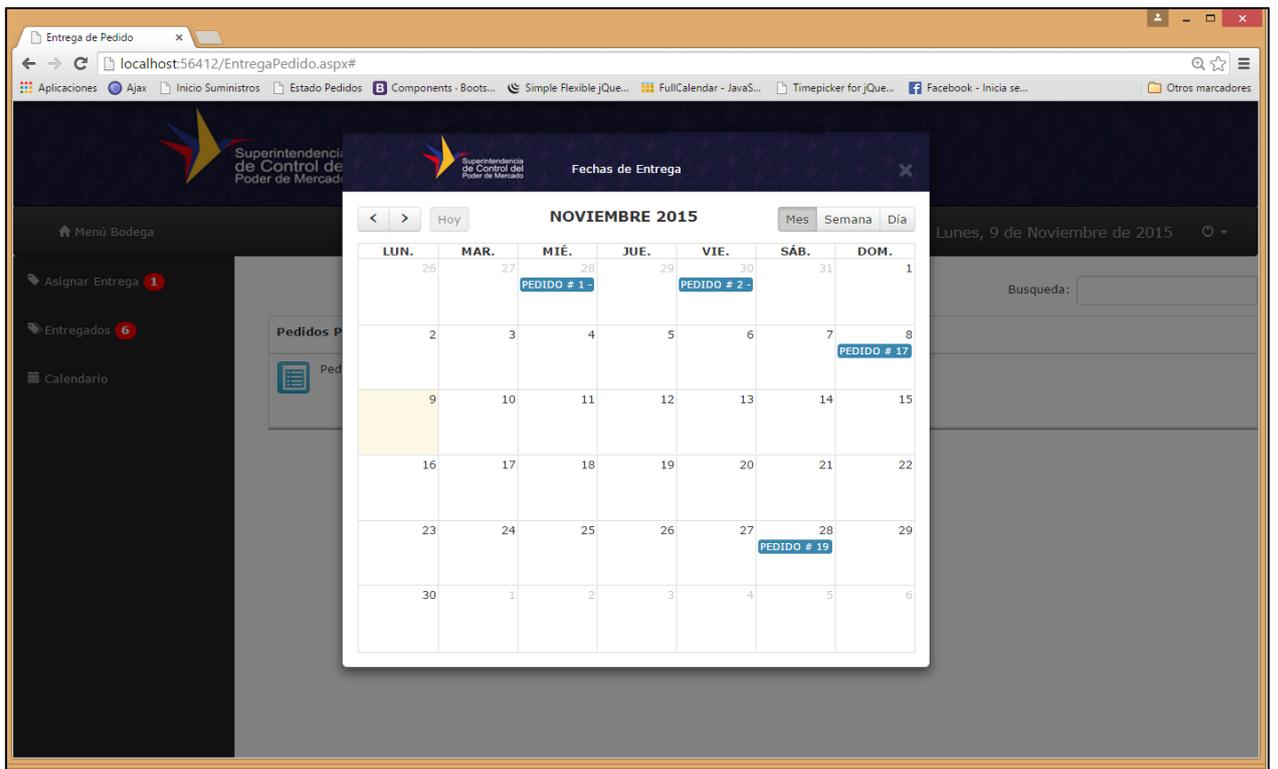


Figura 36. Calendario de entrega de pedidos.

Elaborado por: Luis Núñez & Daniel Lincango.

El resultado que se obtuvo de este módulo fue la asignación de una fecha de entrega por pedido aprobado, para luego mostrar en un calendario la fecha asignada.

3.5.7 Pruebas reportes suministros

El módulo de reportes se suministros está compuesto de un panel de búsqueda por suministro y un rango de fechas, en donde la fecha desde no puede ser mayor a la fecha hasta.

El resultado obtenido de la búsqueda permite visualizar las transacciones realizadas bajo los parámetros de búsqueda ingresados, estas transacciones se encuentran estructuradas en formato de Kardex en donde mostramos todos sus detalles como muestra la (Figura 37).

