

**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



PROYECTO DE GRADO

“SOFTWARE PARA EL SERVICIO NACIONAL DE LABORATORIO”

CASO: CAJA NACIONAL DE SALUD

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA

MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS

POSTULANTE: RAPHAEL FLORES CHINO

TUTOR METODOLÓGICO: Ph. D. REYES PACHECO JAVIER HUGO

ASESOR: Ph. D. JOSE MARIA TAPIA BALTAZAR

**LA PAZ – BOLIVIA
2020**

Dedicatoria

A mis padres Eusebia y Oscar por su apoyo incondicional durante todo este tiempo, por su paciencia y amor en los momentos más difíciles, alentándome a continuar siempre sin importar las adversidades.

A mis hijos Jefferson, Anderson, Emerson, mi esposa Eva del Carmen y mi querida hermana Raquel por apoyarme en todo momento y por su aliento continuo dándome fuerzas y consejos. Todo se los debo a ustedes y a los sacrificios que hicieron; además de los valores y principios que me inculcaron.

Agradecimientos

A Dios que siempre me guio en todo momento permitiéndome llegar hasta aquí y culminar esta etapa tan importante de mi vida.

A mi tutor el Ph. D. Reyes Pacheco Javier Hugo, quien me brindo de su conocimiento y sabiduría para realizar este proyecto hasta su finalización, gracias por su paciencia para revisar una y otra vez este trabajo que es la culminación de varios años de estudio en esta linda carrera, gracias.

A mi asesor el Ph. D. José Maria Tapia Baltazar, por aceptarme y ser parte de este proyecto brindándome sus conocimientos y consejos para lograr este proyecto.

A todos mis amigos y compañeros que siempre estuvieron a mi lado ayudándome incondicionalmente, gracias por brindarme su amistad, los quiero mucho.

Gracias a todos

RESUMEN

El control de la producción de este sistema no constituye un fin en sí misma, sino un medio para alcanzar la eficacia y eficiencia de la institución, para establecer condiciones favorables que les permitan conseguir los objetivos deseados.

La Caja Nacional de Salud, institución descentralizada de derecho público sin fines de lucro, uno de los objetivos que pretende alcanzar un servicio y control eficiente, con una atención e información confiable y con reducción de tiempo en la documentación de forma confiable y oportuna, como ser: Enfermedad, Maternidad, Riesgos Profesionales y Asignaciones Familiares que comprenden los subsidios, natalidad, lactancia y sepelio.

El presente proyecto pretende coadyuvar a este fin, a través de la implementación de un sistema, el cual permite realizar el control eficiente, así ofreciendo información ordenada, completa, oportuna para la toma de decisiones.

De esta manera este Sistema aporta un control transparente, eficiente, eficaz y moderno. Para este propósito se utiliza los recursos tecnológicos y métodos apropiados para su desarrollo e implementación.

ABSTRACT

The control of the production of this system is not an end in itself, but a means to achieve the effectiveness and efficiency of the institution, to establish favorable conditions that allow them to achieve the desired objectives.

The Caja Nacional de Salud, decentralized institution of non-profit public law, one of the objectives that aims to achieve efficient service and control, with reliable attention and information and with reduction of documentation time in a reliable and timely manner, such as be: Sickness , Maternity Profes Professional Risks and Family Allowances that include the benefits , birth , breastfeeding and funeral.

This project aims to contribute to this end, through the implementation of a system, which allows for efficient control, thus offering orderly, complete, timely information for decision making.

In this way this System provides a transparent, efficient, effective and modern control. For this purpose, technological resources and appropriate methods are used for their development and implementation.

ÍNDICE

CAPITULO I MARCO INTRODUCTORIO	10
1.1. INTRODUCCIÓN	1
1.2. ANTECEDENTES	2
1.2.1. Análisis General de la Caja Nacional de Salud.....	2
1.2.1.1. Objetivo General.....	2
1.2.1.2. Misión.....	2
1.2.1.3. Visión	2
1.2.1.4. Antecedentes Históricos	2
1.2.1.5 Organigrama	3
1.2.2. Antecedentes del Proyecto	4
1.2.3. Trabajos o Proyectos Similares	4
1.2.4. Trabajos Internacionales	5
1.3. PROBLEMÁTICA	5
1.3.1. Problema Principal	5
1.3.2. Problemas Específicos.....	5
1.4. OBJETIVOS	6
1.4.1. Objetivo Principal	6
1.4.2. Objetivos Secundarios.....	6
1.5. JUSTIFICACIÓN	7
1.5.1. Técnica	7
1.5.2. Social.....	7
1.5.3. Económica.....	7
1.6. LIMITES Y ALCANCES.....	7

CAPITULO II MARCO TEÓRICO.....	8
2.1. SISTEMA DE INFORMACIÓN.....	8
2.2. QUE ES UN REEMBOLSO.....	8
2.3. INGENIERÍA DE SOFTWARE	8
2.3.1 Etapas de la Ingeniería de Software	9
2.3.1.1. Análisis de Requerimientos	9
2.3.1.2. Especificación.....	10
2.3.1.3. Diseño y Arquitectura.....	10
2.3.1.4. Programación.....	10
2.3.1.5. Prueba	10
2.4. METODOLOGÍA	11
2.4.1. Metodologías Agiles	11
2.4.2. ¿Qué es la Metodología Scrum?.....	12
2.5. INTRODUCCIÓN AL MODELO SCRUM PARA DESARROLLO DE SOFTWARE	12
2.5.1.- Preparación del Proyecto	12
2.5.1.1. Pila del Producto.....	13
2.5.1.2. Formato de la Pila del Producto	14
2.5.1.3. Pila del Sprint	15
2.5.1.4. Casos de uso UML	16
2.5.1.5. Elementos de los Diagramas.....	18
2.5.1.6. Asociaciones entre actores y casos de uso.....	20
2.5.1.7.- Generalización - Especificación entre actores	20
2.6. HERRAMIENTAS DEL DESARROLLO	21
2.6.1. Microsoft Visual Studio	21

2.6.1.1. Visual Studio 2013	21
2.6.2. SQL Server	22
2.7 PRUEBA	23
CAPITULO III MARCO APLICATIVO.....	24
3.1. Preparación del Proyecto	24
3.1.1. Requisitos y Visión del Producto	24
3.1.1.1.- Pila del Producto	24
3.2.- PLANIFICAR UN SPRINT	25
3.2.1.- Pila del SPRINT.....	25
3.2.2.- Herramientas de Trabajo.....	26
3.2.3.- Análisis de Riesgo.....	26
3.3.- Diseño Interfaces	26
3.3.1.- Diseño de Interfaz Gráfico del Sistema	26
CAPITULO IV CALIDAD Y SEGURIDAD	34
4.1. Calidad de Software.....	34
4.2. Funcionalidad.....	35
4.3. Confiabilidad	35
4.4.- Seguridad Informática	36
4.4.1.- ¿Por qué son necesarios los mecanismos de Seguridad?.....	36
4.4.2.- Principios de Seguridad Informática.....	37
4.4.3.- Seguridad en la Aplicación	38
4.4.4.- Seguridad Base de Datos	38
CAPITULO V CONCLUSIONES Y RECOMENDACIONES	39
5.1. Conclusiones	39
5.2. Recomendaciones	40

ÍNDICE DE FIGURAS

Figura 1.1. Organigrama de Caja Nacional de Salud (CNS).....	4
Figura 2.1. Desarrollo de la metodología SCRUM.	13
Figura 2.2. Planificación de Proyecto.....	14
Figura 2.2.1. Pila de Producto.....	16
Figura 2.2.3. Pila de Sprint.....	17
Figura 2.2.4. Elementos de diagrama.....	20
Figura 2.2.5. Especificación entre autores.....	22
Figura 3.1. Ingreso al Sistema.	27
Figura 3.1.1. Ingresando al Sistema.....	28
Figura 3.2. Ingresando al Sistema del Servicio de Laboratorio.	28
Figura 3.3. Modificación de Datos.	29
Figura 3.4. Modificación en los Datos.	29
Figura 3.5. Modificación en los Datos.....	30
Figura 3.6. Configuración de Usuario.....	30
Figura 3.7. Eliminación de Datos.	31
Figura 3.8. Eliminación de Datos.	31
Figura 3.9. Agregación de Datos.	32
Figura 3.10. Búsqueda de Insumos de Laboratorio.....	32
Figura 3.11. Inicio de Tramite.	33
Figura 3.12. Facturas.	33
Figura 3.13. Reporte.	34

Figura 4. Norma de Evaluación ISO/IEC 9126.....	35
Figura 4.1. Características de Confiabilidad.....	37

ÍNDICE DE TABLAS

Tabla 2. Casos de Uso.....	21
Tabla 3.1. Backlog del Producto (Pila del Producto).....	44
Tabla 3.2. Pila del Sprint.....	46

CAPITULO I MARCO INTRODUCTORIO

1.1. INTRODUCCIÓN

La necesidad de contar con tecnologías de información que en un principio permitieron al ser humano comunicarse y facilitar la ejecución de algunas actividades como se va reflejando hoy en día en todas las instituciones, empresas, fabricas, y otros, tienen la necesidad de manejar su información de forma segura, confiable, y que siempre este a su disposición en el momento que ellos más lo requieran, así de esta manera ellos pueden consultar la información, generar reportes, validar información, y de alguna manera tomar decisiones muy importantes.

La Oficina Nacional de la Caja Nacional de Salud se encuentra ubicada en la ciudad de La Paz, actualmente conlleva con mucha documentación que imposibilita la rapidez y fluidez de la misma, generando la susceptibilidad de pérdidas.

El proceso de reembolso es una etapa para la devolución de dinero por los servicios ofrecidos en otros hospitales o clínicas que se encuentran fuera de la institución y estos servicios son pagados por el asegurado(a) o beneficiario(a) con dinero propio.

Para una solicitud de reembolso tiene que cumplir con ciertos requisitos, como por ejemplo: la factura tiene que tener el NIT y el Nombre de la Institución, adjuntar el detalle de la factura donde detalla el procedimiento, como ser: Los Honorarios Médicos, el Servicio ofrecido, Insumos Médicos, Medicamentos y Laboratorios.

En el Servicio Nacional de Laboratorio uno de los problemas que existen actualmente son: Varios cálculos de facturas, los nombres o servicios de laboratorios que hacen en los Hospitales y Policlínicos, donde generan retrasos. Mencionar que existe personal encargado para este tipo de trabajo, este personal mayormente son de contrato temporal y su permanencia es dudosa por la existencia de cambios frecuentemente en la administración, esto genera en varias ocasiones una ruptura de gestión y la nueva capacitación y adaptación al nuevo personal genera un tiempo perdido, mientras existe este cambio genera la molestia en los asegurados y beneficiarios donde exigen su devolución inmediata de dinero.

La Caja Nacional de Salud actualmente existe aproximadamente más de 6 millones de Asegurados con sus respectivos beneficiarios y la tasa poblacional en crecimiento.

Por cada asegurado o beneficiario los laboratorios realizados son de dos o más exámenes por factura, eso nos da un punto de cuánto por día tiene verificar el personal encargado.

1.2. ANTECEDENTES

1.2.1. Análisis General de la Caja Nacional de Salud

La Caja Nacional de Salud, institución descentralizada de derecho público sin fines de lucro, encargada de la gestión, aplicación y ejecución del Régimen de Seguridad Social a corto plazo como ser: Enfermedad, Maternidad, Riesgos Profesionales y Asignaciones Familiares que comprenden los subsidios, natalidad, lactancia y sepelio.

1.2.1.1. Objetivo General

Realizar un servicio y control eficiente, con una atención e información confiable y con reducción de tiempo en la documentación de forma confiable y oportuna.

1.2.1.2. Misión

Proporcionar protección y atención integral de salud en los regímenes de enfermedad, maternidad y riesgos profesionales a la población asegurada y beneficiaria de la seguridad social de corto plazo, bajo los principios de solidaridad, unidad de gestión, economía, oportunidad y eficacia.

1.2.1.3. Visión

La Caja Nacional de Salud es referente nacional e internacional de la seguridad social a corto plazo en la promoción de la salud y en las prestaciones de servicios de salud integral con calidad, calidez, oportunidad, eficiencia y transparencia.

1.2.1.4. Antecedentes Históricos

El año 1987 esta institución cambia de nombre de Caja Nacional de Seguridad Social C.N.S.S. a Caja Nacional de Salud C.N.S., el Ministerio de Salud y Deportes junto al Instituto Nacional de Seguros de Salud INASES en Resolución Administrativa aprueba el Estatuto Orgánico de la C.N.S., que actualmente se rige bajo el Código de Seguridad Social y los dictámenes emanados por las autoridades en el transcurso de los años.

Se tiene el Seguro de Trabajador Dependiente, Rentistas, Seguro de Salud para el Adulto Mayor (SSPAM), Seguro Voluntario, Seguro del Abogado, Niños huérfanos, Niños especiales,

Instituto de ceguera, D.S. 20989, Seguro para excombatientes y viudas; la institución tiene la responsabilidad de atender la salud de sus asegurados y beneficiarios; esta actividad se realiza con el adecuado conocimiento de sus beneficiarios sobre sus derechos y obligaciones para poder acceder a los derechos de la Seguridad Social.

1.2.1.5 Organigrama

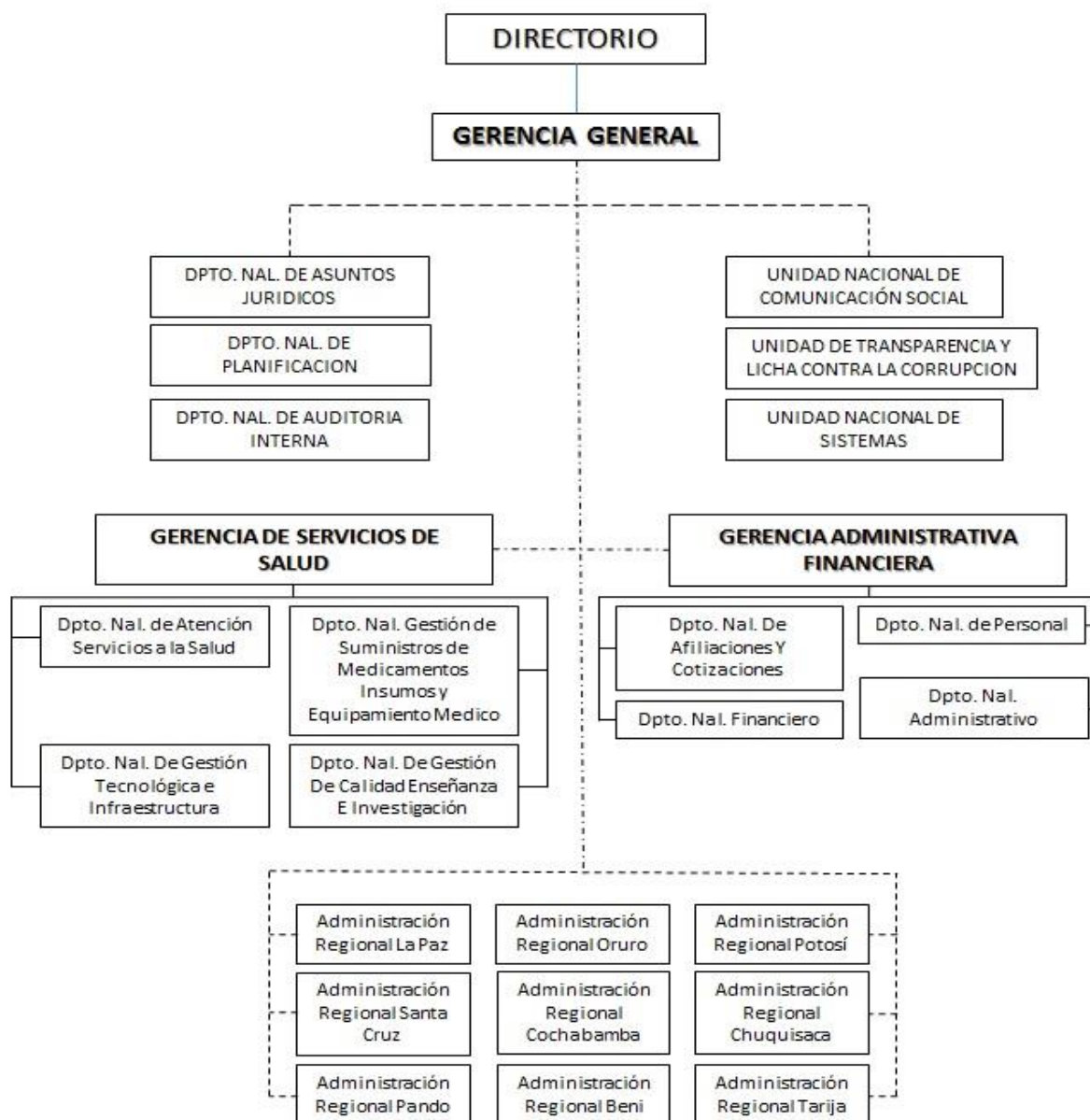


Figura # 2.0: Organigrama Caja Nacional de Salud

Fuente: [<https://www.cns.gob.bo/Site/organigrama>, 2009]

1.2.2. Antecedentes del Proyecto

Para el desarrollo del presente proyecto, se ha visto la necesidad de revisar algunos trabajos desarrollados para el uso y control que tienen relación al proyecto propuesto.

1.2.3. Trabajos o Proyectos Similares

En la carrera de informática de la Universidad Mayor de San Andrés (UMSA) se encontraron proyectos de grado relacionados al control de producción como ser:

- Sistema de Información para la Producción Planificada BROSSO, tiene el objetivo de implantar un Sistema de Información que permita mejorar la planificación de la producción en sus puntos más críticos: ajuste de presupuesto, gestión de almacén con el método ponderado y el seguimiento de producción, aplicando la metodología del el Lenguaje de Modelado Unificado (UML), donde se captura los requisitos de los sistemas a través de paquetes transaccionales llamados casos de uso. [BROSSO, 2015]
- Sistema de Almacenes y Producción SOCOVIAL, tiene el objetivo de diseñar e implementar un sistema de información basado en el control de inventarios y mejorar el control productivo, acorde a las necesidades y el diseño orientado a objetos de Martín & Odell.
- Sistema de Información Integrado PIL-CHUQUISACA S.A. (SII-PIL Chuquisaca), tiene como objetivo controlar y optimizar el proceso de la información en la planta Industrializadora. Además de un modelo de implantación del usuario y del sistema, aplicando la metodología de Análisis y Diseño Estructurado, la cual parte el sistema funcionalmente y según distintos comportamientos se establece la esencia de lo que se debe construir, donde las herramientas de modelado son el Modelo Esencial que a la vez se divide en el Modelo Ambiental y el Modelo de Comportamiento. [PIL ANDINA, 2015]
- El Instituto de Servicios de Laboratorio de Diagnóstico e Investigación en Salud (SELADIS) se origina en la Facultad de Ciencias Farmacéuticas y Bioquímicas de la UMSA como parte de una propuesta académica alternativa, tomándose como objetivo Principal los Procesos de SERVICIOS, INVESTIGACIÓN y ENSEÑANZA. En principio el Instituto SELADIS, se encontraba instalado en dependencias de la F.C.F.B y microbiología., donde se realizaban pruebas básicas en Análisis Clínicos, Hematología, Inmunología, Parasitología y Serología. Debido a la creciente demanda de la población en relación a pruebas de laboratorio fidedignas y actualizadas se fueron agregando otras Unidades Laboratoriales, tales como

Inmunología, Microbiología, Biología Molecular y otros. Finalmente el instituto SELADIS, crea sus dependencias propias en la Av. Saavedra # 2224, donde actualmente se encuentra funcionando. [SELADIS, página oficial].

1.2.4. Trabajos Internacionales

- Tesis, ClickBalance, en términos contables refleja la existencia física de la mercancía, materia prima, productos semi-terminados o terminados que tiene una empresa en un lugar y fecha determinada. Para llevar el control existen los sistemas de inventarios, que pueden ser que se realizará el registro contable del inventario inicial, las compras, devoluciones o rebajas sobre compras, las ventas y el inventario final, dando como resultado en forma global la utilidad o pérdida por las ventas de las mercancías correspondientes.

Además, para este tipo de control de inventarios se deberá realizar un inventario físico del mismo para determinar cuánto queda en el almacén, así como la utilidad o pérdida por las ventas realizadas. [CLICKBALACE, página oficial].

1.3. PROBLEMÁTICA

1.3.1. Problema Principal

Uno de los problemas principales son los reembolsos en demora que dura hasta tres semanas dependiendo de la dificultad, el detalle y la espera en la que tiene que ser atendido la documentación, para luego ser entregados a otra unidad para su revisión correspondiente.

Por lo expuesto anteriormente se plantea el siguiente problema:

¿Cómo mejorar el manejo de la información, rendimiento y optimización de producción que posibilite la reducción de tiempo, la obtención de información confiable y oportuna en los trámites de reembolso?

1.3.2. Problemas Específicos

- El proceso de acelerar el trámite correspondiente y del rendimiento que genera un tiempo moroso para el servicio de Laboratorio.
- El sistema de producción actual es realizado en Word, haciéndolo muy inseguro y con bastantes limitaciones y un retraso inminente.