Kardex de Suministros

Generación Reportes Kardex

SUMINISTRO: AMBIENTAL EN TABLETA DESDE: 16/09/2015 HASTA: Consultar

ALCOHOL ANTISEPTICO

CODIGO	LIMP001	PRESENTACION	INGRESO			EGRESOS			EXISTENCIAS		
			CANT	C/U	TOTAL	CANT	C/U	TOTAL	CANT	C/U	TOTAL
05/10/2015	Nuevo ingreso por Factura 001	6	8.48 \$	50.88 \$	-	-	-	6	8.48 \$	50.88 \$	
10/10/2015	EGRESO POR PEDIDO # 2	-	-	-	1	8.48 \$	8.48 \$	5	8.48 \$	42.40 \$	
10/10/2015	EGRESO POR PEDIDO # 4	-	-	-	1	8.48 \$	8.48 \$	4	8.48 \$	33.92 \$	
15/10/2015	EGRESO POR PEDIDO # 15	-	-	-	1	8.48 \$	8.48 \$	3	8.48 \$	25.44 \$	
07/11/2015	INGRESO POR FACTURA 212121 CON PROVEEDOR ARMANI	12	9.68 \$	116.16 \$	-	-	-	15	9.68 \$	145.20 \$	
07/11/2015	INGRESO POR FACTURA 1313131 CON PROVEEDOR CORPORACION LA FAVORITA	1	12.52 \$	12.52 \$	-	-	-	16	12.52 \$	200.32 \$	
08/11/2015	INGRESO POR FACTURA 1324553 CON	5	26.30 \$	131.50 \$	-	-	-	21	26.30 \$	551.10 \$	

Figura 37. Reporte obtenido bajo parámetros de búsqueda.

Elaborado por: Luis Núñez & Daniel Lincango

3.5.8 Pruebas administración de categorías.

El módulo de administración de categorías está constituido por los siguientes sub módulos:

- Administración de Tipos de Bien.
- Administración de Sub Categorías.
- Administración de Materiales.
- Administración de Marcas.

Los sub módulos mencionados cuentan con funcionalidades de ingreso y edición de nuevos elementos. Cada módulo contiene validaciones para evitar la duplicidad de datos.

En la (Figura 38), se puede visualizar las funcionalidades que posee cada sub módulo generando una respuesta similar, es decir permite el ingreso de nuevos elementos y la edición de elementos registrados.