- Actualmente la plataforma del programa Word, que utiliza el Servicio de Laboratorio, tiene la facilidad en perderse la información.
- El proceso de agregar, sumar y eliminar datos en el sistema actual, es muy moroso, donde usualmente los datos se repiten, generando el llenado de datos más de dos veces, y existiendo redundancia de datos.
- Los datos no son íntegros, debido a que se encuentran en Word, muchas de los datos contienen información redundante.
- La solicitud del reembolso no solamente es del Servicio Nacional de Laboratorio sino también en otros servicios y/o departamentos, como por ejemplo el departamento de Gestión Suministros que se encarga de la verificación de insumos y servicios prestados fuera de la institución y Farmacia que es la encargada de verificar los distintos medicamentos que se le generan al paciente al momento de ser atendido fuera de la institución, eso genera mayormente el problema y donde puede existir la susceptibilidad de errores en la Suma total de la(s) Factura(s) y generando así un tiempo perdido el proceso de trámite.

1.4. OBJETIVOS

1.4.1. Objetivo Principal

Desarrollar un Sistema Administrativo para realizar el control eficiente de los reembolsos del Servicio Nacional de Laboratorio, con una reducción de tiempo y una información confiable y oportuna.

1.4.2. Objetivos Secundarios

Refinar instrumentos metodológicos, instrumentos de trabajo para el desarrollo, diseño y centro de calidad del proyecto.

Estructurar, diseñar y desarrollar el sistema de aplicación con fluidez y eficiencia.

Verificación de calidad del proyecto con la seguridad de la aplicación y evitar alteraciones.

Implementar tecnologías de arquitectura de software para el desarrollo del proyecto que sea intuitivo, entendible y fácil de usar.

Creación de una mayor seguridad de la información y sea entendible tanto general como específico.

1.5. JUSTIFICACIÓN

1.5.1. Técnica

El proyecto es técnicamente justificable para el Servicio Nacional de Laboratorio debido a las facilidades en cuanto a tecnología, hardware y software, que serán utilizados para la implementación del presente proyecto. Es importante mencionar que el desarrollo del proyecto es realizado en su totalidad, con herramientas sofisticadas, que permitirá el buen uso en este proyecto.

1.5.2. Social

El presente proyecto se convierte en un instrumento que facilite la toma de decisiones, permitiendo así al Servicio Nacional de Laboratorio contar con un mejor proceso a la hora de tomar decisiones. Adicionalmente podrá contar óptimos rendimientos, que beneficiara mayormente a la Institución.

1.5.3. Económica

El presente proyecto es económicamente justificable debido a que es generado para ser devueltos su(s) reembolso(s) de manera rápida, eficaz y que permitirá agilizar su desembolso y permitirá a la Institución no perder tiempo y dinero, la justificación de este software tiene una base muy sólida en lo que se pretende.

1.6. LIMITES Y ALCANCES

El presente proyecto, aparte de ser escalable, para una primera versión, permitirá obtener datos, óptimos rendimientos y una rapidez en el trámite que se está desarrollando los cuales no son contemplados actualmente.

Tendrá una amplitud de datos que le permitirá facilitar y tener fluidez en el momento de cualquier verificación de la factura y permitirá a este Servicio de Laboratorio a tomar decisiones.

CAPITULO II MARCO TEÓRICO

Para el desarrollo del proyecto se usa los conceptos descritos en el presente capítulo, relacionados con la metodología a usar durante el desarrollo del sistema, así como las distintas herramientas y métodos que servirán para realizar el presente proyecto.

2.1. SISTEMA DE INFORMACIÓN

Un sistema de información (SI) es un conjunto de elementos interrelacionados con el propósito de prestar atención a las demandas de información de una organización, para elevar el nivel de conocimientos que permitan un mejor apoyo a la toma de decisiones y desarrollo de acciones.

La construcción de un sistema de información implica la conjugación de esfuerzos, conocimientos, experiencias, recursos y tiempo muy valiosos; por lo que es necesario contar con un adecuado rumbo de acción que garantice el éxito del proyecto, empleado al máximo los elementos disponibles. Por esta razón es conveniente apoyarse en una metodología que establezca las etapas con objetivos, actividades y técnicas necesarias en la creación de un sistema. [Peña, 2006]

2.2. QUE ES UN REEMBOLSO

Para poder hablar de un reembolso, en primer lugar deberíamos definir su concepto: un reembolso es la devolución de dinero a la persona que la había desembolsado con sus recursos propios. Se reconoce como reembolso a la operación económica mediante la cual una persona o entidad recibe de vuelta alguna cantidad de dinero o de bienes materiales que había entregado como pago de un servicio o producto (<https://www.definicionabc.com/economia>).

2.3. INGENIERÍA DE SOFTWARE

La ingeniería del software es el proceso formal de desarrollo de software en el que las necesidades del usuario se traducen en requerimientos, estos se transforman en diseño que se implementa en código que se prueba, documenta y se certifica para su uso operativo. La ingeniería del software se define como “(1) la aplicación de un método sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, esto es, la aplicación de la ingeniería al software” y “(2) el estudio de los métodos de (1)” [Xavi, 2013].

Un aspecto muy importante de Ingeniería de Software es que proporciona parámetros formales para lo que se conoce como Gestión (o Administración) de Proyectos de Software. Esto se

refiere a que Ingeniería de Software proporciona diversas métricas y metodologías que pueden usarse como especificaciones para todo lo referente a la administración del personal involucrado en proyectos de software, ciclos de vida de un proyecto de software, costos de un proyecto, y en si todo el aspecto administrativo que implica el desarrollar software. Por supuesto que estos aspectos no son relevantes para los fines de este proyecto, principalmente porque este proyecto no se desarrolla con fines lucrativos monetariamente hablando. [Peña, 2006]

2.3.1 Etapas de la Ingeniería de Software

El proceso requiere una metodología con las siguientes etapas:

2.3.1.1. Análisis de Requerimientos

Se extraen los requisitos del producto de software. En esta etapa la habilidad y experiencia en la ingeniería del software es crítica para reconocer requisitos incompletos, ambiguos o contradictorios. Usualmente el cliente/usuario tiene una visión incompleta/inexacta de lo que necesita y es necesario ayudarlo para obtener la visión completa de los requerimientos. El contenido de comunicación en esta etapa es muy intenso ya que el objetivo es eliminar la ambigüedad en la medida de lo posible. [Xavi, 2013].

Análisis de los requisitos del software. El proceso de reunión de requisitos se intensifica y se centra especialmente en el software. Dentro del proceso de análisis es fundamental que a través de una colección de requerimientos funcionales y no funcionales, el desarrollador o desarrolladores del software comprendan completamente la naturaleza de los programas que deben construirse para desarrollar la aplicación, la función requerida, comportamiento, rendimiento e interconexión.

La determinación de requerimientos se realiza mediante las tareas siguientes:

Definición del caso de estudio. Se identifica el tema central que motiva el inicio del estudio, pudiendo ser la creación de un nuevo sistema o la modificación a uno ya existente.

Estudio de la organización. Se determina con precisión las áreas

Estudio de la organización. Se determina con precisión las áreas usuarias participantes, su estructura orgánica, funciones, interrelaciones y compromisos con otras.

Análisis de procedimientos. Se estudian todos los procedimientos relacionados con el problema planteado, identificando para cada uno de ellos: los objetivos que persiguen, las actividades que

realizan, secuencia y periodicidad, responsables, niveles de agregación, sus relaciones con otros puntos de control y situaciones especiales que imperan.

Análisis de información. Se identificarán los flujos de información, documentos y reportes, operaciones (de registro, validación, almacenamiento, clasificación, cálculo y presentación), volúmenes y períodos; que se desprenden de la ejecución de los procedimientos estudiados.

Identificación de recursos. Se hace un reconocimiento de los recursos humanos y materiales participantes en el desarrollo de las actividades. [Peña, 2006]

2.3.1.2. Especificación

Es la tarea de describir detalladamente el software a ser escrito, de una forma rigurosa. Se describe el comportamiento esperado del software y su interacción con los usuarios y/o otros sistemas. [Xavi, 2013]

2.3.1.3. Diseño y Arquitectura

Determinar cómo funcionará de forma general sin entrar en detalles incorporando consideraciones de la implementación tecnológica, como el hardware, la red, etc. Consiste en el diseño de los componentes del sistema que dan respuesta a las funcionalidades descritas en la segunda etapa también conocidas como las entidades de negocio. Generalmente se realiza en base a diagramas que permitan describir las interacciones entre las entidades y su secuenciado. [Xavi, 2013]

2.3.1.4. Programación

Se traduce el diseño a código. Es la parte más obvia del trabajo de ingeniería de software y la primera en que se obtienen resultados “tangibles”. No necesariamente es la etapa más larga ni la más compleja aunque una especificación o diseño incompletos/ambiguos pueden exigir que, tareas propias de las etapas anteriores se tengan que realizarse en esta. [Xavi, 2013]

2.3.1.5. Prueba

Consiste en comprobar que el software responda/realice correctamente las tareas indicadas en la especificación. Es una buena praxis realizar pruebas a distintos niveles (por ejemplo primero a nivel unitario y después de forma integrada de cada componente) y por equipos diferenciados del de desarrollo (pruebas cruzadas entre los programadores o realizadas por un área de test independiente). [Xavi, 2013]

2.4. METODOLOGÍA

Una metodología trata de organizar en la medida de lo posible los procedimientos a realizar durante la ejecución de los procesos que conlleva un proyecto. Los procesos de las metodologías tradicionales implican una planificación muy detallada resultando muy costoso el trabajo de planificación, diseño y documentación. Esto no era lo que se necesitaba para proyectos pequeños o medianos en los que el esfuerzo invertido en estas tareas superaba con creces la programación del sistema software.

Esto ha provocado que en sus inicios se utilizaran metodologías existentes en otras áreas para la gestión de proyectos informáticos. [Eraso, 2013]

2.4.1. Metodologías Ágiles

Principios de las metodologías ágiles.