Mantenedor para Sub Categorías

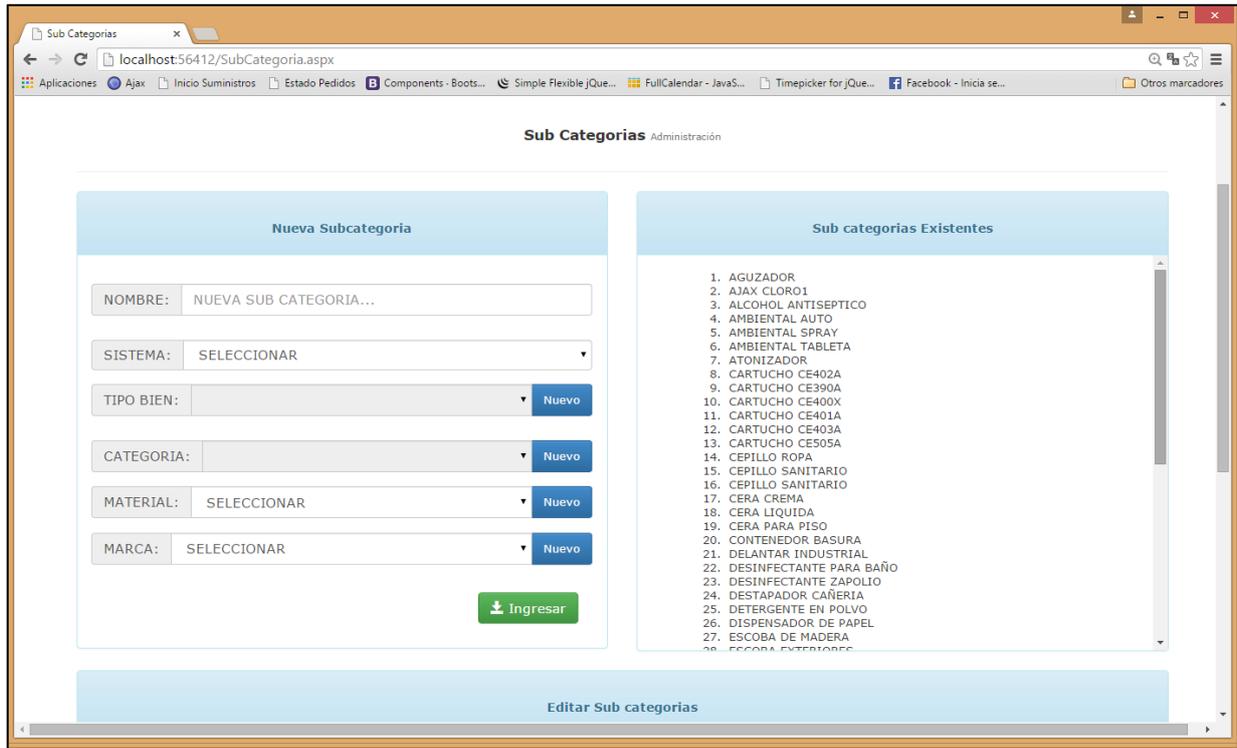


Figura 38. Administración de una sub categoría.

Elaborado por: Luis Núñez & Daniel Lincango

Al igual que en el módulo de administración de suministros existen valores que dependen de otros para ser mostrados.

3.5.9 Pruebas administración de bienes asignados.

El módulo de administración de bienes asignados cuenta con los siguientes sub módulos:

- Ingreso de nuevos bienes.
- Asignación de bienes.
- Baja del bien.
- Administración de casos.
- Administración de condiciones.

- Administración de formas de adquisición.

Los módulos de administración tienen la misma funcionalidad de los sub módulos de administración de categorías.

Este módulo permite crear nuevos bienes, así como también tiene la opción de asignarles una persona responsable. El ingreso del bien es controlado por tipos de datos requeridos el formulario de ingreso, para la asignación permite asignar solamente al personal existente dentro de la SCPM.

El resultado obtenido de este módulo fue el ingreso exitoso de nuevos bienes y la asignación de estos bienes a una persona responsable. Como también administrar casos, condiciones y formas de adquisición.

CONCLUSIONES

- En la actualidad la tecnología a la que se tiene acceso ha obligado a las empresas a ser más competitivas ofreciendo mayores facilidades al usuario mediante la implementación de sistemas automatizados y sistemas web. En la SCPM no es la excepción debido a que con las nuevas leyes en el sector público respecto al uso de papel y el desarrollo del software libre, se han planteado soluciones informáticas para reducir estos gastos. Por tal motivo, se puede concluir que este sistema aspira a brindar un mejor servicio al personal de la SCPM que hará uso del mismo, agilizando los diferentes procesos de despacho y adquisición de bienes y suministros, sin dejar de lado el cuidado del medio ambiente con la disminución de la cantidad de papel que se utiliza actualmente en este proceso. Esto ha hecho de este proyecto un objetivo dentro de la SCPM para ser implementado.
- Las metodologías ágiles, especialmente la metodología XP que es la usada con mayor frecuencia en proyectos pequeños y medianos, ha permitido tener unas guías claras sobre cómo se debe realizar los entregables, la recolección de requerimientos, el desarrollo del sistema, pruebas y otros aspectos relacionados directamente al control de calidad. Con estas pautas se ha podido desarrollar un sistema de gestión de suministros y bienes controlados que cumpla con los requerimientos solicitados por la SCPM, poniendo además atención a los estándares de calidad que rigen a esta metodología, logrando la adecuada utilización de las buenas prácticas durante todo el proceso de desarrollo.

- Como se pudo observar, la metodología que se utilizó fue de gran utilidad ya que actualmente no se contaba con un proceso bien definido sobre el cual el sistema se iba a desarrollar. Partiendo de este antecedente se procedió a crear un proceso nuevo que al contar con la aprobación de la SCPM se utilizó para elaborar el sistema en cuestión.
- La utilización de las diferentes herramientas usadas para crear el proceso base desde el cual el diseño de los diagramas seguido del desarrollo del sistema, han sido de ayuda en gran medida, ya que han permitido visualizar y corregir el proceso y los diagramas antes de pasar a la etapa del desarrollo. Estas herramientas han sido siempre parte de las buenas prácticas al momento de desarrollar un software, dando cumplimiento a los objetivos iniciales de este proyecto.
- La aprobación de este software ha pasado por diferentes cambios que áreas como la Dirección de TICS y la Dirección de Bienes y Suministros de la SCPM han solicitado. Por ser las áreas que harán uso y mantenimiento del sistema serán ellas las encargadas de realizar cambios, tomando en consideración nuevas actividades o funciones que a estos departamentos asigne la SCPM.

TRABAJOS FUTUROS

- Este sistema solventará la necesidad actual, según lo requerido y expresado por la SCPM en las diferentes reuniones realizadas para el efecto.

Pero al ser escalable, la institución u otro grupo de alumnos en un futuro pueden realizar modificaciones o mejoras al sistema, agregando nuevos módulos, según sea la necesidad de la institución al momento de enfrentar un nuevo obstáculo.

- La implementación de un sistema de auditoria puede hacer que el sistema actual interactúe con dicho sistema auditor para entregar informes más precisos y detallados sobre cómo se lleva el manejo de los bienes y suministros de la institución. Este escenario seria establecido, en caso de que la institución o las nuevas leyes sobre el sector público lo requieran.

REFERENCIAS

(12 de diciembre de 2015). Obtenido de bootstrap-datepicker sandbox:

<http://eternicode.github.io/bootstrap-datepicker/?markup=input&format=&weekStart=&startDate=&endDate=&startView=0&minViewMode=0&todayBtn=false&clearBtn=false&language=en&orientation=auto&multidate=&multidateSeparator=&keyboardNavigation=on&forceParse=on#s>

(23 de julio de 2015). Obtenido de Javascript Source Data:

http://datatables.net/examples/data_sources/js_array.html

Acerca de UML. (14 de septiembre de 2015). Obtenido de

<https://docs.kde.org/stable4/es/kdesdk/umbrello/uml-basics.html#about-uml>

AJAXYA. (12 de julio de 2015). *¿Qué es AJAX?* Obtenido de

<http://www.ajaxya.com.ar/temarios/descripcion.php?cod=8&punto=1>

Alicante, U. d. (19 de enero de 2016). *ASP.NET MVC 3 Framework*. Obtenido de

<http://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>

ARWEB. (3 de SEPTIEMBRE de 2015). *La caja de herramientas de ARWEB*.

Obtenido de <http://www.arweb.com/chucherias/editorial/%C2%BFque-es-bootstrap-y-como-funciona-en-el-diseno-web.htm>

Bernal, T. V. (20 de enero de 2016). *LA COSATECA*. Obtenido de

<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwiL6bTKrOTLAhXLHR4KHU09A1oQFggdMAA&url=http%3A%2F%2F1984.lsi.us.es%2Fpfe%2Ftrac%2Fpfe-cosateca-tveloso%2Fraw-attachment%2Fwiki%2FWikiStart%2FCap%25C3%25ADtulo%25205%2>

bizagi. (14 de enero de 2016). *bizagi*. Obtenido de

<http://www.bizagi.com/es/productos/bpm-suite/modeler>

Bolivariana, U. U. (27 de Julio de 2015). *Ingenieria de Software*. Obtenido de

http://ingenieriadesoftware.mex.tl/52682_Metodologias-Agiles.html

Bootstrap. (20 de julio de 2015). *Bootstrap getting started*. Obtenido de <http://getbootstrap.com/getting-started/>

Características metodología XP. (12 de agosto de 2015). Obtenido de <https://sites.google.com/site/xpmetodologia/marco-teorico/caracteristicas>