- Individuos e interacciones sobre procesos y herramientas.
- Software funcionando sobre documentación extensiva.
- Colaboración con el cliente sobre negociación contractual.
- Respuesta ante el cambio sobre seguir un plan

En las metodologías ágiles podemos encontrar una serie de prácticas o técnicas habituales a la hora de afrontar la ejecución de un proyecto. Por ejemplo, a la hora de hacer reuniones tienen que ser rápidas y frecuentes, lo suficientemente rápidas (ágiles) como para no perder el tiempo pero con una frecuencia suficiente para que los integrantes del equipo estén informados de todo. Suelen ser reuniones diarias. Cuando se trata de una reunión de planificación de iteración en la que hay una parte del producto para entregar se re-planifica la siguiente iteración a partir del feedback obtenido del cliente. El intervalo de tiempo entre entregas se denomina iteración. Como se puede deducir en este tipo de reuniones participa el cliente, al que se considera uno más del equipo. [Eraso, 2013]

Ejemplos de metodologías ágiles:

- Extreme Programming (XP) se rige sobre la suposición de que es posible obtener software de gran calidad a pesar, o incluso como consecuencia del cambio continuo.

- Su principal asunción es que con un poco de planificación, un poco de codificación y unas pocas pruebas, se puede decidir si se está siguiendo un camino acertado o equivocado, evitando tener que echar marcha atrás demasiado tarde
- Scrum parte de la esencia del desarrollo ágil. Se centra en las funcionalidades con más prioridad y que pueden ser ejecutadas en un periodo corto de tiempo. Los ciclos de desarrollo, llamados sprints en Scrum, producen un incremento de funcionalidad terminado y operativo. [Eraso , 2013]

Para el desarrollo de este proyecto se utilizara el Método ágil Scrum

2.4.2. ¿Qué es la Metodología Scrum?

Scrum es una metodología ágil y flexible para gestionar el desarrollo de software, cuyo principal objetivo es maximizar el retorno de la inversión para su empresa (ROI). Se basa en construir primero la funcionalidad de mayor valor para el cliente y en los principios de inspección continua, adaptación, auto-gestión e innovación. [SOFTENG, 2012].

(Ver figura 2.1)

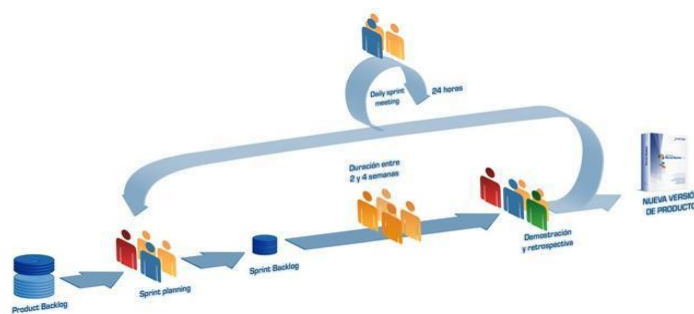


Figura # 2.1: Desarrollo de la metodología Scrum

Fuente: [SOFTENG, 2012]

2.5. INTRODUCCIÓN AL MODELO SCRUM PARA DESARROLLO DE SOFTWARE

2.5.1.- Preparación del Proyecto

Conocida como sprint 0, es la fase inicial en la que se intenta comprender el caso de negocio con la finalidad de tomar decisiones que aseguren valor al producto.

Durante esta fase se producen gran número de inexactitudes con las estimaciones, pero es lógico, debido a que se hacen a alto nivel, por lo tanto es aconsejable no perder tiempo en buscar las

estimaciones exactas, es mejor invertir ese tiempo en el desarrollo. [Trigas, 2012]
(Ver figura 2.2)

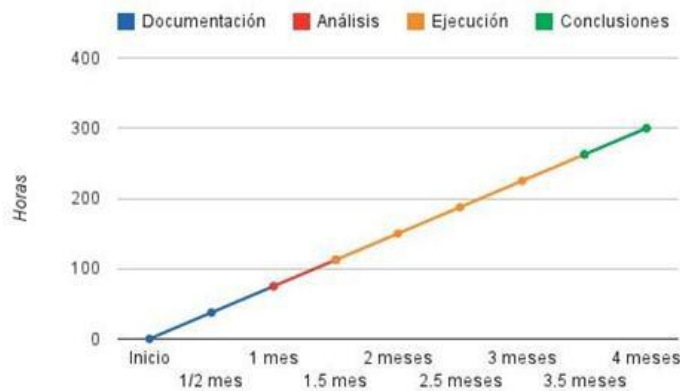


Figura # 2.2 Planificación del proyecto

Fuente [Eraso, 2013]

En el gráfico anterior se representa la línea de tiempo planificada. Al comienzo del mismo está la fase de documentación, compuesta por la exposición del problema y documentación sobre el tema del proyecto que formará el primer mes. La segunda parte, Análisis, nos ocupa la primera mitad del segundo mes, que continúa con el desarrollo de la aplicación durante 2 meses. Y para finalizar dejamos 2 semanas como mínimo para la redacción de la memoria final, sacar conclusiones y finalizar el proyecto. [Palacio, 2011]

2.5.1.1. Pila del Producto

La pila del producto es el inventario de funcionalidades, mejoras, tecnología y corrección de errores que deben incorporarse a través de las sucesivas iteraciones de desarrollo. Todo lo que suponga un trabajo que debe realizar el equipo tiene que estar reflejado en esta pila.

Estos son algunos ejemplos de posibles entradas de un backlog:

- Permitir a los usuarios la consulta de las obras publicadas por un determinado autor.
- Reducir el tiempo de instalación del programa.
- Mejorar la escalabilidad del sistema.
- Permitir la consulta de una obra a través de un API web.

A diferencia de un documento de requisitos del sistema, la pila del producto nunca se da por completada; está en continuo crecimiento y evolución.

Habitualmente se comienza a elaborar con el resultado de una reunión de "fertilización cruzada" o brainstorming; o un proceso de "Exploración" (Xtreme Programming) donde colabora todo el equipo a partir de la visión del propietario del producto.

El formato de la visión no es relevante. Según los casos, puede ser una presentación informal del responsable del producto, un informe de requisitos del departamento de marketing, etc. Sí que es importante sin embargo disponer de una visión real, comprendida y compartida por todo el equipo.

La pila evolucionará de forma continua mientras el producto esté en el mercado, para darle valor de forma continua, y mantenerlo útil y competitivo. Para dar comienzo al desarrollo se necesita una visión de los objetivos de negocio que se quieren conseguir con el proyecto, comprendida y conocida por todo el equipo, y elementos suficientes en la pila para llevar a cabo el primer sprint. [Palacio, 2011]

2.5.1.2. Formato de la Pila del Producto

El desarrollo ágil prefiere la comunicación directa, a la comunicación con documentos.

La pila del producto no es un documento de requisitos, sino una herramienta de referencia para el equipo. Si se emplea formato de lista, es recomendable que al menos incluya la siguiente información en cada línea:

- Identificador único de la funcionalidad o trabajo.
- Descripción de la funcionalidad.
- Campo o sistema de priorización.
- Estimación

Dependiendo del tipo de proyecto, funcionamiento del equipo y la organización, pueden resultar aconsejables otros campos:

- Observaciones
- Criterio de validación
- Persona asignada
- Nº de Sprint en el que se realiza
- Módulo del sistema al que pertenece. Etc.

Es preferible no adoptar ningún protocolo de trabajo de forma rígida. El formato del product backlog no es cerrado. Los resultados de Scrum Management no dependen de la rigidez en la aplicación del protocolo, sino de la institucionalización de sus principios y la implementación en un formato adecuado a las características de la empresa y del proyecto. [Palacio, 2011] (Ver figura 2.2.1)

Id	Prioridad	Descripción	Est.	Por
1	Muy alta	Plataforma tecnológica	30	AR
2	Muy alta	Interfaz usuario	40	LR
3	Muy alta	Un usuario se registra en el sistema	40	LR
4	Alta	El operador define el flujo y textos de un expediente	60	AR
5	Alta	Etc...	999	XX

Figura # 2.2.1: Pila del Producto

Fuente: [Palacio, 2011]

2.5.1.3. Pila del Sprint

La pila del sprint, (sprint backlog en inglés) es la lista que descompone las funcionalidades de la pila del producto en las tareas necesarias para construir un incremento: una parte completa y operativa del producto.

La realiza el equipo durante la reunión de planificación del sprint, asignando cada tarea a una persona, e indicando en la misma lista cuánto tiempo falta aún para que la termine.

Es útil porque descompone el proyecto en unidades de tamaño adecuado para determinar el avance a diario, e identificar riesgos y problemas sin necesidad de procesos complejos de gestión.

Es también una herramienta de soporte para la comunicación directa del equipo.

Realizada de forma conjunta por todos los miembros del equipo.

- Cubre todas las tareas identificadas por el equipo para conseguir el objetivo del sprint.
- Sólo el equipo lo puede modificar durante el sprint.
- El tamaño de cada tarea está en un rango de 2 a 16 horas de trabajo.
- Es visible para todo el equipo. Idealmente en una pizarra o pared en el mismo espacio físico donde trabaja el equipo. [Palacio , 2011] (Ver figura 2.2.3)

SPRINT	INICIO	DURACIÓN	
1	1-mar-07	12	J
			23
			276

SPRINT BACKLOG			
Tarea	Estad.	Responsal	
Descripción de la tarea 1	Terminada	Luis	16
Descripción de la tarea 2	Terminada	Luis	12
Descripción de la tarea 3	Terminada	Luis	4
Descripción de la tarea 4	Terminada	Elena	8
Descripción de la tarea 5	Terminada	Elena	16
Descripción de la tarea 6	Terminada	Elena	6
Descripción de la tarea 7	Terminada	Antonio	16
Descripción de la tarea 8	Terminada	Antonio	16
Descripción de la tarea 9	Terminada	Antonio	12
Descripción de la tarea 10	En curso	Luis	12
Descripción de la tarea 11	Pendiente	Luis	8

Figura # 2.2.3: Pila del Sprint

Fuente: [Palacio, 2011]

2.5.1.4. Casos de uso UML

Un caso de uso representa una unidad funcional coherente de un sistema, subsistema o clase.

En un caso de uso uno o más actores interactúan con el sistema que realiza algunas acciones.

Elementos de un modelo de casos de uso:

- Actores
- Casos de uso
- Relaciones

Un actor podrá ser cualquier cosa que se comunica (interacciona) con el sistema y que es externo a él.

Los actores no necesariamente coinciden con los USUARIOS. Un usuario puede interpretar distintos roles, correspondientes a distintos actores.

Los actores representan papeles (ROLES) que interpretan personas, periféricos u otros sistemas cuando el sistema está en uso.

Un actor podrá desempeñar distintos papeles dependiendo del caso de uso en que participe.

Un actor representa un conjunto coherente de papeles que los usuarios de una entidad (sistema, subsistema, clase) pueden desempeñar al interactuar con la misma.

La especificación de un caso de uso debe describir el modo en que un actor interactúa con el sistema.

Es una narración que describe el rol desempeñado por el actor en su interacción con el sistema. Lo más importante de los casos de uso es su descripción, mucho más que los diagramas de casos de uso.