Elementos de UML. (14 de septiembre de 2015). Obtenido de <https://docs.kde.org/stable4/es/kdesdk/umbrello/uml-elements.html>

es.slideshare.net. (7 de Septiembre de 2015). Obtenido de http://es.slideshare.net/PIONEROSL/ejemplo-de-historia-de-usuario?next_slideshow=1

Fases de la Programación Extrema. (7 de agosto de 2015). Obtenido de <http://programacionextrema.tripod.com/fases.htm>

Full Calendar 1.6 Released. (13 de diciembre de 2015). Obtenido de <http://fullcalendar.io/blog/2013/03/fullcalendar-16-released/>

Informáticos, A. S. (12 de agosto de 2015). *aner*. Obtenido de ¿Qué es un ERP?: <http://www.aner.com/software-de-gestion-empresarial/que-es-un-erp.html>

Introducción a JSON. (22 de julio de 2015). Obtenido de <http://www.json.org/json-es.html>

jQuery. (12 de julio de 2015). Obtenido de Downloading jQuery: <https://jquery.com/download/>

jQuery. (18 de noviembre de 2015). *Dialog*. Obtenido de <https://jqueryui.com/dialog/>

jquery-maskMonkey. (27 de diciembre de 2015). Obtenido de <http://plentz.github.io/jquery-maskmoney/>

Jummp. (10 de Enero de 2012). *Jummp*. Obtenido de <https://jummp.wordpress.com/2012/01/10/desarrollo-de-software-tarjetas-crc/>

LIBROSWEB. (25 de agosto de 2015). *Introducción a JavaScript*. Obtenido de http://librosweb.es/libro/javascript/capitulo_1.html

mexired. (12 de octubre de 2015). *¿Qué es jquery?* Obtenido de <http://www.mexired.com/blog/que-es-jquery>

Network, M. D. (10 de octubre de 2015). *Introduccion a Visual Studio.Net*. Obtenido de [https://msdn.microsoft.com/es-es/library/aa291755\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa291755(v=vs.71).aspx)

Network, M. D. (10 de julio de 2015). *Microsof SQL Server*. Obtenido de <https://msdn.microsoft.com/es-es/library/bb545450.aspx>

Power Designer. (16 de diciembre de 2015). Obtenido de <http://powerdesigner.de/en/>

PUBLICA, S. N. (23 de julio de 2015). *Software Libre*. Obtenido de <http://www.administracionpublica.gob.ec/software-libre/>

Universidad Union Bolivariana. (28 de julio de 2015). *Ingenieria de Software*. Obtenido de http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html

ANEXOS

Anexo 1. Glosario de términos

Software: Es una palabra en inglés cuya masificación ha logrado que la real academia española la acepte. Según la RAE, el software se define como un conjunto de programas, instrucciones y reglas informáticas las cuales permiten ejecutar distintas tareas en una computadora.

RAE - Real academia española: institución fundada en 1713 con el propósito de elaborar un diccionario que contribuyera a mejorar y divulgar el conocimiento de la lengua española.

SCPM - Superintendencia de control de poder del mercado: entidad pública de derecho público encargada en asegurar la transparencia del mercado ecuatoriano, en donde su principal objetivo es brindar competencia económica y prevenir el abuso del control de poder del mercado.

Software Libre: se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

Heurísticas: generalmente son usadas cuando no existe una solución óptima bajo las restricciones dadas (tiempo, espacio, etc.)

Artefactos: es un producto tangible resultante del proceso de desarrollo de software. Algunos artefactos como los casos de uso, diagrama de clases u otros modelos UML ayudan a la descripción de la función, la arquitectura o el diseño del software.

SQL - Structured query language: es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

UPS: Universidad Politécnica Salesiana.

ERP - Enterprise resource planning: planificación de recursos empresariales es un paquete de software que permite administrar todos los procesos operativos de una empresa.

Metodologías ágiles: son una serie de técnicas para la gestión de proyectos.

XP – Xtremm programming: centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

RUP - Rational Unified Process: Es un proceso de ingeniería de software que suministra un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta y de mayor calidad para satisfacer las necesidades de los usuarios que tienen un cumplimiento al final dentro de un límite de tiempo y presupuesto previsible.

SCRUM: es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. En Scrum se realizan entregas parciales y regulares del producto final.

UML: es una técnica para la especificación sistemas en todas sus fases.

Refactorización: La refactorización consiste en tomar código ya existente y mejorarlo.

Tarjetas CRC: es un artefacto que permite exponer a un significado de una clase.

OMG - Object Management Group: es un consorcio, formado en 1989, dedicado al cuidado y el establecimiento de diversos estándares de tecnologías orientadas a objetos, como por ejemplo UML.

NULL: usado para indicar que el puntero no apunta a un objeto o dato válido.

Modelo ER – Entidad Relación: es un modelo de datos que permite representar cualquier abstracción, percepción y conocimiento en un sistema de información formado por un conjunto de objetos denominados entidades y relaciones, incorporando una representación visual conocida como diagrama entidad-relación.

Web: Conjunto de información que se encuentra en una dirección determinada de internet.

ASP – Active Server Pages: es una tecnología de Microsoft del tipo "lado del servidor" para páginas web generadas dinámicamente, que ha sido comercializada como un anexo a Internet Information Services.

XML - Xtensible Markup Language: es un lenguaje de marcas desarrollado por el World Wide Web Consortium utilizado para almacenar datos en forma legible.

IDE - Integrated Development Environment: es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

Framework: se define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

Web 2.0: comprende aquellos sitios web que facilitan el compartir información, la interoperabilidad, el diseño centrado en el usuario y la colaboración en la World Wide Web. Un sitio Web 2.0 permite a los usuarios interactuar y colaborar entre sí como creadores de contenido generado por usuarios en una comunidad virtual.

Script: es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.

CSS - Cascading style sheets: es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML;

PC – Personal Computer: ordenador personal.

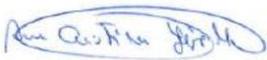
CSS3 - Cascading style sheets versión 3: es un lenguaje usado para definir y crear la presentación

Html - HyperText Markup Language: hace referencia al lenguaje de marcado para la elaboración de páginas web.

Metadatos: son datos que describen otros datos.

Anexo 2. Formulario para solicitud de suministros

FORMULARIO PEDIDO DE SUMINISTROS				
AREA REQUERENTE		COORDINACIÓN GENERAL DE TECNOLOGÍA		
FECHA		16/06/2015		
CÓDIGO	ITEM	CANTIDAD	SOLICITA	ENTREGA
SUMOF007	Aguzador eléctrico	Unidad		
SUMOF057	Almoadillas para sello	Unidad		
SUMOF032	Aprieta papeles tipo pinza 19mm	Caja x 12 pcs.		
SUMOF067	Aprieta papeles tipo pinza 25mm	Caja x 12 pcs.		
SUMOF068	Aprieta papeles tipo pinza 32mm	Caja x 12 pcs.		
SUMOF069	Aprieta papeles tipo pinza 41mm	Caja x 12 pcs.		
SUMOF070	Aprieta papeles tipo pinza 51mm	Caja x 12 pcs.		
SUMOF017	Archivador de cartón plegable	Unidad		
SUMOF060	Archivador tamaño oficio lomo 4	Unidad	5	
SUMOF027	Archivador tamaño oficio lomo 8cms.	Unidad	5	
SUMOF034	Bases de exfoliador	Unidad		
SUMOF014	Bolígrafo azul punta gruesa	Unidad		
SUMOF050	Bolígrafo negro punta fina	Unidad	10	
SUMOF051	Bolígrafo rojo punta fina	Unidad		
SUMOF047	Borrador de lápiz PZ20	Unidad	5	
SUMOF026	Borradores de esfero BR-40	Unidad		

SOLICITANTE		JEFE INMEDIATO	
			
NOMBRES:	ANA CRISTINA YÉPEZ	NOMBRES:	JUAN BAZANTE
CARGO:	DIRECTORA DE GESTIÓN TECNOLÓGICA (S)	CARGO:	COORDINADOR GENERAL DE TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN
GUARDALMACEN		RESPONSABLE DE BIENES	
NOMBRES:	Marco Quishpe	NOMBRES:	Paola Pazmiño