Aunque hay descripciones de media página, y algunas de 30, es más habitual que ocupen entre 5 y 15 páginas.

Los diagramas de casos de uso muestran las relaciones entre los casos de uso de un sistema y sus actores. Los diagramas de casos de uso dan solo una visión general del modelo de casos de uso.

El 90% del contenido del modelo de casos de uso está en las descripciones de los casos. Ayudan a interpretar y esclarecer los casos de uso.

Se suelen elaborar durante el análisis inicial del caso de uso. [Vega, 2010]

(Ver tabla 2.2.)

Tabla #2: Casos de uso

Fuente: [Vega, 2010]

Caso de Uso	<< Nombre del CU >>		<< Identificador >>		
Actores	<< Listado de los actores participantes en el CU >> << Podemos indicar quien es el que inicia el CU usando (I) >>				
Tipo	<< Tipo del caso de uso >> << Primario, Secundario u Opcional >> <<Esencial o Real >>				
Referencias	<< Indicamos que requisitos se pueden incluir dentro de este CU >>		<< CU que tienen relación con este >>		
Precondición	<< Condiciones sobre el estado del sistema que tienen que ser ciertas para que se pueda realizar el CU >>				
Postcondición	<< Efectos que de forma inmediata tiene la realización del CU sobre el estado del sistema >>				
Autor	<< Esta línea se podría repetir para mantener una historia de cambios del CU >>		Fecha		Versión
Propósito					
<< Descripción general del CU (Suficiente con una línea) >>					
Resumen					
<< Descripción de alto nivel del flujo normal (básico) del caso de uso (Suficiente con un pequeño párrafo) >>					

Otros datos			
Frecuencia esperada	<< Numero de veces que se realiza el CU por unidad de tiempo >>	Rendimiento	<< Rendimiento esperado de la secuencia de acciones del CU >>
Importancia	<< Importancia de este CU en el sistema (vital, alta, moderada, baja) >>	Urgencia	<< Urgencia en la realización de este CU, durante el desarrollo (alta, moderada, baja) >>
Estado	<< Estado actual del CU en el desarrollo >>	Estabilidad	<< estabilidad de los requisitos asociados a este CU (alta, moderada, baja) >>

Comentarios
<< Comentarios adicionales sobre este CU >>

Curso Normal (Basico)			
1	Actor 1: Acción realizada por el actor		
2	Actor 2: Acción realizada por el actor	3	Acción realizada por el sistema
		N	Cuando se realiza la inclusión de otro caso de uso lo representaremos de la forma. Incluir (CU_identificador. CU_Nombre)
	<< Se incluyen la secuencia de acciones realizadas por los actores que intervienen en el CU , se usaran, frases cortas, que describan el dialogo entre los actores y el sistema>> << Se pueden añadir referencias a elementos de un boceto del Interfaz del Usuario >>		<< Se incluyen la secuencia de acciones que realiza el sistema ante las acciones de los actores >>

Cursos Alternos	
1a	Descripción de la secuencia de acciones alternas a la acción 1 del Curso Normal
1b	
	<< Secuencia de los cursos alternos del CU >>

2.5.1.5. Elementos de los Diagramas

- Actores
- Casos de uso
- Relaciones
- Puede aparecer un rectángulo que muestre los límites del sistema

Los casos de uso se representan mediante elipses con el nombre del caso

Los actores pueden representarse mediante unos monigotes o mediante rectángulos en que se indique actor

En los diagramas, tanto los actores como los casos de uso representan no las instancias particulares, sino los conjuntos de todos los actores de un tipo y de todos los escenarios. [Vega, 2010] (Ver figura 2.2.4)

Figura # 2.2.4: Elementos de diagramas

Fuente: [Vega, 2010]

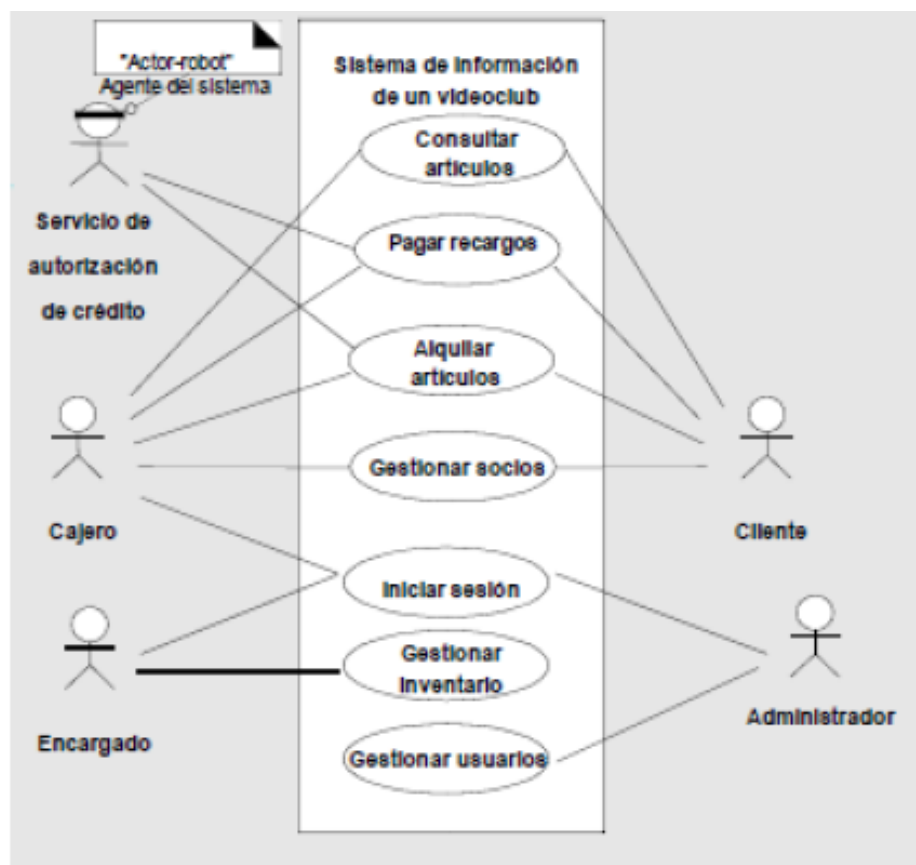
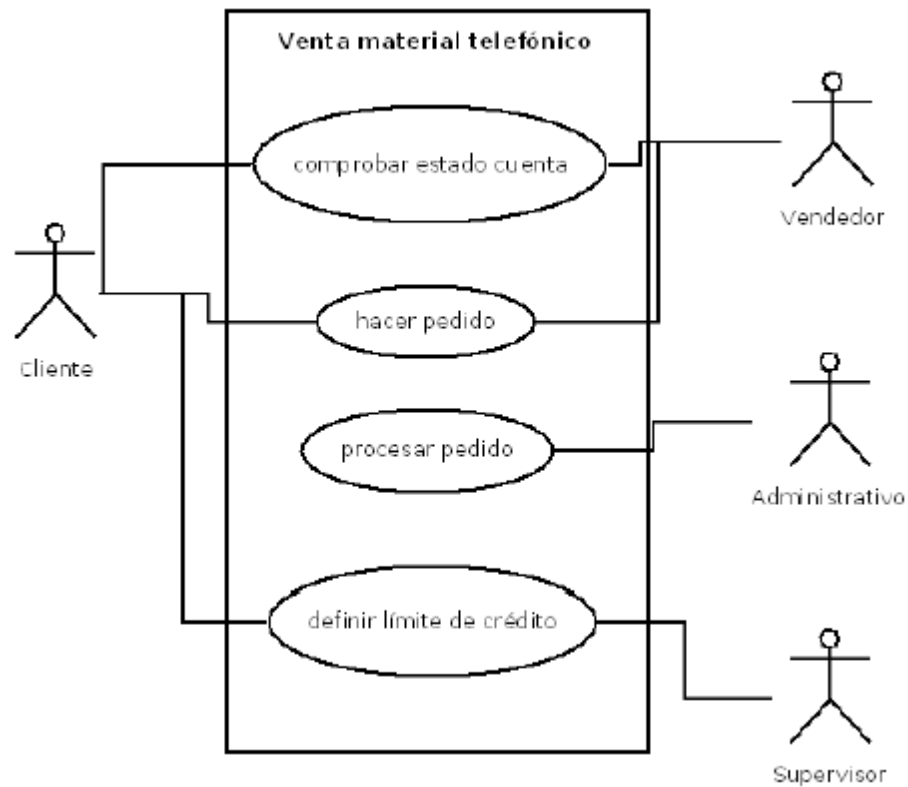


Figura # 2.2.4: Elementos de diagramas

Fuente: [Vega, 2010]



2.5.1.6. Asociaciones entre actores y casos de uso

Las asociaciones entre actores y casos de uso: se representan mediante una línea continua significan la participación del actor en el caso de uso pueden indicarse restricciones de cordialidad. [Vega, 2010]

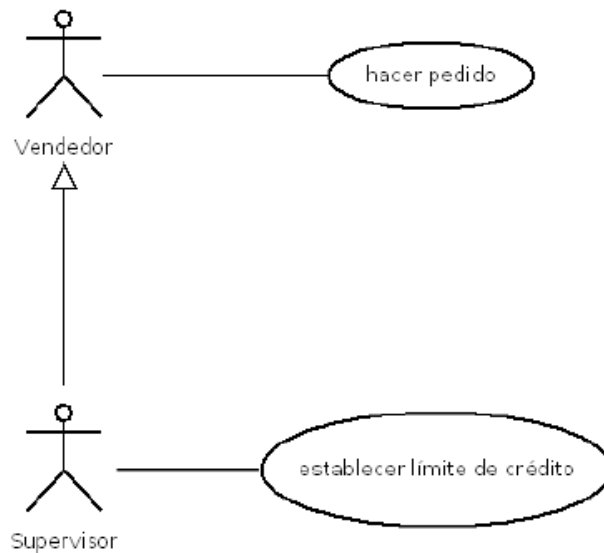
2.5.1.7.- Generalización - Especificación entre actores

Indicaran que un actor es más general que otro si A es una especialización de B, una instancia de A podrá comunicarse con los mismos casos de uso que B

(Ver figura 2.2.5)

Figura # 2.2.5: Especificación entre autores

Fuente: [Vega, 2010]



2.6. HERRAMIENTAS DEL DESARROLLO

Para el desarrollo de la metodología Scrum se utiliza diferentes herramientas como se mencionara a continuación.

2.6.1. Microsoft Visual Studio

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc., a lo cual hay que sumarle las nuevas capacidades online bajo Windows Azure en forma del editor Monaco.

Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y consolas, entre otros.

2.6.1.1. Visual Studio 2013

Fue la primera revisión de Visual Studio en incluir una versión "Community", que básicamente ofrece las mismas capacidades que la versión "Professional" pero limitando su uso a empresas

de pequeño tamaño, desarrolladores de software libre y estudiantes. La gran ventaja de esta versión de Visual Studio es que es gratuita.

Permite trabajar con los frameworks:

- .NET Framework 2.0
- .NET Framework 3.0
- .NET Framework 3.5
- .NET Framework 4.0
- .NET Framework 4.5
- .NET Framework 4.5.1
- .NET Framework 4.5.2

2.6.2. SQL Server

Microsoft **SQL Server**. 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. Microsoft SQL Server es una plataforma de base de datos que se utiliza en el procesamiento de transacciones en línea (OLTP) a gran escala, el almacenamiento de datos y las aplicaciones de comercio electrónico; es también una plataforma de Business Inteligencia para soluciones de integración, análisis y creación de informes de datos. SQL Server 2008 incluye varias características de seguridad configurables y de gran precisión. Estas características permiten a los administradores implementar una defensa optimizada para los riesgos de seguridad específicos de su entorno. Introduce "estudios" que le ayudarán en las tareas de programación y administración: SQL Server Management Studio y Business Intelligence Development Studio. En Management Studio, se desarrolla y administra SQL Server Database Engine (Motor de base de datos de SQL Server) y soluciones de notificación, se administran las soluciones de Analysis Services implementadas, se administran y ejecutan los paquetes de Integration Services, y se administran los servidores de informes y los informes y modelos de informe de Reporting Services.

2.7 PRUEBA

2.7.1. Prueba Unitaria

Un test unitario (Unit Test) es un trozo de código desarrollado con el único objetivo de verificar que una rutina o función de nuestro código está funcionando según esperamos.

Debido al gran éxito que esta metodología está teniendo en el mundo de la ingeniería del software, son muchas y variadas las herramientas que se han ido desarrollando hasta el día de hoy. Algunas han surgido en forma de Frameworks, otras en formato plugin bajo el control de algún browser o incluso como librerías para su uso en plataformas de programación.

Entre los que podemos citar a:

- JUnit: Entorno de pruebas para Java creado por Erich Gamma y Kent Beck.
- HttpUnit: viene a ser un cliente web programable. Se puede usar de forma aislada o como complemento del framework Junit.
- DBUnit: en realidad es una extensión de JUnit que tiene como ventaja el poder tener en cuenta la existencia de una base de datos a la hora de realizar las pruebas de los test.
- SimpleTest: Entorno de pruebas para aplicaciones realizadas en PHP.
- PHPUnit: Framework para realizar pruebas unitarias en PHP.
- CPPUNIT: Versión del Framework para lenguajes C/C++. La mayoría de las herramientas tienen assert, que son afirmaciones que son usada para comprobar suposiciones en el programa, colocada donde el desarrollador considera que su enunciado es siempre verdadero. Esto da lugar a que existan afirmaciones y por lo tanto condiciones antes (precondiciones) y después (post-condiciones) de la ejecución de determinadas líneas de código, lo que da lugar a las pruebas unitarias.

CAPITULO III MARCO APLICATIVO

Para el desarrollo de este proyecto se utiliza la metodología Ágil SCRUM con las siguientes fases:

- Preparación del Proyecto
- Planificar un Sprint
- El desarrollo del Sprint
- Diagrama detallado de las fases del SCRUM

3.1. Preparación del Proyecto

3.1.1. Requisitos y Visión del Producto

Scrum emplea este formato para registrar los requisitos mediante la pila del producto

3.1.1.1.- Pila del Producto

Se inicia con la creación del Backlog del producto (pila del producto) lista de requisitos de usuario que a partir de la visión inicial del producto crece y evoluciona durante el desarrollo.

(Ver acápite 2.5.1.3. del capítulo II) y (Ver tabla 3)

DESCRIPCIÓN	MODULO	PRIORIDAD
Base de datos independiente	Plataforma	Alta
Soporte para la actualización en el registro	Plataforma	Alta
Soporte para el usuario y Administrador	Plataforma	Alta
Soporte para administración y control de Insumos Activos	Plataforma	Alta
Control de acceso seguro y diferenciado a usuarios	Plataforma	Media
Registro para Insumos de Laboratorio nuevos	Plataforma	Media
Registro y/o Modificación para usuarios nuevos	Plataforma	Media

Tabla # 3.1: Backlog del Producto (Pila del Producto)

3.2.- PLANIFICAR UN SPRINT

3.2.1.- Pila del SPRINT

En la Pila del Sprint (sprint backlog) los requerimientos establecidos para el desarrollo del sistema son los siguientes: (Ver tabla 3.1).

ID	DESCRIPCIÓN SPRINT N°	0	1	2	3
SPRINT 0 Diseño					
1	Configurar la Base de Datos SQL Server	2	0	0	0
2	Creación de Formularios Visual Studio	7	0	0	0
3	Configurar entorno programación	2	0	0	0
4	Crear estructura en Visual Studio	2	0	0	0
5	Crear estructura base de datos	5	0	0	0
SPRINT N° 1 Control Sistema / Administración					
6	Implementar ingreso Sistema	0	3	0	0
7	Implementar perfil de usuario	0	3	0	0
8	Implementar administración de usuarios	0	3	0	0
9	Implementar administración de Asegurados y/o Beneficiarios	0	3	0	0
10	Implementar administración de Insumos Activos	0	3	0	0
SPRINT N° 2 Control de Materiales / Administración /Usuario					
11	Implementar ingreso de Insumos a laboratorio /Administrador	0	0	3	0
12	Implementar ingreso de Insumos generales /Administrador	0	0	3	0
13	Implementar registro de nuevos Insumos por laboratorio y general /Administrador	0	0	3	0
14	Implementar eliminar de Insumos/Administrador	0	0	3	0
15	Implementación de Insumos /Usuario	0	0	3	0

SPRINT N° 3 Información de Datos					
16	Reportes	0	0	0	6
17	Pruebas Usuarios /Administrador	0	0	0	8
	Esfuerzo necesario hasta la entrega de software	62 días			

Tabla # 3.2: Pila del sprint

3.2.2.- Herramientas de Trabajo

Para el desarrollo del software se utilizan las siguientes herramientas:

- Microsoft SQL Server - SQL Server 2008, para el desarrollo de consultas SQL y la administración de la base de datos.
- Microsoft Visual Studio - Visual Studio 2013 para el desarrollo de la programación.
- Microsoft Windows, Sistema Operativo desde Windows 7 al Windows 10, como proyección del proyecto.

3.2.3.- Análisis de Riesgo

Un riesgo es la probabilidad de que ocurra algo adverso (Ver tabla 3.2)

3.3.- Diseño Interfaces

El diseño de la interfaz gráfica del sistema se enfoca en implementar gráficamente los requisitos establecidos en el diseño del sistema. Las interfaces de las distintas secciones de la página web fueron enfocadas tanto a la funcionalidad como a la facilidad de manejo de las funciones del sistema, teniendo en cuenta las sugerencias y recomendaciones del usuario.

3.3.1.- Diseño de Interfaz Gráfico del Sistema



Figura # 3.1: Ingreso Al Sistema

La figura 3.1 muestra la página principal del sistema donde debe logearse el usuario ya sea este administrador o un usuario encargado. (Ver figura 3.1)



Figura # 3.1.1: Ingresando al Sistema

Estas interfaces están enfocadas al ingreso al sistema de laboratorio. (Ver figura 3.1.1)



Figura # 3.2: Ingresando al Sistema del Servicio de Laboratorio

En la figura 3.2 muestra la interfaz del usuario donde tienes los siguientes módulos:

- Inicio: Describe el inicio para el trámite.
- Búsqueda: Realiza la búsqueda de los insumos de laboratorio.
- Modificar: donde se modifica los insumos de laboratorio o al usuario.



Figura # 3.3: Modificación de Datos

En la figura 3.2 muestra la interfaz de la modificación en los datos de los insumos de laboratorio, donde tienes los siguientes módulos:

- **Agregar:** Agrega un Ítem a la base de datos de un Insumo de Laboratorio.
- **Modificar:** Modifica un Ítem a la base de datos de un Insumo de Laboratorio.
- **Eliminar:** Elimina un Ítem a la base de datos de un Insumo de Laboratorio.
- **Configuración de Usuario:** Modifica al Usuario para agregar nuevos datos.

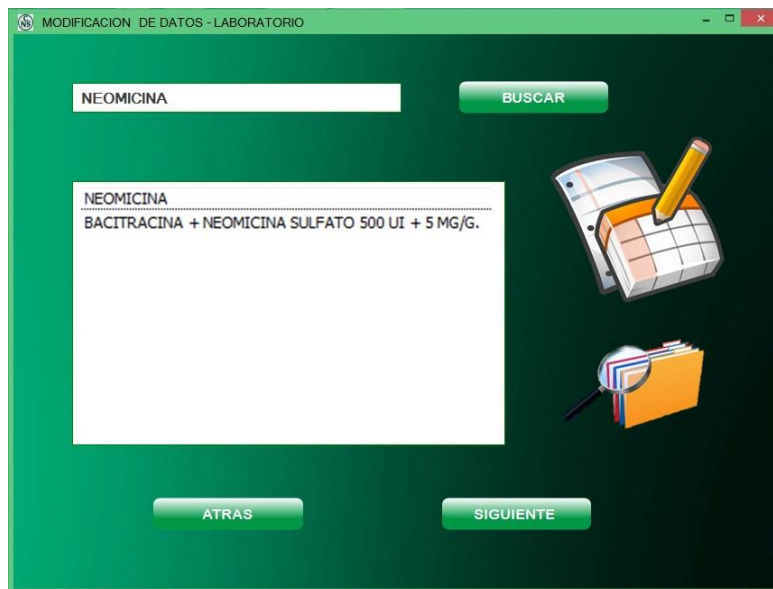
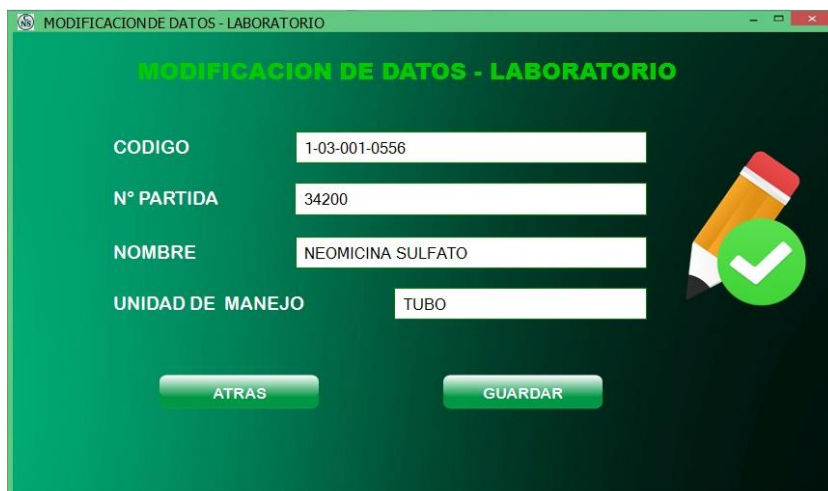


Figura # 3.4: Modificación en los Datos

En la figura 3.4 muestra la búsqueda de un insumo de laboratorio para su respectiva modificación en los datos.



CODIGO	1-03-001-0556
N° PARTIDA	34200
NOMBRE	NEOMICINA SULFATO
UNIDAD DE MANEJO	TUBO

ATRAS GUARDAR

Figura # 3.5: Modificación en los Datos

En la figura 3.5 muestra el Ítem completo del Laboratorio donde menciona su código, numero de partida, su nombre y la unidad de manejo, donde se procede la modificación.



CODIGO	FCR241289
NOMBRE (S)	RAPHAEL
APELLIDO PATERNO	FLORES
APELLIDO MATERNO	CHINO
CONTRASEÑA	*****

ATRAS GUARDAR

Figura # 3.6: Configuración de Usuario

En la figura 3.6 muestra la configuración de un nuevo usuario para el respectivo control en el sistema donde debe agregar su código, nombre(s), apellido paterno, apellido materno y la contraseña para entrar al sistema y se procede la modificación.

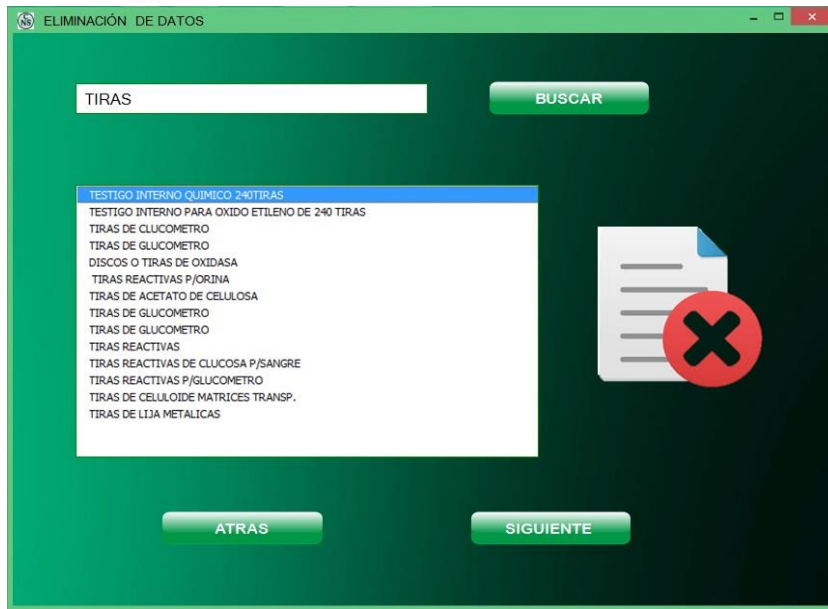


Figura # 3.7: Eliminación de Datos

En la figura 3.7 se introduce un ítem y muestra un listado para su respectiva eliminación.



Figura # 3.8: Eliminación de Datos

En la figura 3.8 muestra el Ítem completo del Laboratorio donde menciona su código, numero de partida, su nombre y la unidad de manejo, donde se procede a la eliminación.



Figura # 3.9: Agregación de Datos

En la figura 3.9 muestra el Ítem completo del Laboratorio donde se agrega su código, numero de partida, su nombre y la unidad de manejo, donde se procede a agregar los datos correspondientes.

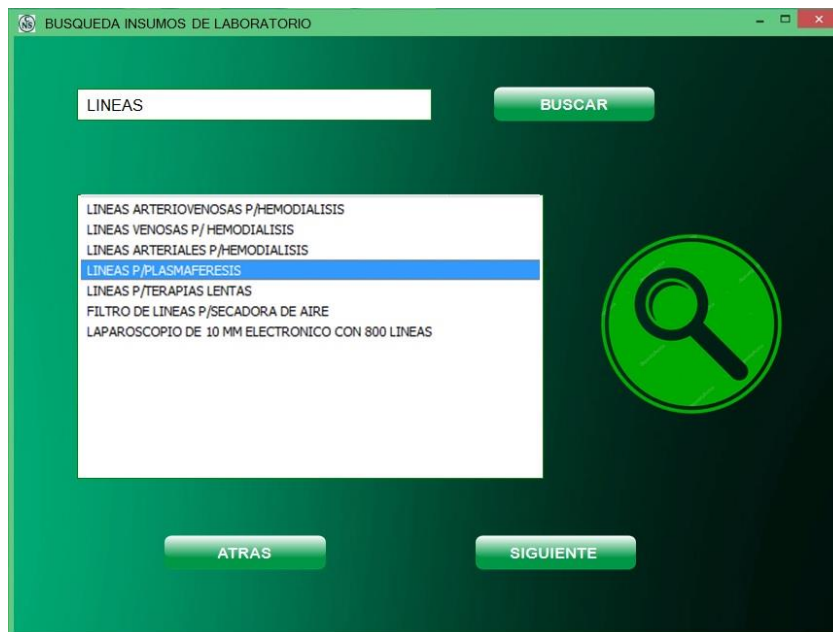


Figura # 3.10: Búsqueda de Insumos de Laboratorio

En la figura 3.10 muestra la búsqueda de algún ítem para determinar si le corresponde dicho insumo al servicio de laboratorio, o si se acerca al insumo buscado.

INICIO DE TRAMITE	
FECHA	09/03/2020
NOMBRE JEFATURA CNP	DRA. MARIANA ARIAS SALAS
NOMBRE DEL ASEGURADO	EFRAIN CHAMBI CONDORI
N° CITE	0024/2020
FECHA DE RECEPCION	20/02/2020
N° DE FACTURAS	1

Figura # 3.11: Inicio de Tramite

En la figura 3.11 muestra el inicio del trámite a realizar donde se pide la fecha, el nombre de jefatura de la comisión nacional de prestaciones (instancia superior), nombre del asegurado, numero de cite, fecha de recepción, y el número de facturas.

FACTURAS	
N° DE FACTURA	4596
FECHA DE LA FACTURA	06/11/2019
TOTAL	3.569,53
TOTAL RECONOCIDO	2.548,12
TOTAL NO RECONOCIDO	0,00
OBSERVACIONES	

Figura # 3.12: Facturas

En la figura 3.12 muestra el llenado de la factura(s) donde se pide la fecha, el número de factura, fecha de la factura, el total de la factura, el total reconocido, el total no reconocido (en caso de coordinar con las instancias pertinentes como ser Suministros o Farmacia), y las observaciones correspondientes (si no cumple con establecido ejemplo si tiene correcto el NIT de la Caja Nacional de Salud o el nombre completo en la factura).

SERVICIO NACIONAL DE LABORATORIO

LA PAZ 30 DE SEPTIEMBRE DEL 2020 SNL-219/2019

SEÑOR(A):
Dra. Mariana Arias Sanchez
 SECRETARIA COM. NAL. DE PRESTACIONES C.N.S.

En atención nota CITE: **CNP-1509/2019** recibida en fecha **27/069/2020**, solicita la verificación de facturas para establecer el monto total de reembolso, por concepto de estudios de laboratorio, emitiendo el Servicio Nacional de Laboratorio, el siguiente criterio:

N° FACTURA	FECHA	IMPORTE TOTAL FACTURA	IMPORTE RECONOCIDO	IMPORTE NO RECONOCIDO	OBSERVACIONES
4534	11/05/2017	150,00	150,00	0,00	
4538	11/05/2017	150,00	150,00	0,00	
		TOTAL	300,00		

Observaciones:
 Se verifico la carpeta y se corrobora el detalle de las pruebas de laboratorio realizadas a la paciente.

En lo que corresponde al Servicio Nacional de Laboratorio, concluimos con informar que el monto correspondiente a exámenes de laboratorio de acuerdo a documentación adjunta y según reglamento es de **Bs. 300,00**, por lo que se remite para la consideración de la Comisión Nacional de Prestaciones quien determine la procedencia o no del reembolso.

Dra. Gloria Ruth Cruz Castro
 JEFE a.i. NAL. DE LABORATORIO

Adjunto fojas 164
 CC: ARCS-PRIV
 CCGRtr241289

Figura # 3.13: Reporte

En la figura 3.13 muestra el Reporte generado con un ejemplo, y cómo se prepara para su respectiva impresión.

CAPITULO IV CALIDAD Y SEGURIDAD

La evaluación del trabajo fue realizada bajo las métricas de calidad de la ISO 9126.

4.1. Calidad de Software

La norma ISO/IEC 9126 permite especificar y evaluar la calidad del software desde diferentes criterios asociados con adquisición, requerimientos, desarrollo, uso, evaluación, soporte, mantenimiento, aseguramiento de la calidad y auditoria de software. Los modelos de calidad para el software se describen así:

Calidad interna y externa: Especifica 6 características para calidad interna y externa, las cuales, están subdivididas. Estas divisiones se manifiestan externamente cuando el software es usado como parte de un sistema Informático, y son el resultado de atributos internos de software.

Calidad en uso: Calidad en uso es el efecto combinado para el usuario final de las 6 características de la calidad interna y externa del software. Especifica 4 características para la calidad en uso.

Al unir la calidad interna y externa con la calidad en uso se define un modelo de evaluación más completo, se puede pensar que la usabilidad del modelo de calidad externa e interna pueda ser igual al modelo de calidad en uso, pero no, la usabilidad es la forma como los profesionales interpretan o asimilan la funcionabilidad del software y la calidad en uso se puede asumir como la forma que lo asimila o maneja el usuario final. Si se unen los dos modelos, se puede definir que los seis indicadores del primer modelo tienen sus atributos y el modelo de calidad en uso sus 4 indicadores pasarían hacer sus atributos, mirándolo gráficamente quedaría así: (Ver figura 2.4)



Figura # 4: Norma de Evaluación ISO/IEC 9126

Fuente: [Borbón, 2013]

Las definiciones se dan para cada característica y subcaracterísticas de calidad del software que influye en la calidad. Para cada característica y subcaracterísticas, la capacidad del software es determinada por un conjunto de atributos internos que pueden ser medidos. Las características y subcaracterísticas se pueden medir externamente por la capacidad del sistema que contiene el software. [Borbón, 2013]

4.2. Funcionalidad

Funcionalidad es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas.

La funcionalidad se divide en 5 criterios:

Adecuación: La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario.

Exactitud: La capacidad del software para hacer procesos y entregar los resultados solicitados con precisión o de forma esperada.

Interoperabilidad: La capacidad del software de interactuar con uno o más sistemas específicos.

Seguridad: La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados

Conformidad de la funcionalidad: La capacidad del software de cumplir los estándares referentes a la funcionalidad. [Borbon, 2013]

4.3. Confiabilidad

La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizando en condiciones específicas. En este caso la confiabilidad se amplía sostener un nivel especificado de funcionamiento y no una función requerida. (Ver figura 2.4.3)



Figura # 4.1: Característica de Confiabilidad

Fuente: [Borbon, 2013]

La confiabilidad se divide en 4 criterios:

Madurez: La capacidad que tiene el software para evitar fallas cuando encuentra errores. Ejemplo, la forma como el software advierte al usuario cuando realiza operaciones en la unidad de cd vacía, o cuando no encuentra espacio suficiente el disco duro donde esta almacenando los datos.

Tolerancia a errores: La capacidad que tiene el software para mantener un nivel de funcionamiento en caso de errores.

Recuperabilidad: La capacidad que tiene el software para restablecer su funcionamiento adecuado y recuperar los datos afectados en el caso de una falla.

Conformidad de la fiabilidad: La capacidad del software de cumplir a los estándares o normas relacionadas a la fiabilidad. [Borbon, 2013]

4.4.- Seguridad Informática

4.4.1.- ¿Por qué son necesarios los mecanismos de Seguridad?

La seguridad es fundamental a la hora de afrontar tareas que se realizan en sistemas informáticos ya que son las únicas medidas que pueden garantizar que éstas se realicen con una serie de garantías que se dan por sentado en el mundo físico. Por ejemplo, cuando se guardan cosas en una caja fuerte en un banco real, no se piensa que cualquier persona del mundo puede llegar a ésta de una forma inmediata como si se tratara, en lugar de un banco, de una estación de autobuses. En el mundo intangible de la informática, tan cerca de un servidor están sus usuarios legítimos como los usuarios que hacen uso de la misma red de comunicaciones. Es más, estos

usuarios, en el caso de una red global, se cuentan por millones. Algunos serán “buenos vecinos” pero otros serán agentes hostiles.

4.4.2.- Principios de Seguridad Informática

Para lograr sus objetivos la seguridad informática se fundamenta en tres principios, que debe cumplir todo sistema informático:

- **Confidencialidad:** Se refiere a la privacidad de los elementos de información almacenados y procesados en un sistema informático, Basándose en este principio, las herramientas de seguridad informática deben proteger el sistema de invasiones y accesos por parte de personas o programas no autorizados. Este principio es particularmente importante en sistemas distribuidos, es decir, aquellos en los que los usuarios, computadores y datos residen en localidades diferentes, pero están física y lógicamente interconectados.
- **Integridad:** Se refiere a la validez y consistencia de los elementos de información almacenados y procesador en un sistema informático. Basándose en este principio, las herramientas de seguridad informática deben asegurar que los procesos de actualización estén bien sincronizados y no se dupliquen, de forma que todos los elementos del sistema manipulen adecuadamente los mismos datos. Este principio es importante en sistemas descentralizados, es decir, aquellos en los que diferentes usuarios, computadores y procesos comparten la misma información.
- **Disponibilidad:** Se refiere a la continuidad de acceso a los elementos de información almacenados y procesados en un sistema informático. Basándose en este principio, las herramientas de seguridad informática deber reforzar la permanencia del sistema informático, en condiciones de actividad adecuadas para que los usuarios accedan a los datos con la frecuencia y dedicación que requieran, este principio es importante en sistemas informáticos cuyos compromiso con el usuario, es prestar servicio permanente.
- **Factores de Riesgo:**
- **Ambientales/Físicos:** factores externos, lluvias, inundaciones, terremotos, tormentas, rayos, humedad, calor entre otros.

4.4.3.- Seguridad en la Aplicación

Control de usuario: para tener control sobre los usuarios que entran al sistema evitar el acceso a terceros. El sistema autentifica al usuario mediante sesiones por medio del nombre de usuario y su password.

Desactivación de cuentas: para evitar el acceso de usuarios cuyas cuentas fueron inhabilitadas.

4.4.4.- Seguridad Base de Datos

Control acceso usuarios: para evitar el acceso de usuarios que no tengan que ver con el Servicio de Laboratorio o el departamento de sistemas se instancio una clave con el fin de evitar irregularidades y la disponibilidad de información confiable.

CAPITULO V CONCLUSIONES Y RECOMENDACIONES

A la conclusión del presente proyecto se detallan las conclusiones y recomendaciones. Para llevar a cabo este trabajo, se recabo información exhaustiva del proceso de desarrollo de software, áreas de conocimiento de ingeniería de software de sistemas y otras disciplinas dentro de la informática, se leyó bibliografía de libros, proyectos de grado similares y páginas web relacionadas con el tema.

5.1. Conclusiones

Se ha desarrollado e implementado una herramienta software para el control de la elaboración de Reembolsos, cumpliendo con los siguientes objetivos específicos:

- Implementar tecnologías de arquitectura de software para desarrollar un sistema escalable.

Se desarrolló una arquitectura de software que permite a cualquier otro aplicativo adecuarse fácilmente al sistema, así de esta manera utilizando metodologías de diseño y desarrollo como SCRUM y ATAM.

- Obtener informes confiables, donde se puedan hacer comparaciones con datos pasados, de esta manera pronosticar los rendimientos, y tomar buenas decisiones.

Se obtienen reportes confiables, donde el usuario fácilmente puede comparar datos pasados, controlar los rendimientos.

- Unir toda la información para lograr integrar los datos y de esta manera no perder información.

Toda la información se guarda en la base de datos, teniendo así de esta manera mucha confianza sobre el resguardo de los datos.

- Agilizar el llenado de datos, para lograr mejores tiempos en el trabajo.
- Se diseñó una planilla de ingreso de información, donde fácilmente es muy amigable el sistema al momento de adicionar y guardar los datos.
- Integrar la información en un solo motor de base de datos confiable.

Toda la información se integró en el motor de base de datos SQL Server , la confiabilidad y el prestigio que tiene este motor de base de datos es muy alta, por lo tanto se confía plenamente la información que resguarda.

- Diseño e implementación de una Base de Datos.

El diseño e implementación se lo realizo de una forma que se puedan adherir fácilmente otros subsistemas.

De esta manera se cumple con el objetivo general:

“Desarrollar un sistema para realizar el control eficiente”.

5.2. Recomendaciones

El software está desarrollado y orientado al uso de personas que tengan conocimiento del área de Administración, esto con el fin de mantener la integridad de la información.

Si bien el sistema cuenta con un nivel de confiabilidad y seguridad, es necesario realizar acciones para permitir mantener la madurez del sistema, para ello se recomienda:

Realizar un mantenimiento preventivo periódicamente, esto con el fin de permitir mejoras funcionales del sistema.

El módulo de administración de usuarios, asigna una clave maestra por defecto, que solo el administrador conoce, es recomendable que al finalizar la instalación se realice la asignación de los perfiles correspondientes.

Se recomienda realizar la continuidad del proyecto, agregando nuevos módulos, como se van mencionando.

Por motivos de seguridad y desconfianza de la institución no se consideró el uso de páginas web, ya que la información que se maneja es muy valiosa para ellos y no hubo la necesidad de plantearla.

REFERENCIA BIBLIOGRÁFICA

[ACM01] ARIÑEZ, M. (2001), Sistema de Información Integrado PIL-Chuquisaca S.A. (SII-PIL Chuquisaca),

[ANM06] APAZA, M. (2006), Sistema de Información para la Producción Planificada DELIZIA, UMSA Biblioteca de Informática, Consultado Octubre 07, 2014.

[FDL16] Plan de desarrollo organizacional 2014-2016, Flor de leche S.R.L.

[FFDLQY11]Formula elaborado por GPR, Flor de leche S.R.L.

[FRQY95] Fórmula matemática para el Cálculo del rendimiento, Flor de Leche S.R.L., consultado Julio 5, 2014

[FXS08]Flexibilidad con Scrum, Principios de diseño e implantación de campos de Scrum, Autor: Juan Palacio. Edición Octubre 2008

[MBA97] MACHICADO, A. (1997), Sistema de Almacenes y Producción SOCOVIAL, UMSA Biblioteca de Informática, Consultado Octubre 07, 2014.

[MMPR12]CHAVEZ, E, (2013) tesis: modelo matemático para predecir el rendimiento del queso a partir de la composición química de la dieta, Consultado Octubre 04,2014

[PURR10]RODRIGUEZ, R. (2010) Pruebas Unitarias, Universidad Nacional de San Juna (Consultado Marzo, 2015)

[PRESS] Pressman Roger, 5ta Ingeniería de software

Referencia Web

Caja Nacional de Salud

<http://www.cns.gob.bo/Institucion/misionvisionDirec>

Metodología Scrum

<https://proyectosagiles.org/que-es-scrum/>

Planillas de Proyecto

<https://sistemas.uniandes.edu.co/es/proyectos-grado>

Calidad de Software, (Consultado Abril, 2015)

http://es.wikipedia.org/wiki/Calidad_de_software

[APO09] Árbol de problemas y Objetivos, Gobierno de Chile, Consejo Nacional de la Cultura y las Artes. Guía para la gestión de proyectos culturales. Valparaíso, 2009, extraído el 06 de Octubre de 2014.

http://docencia.unet.edu.ve/Coordinaciones/SComunitario/archivos/Arbol_problemas_objetivos.doc

[CAFU14]Calidad de Software, funcionalidad y usabilidad, (Consultado Abril, 2015)

<http://albertolacalle.com/hci/funcionalidad-usabilidad.html>

[CBML] Curso breve de marco lógico (visión general del curso), extraído el 08 de octubre de 2014 80

http://www.google.com.bo/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CCMQFjAB&url=http%3A%2F%2Fwww.solucionesong.org%2Fimg%2Fforos%2F4ce164c4dd0de%2FMarco_Logico.ppt&ei=zfs1VJv3PPPksATfsYK4BQ&usq=AFQjCNHXLx0MJJeOWJqSSe9pnDYLNBw-A&sig2=2U_NiNQyVITugG1ToMHwrA&bvm=bv.76943099,d.cWc&cad=rja

[CDS12]Calidad de Software,(Consultado Abril, 2015)

http://es.wikipedia.org/wiki/Calidad_de_software

[CLD10]Cliente Servidor de dos capas, (Consultado Enero, 2015)

<http://www.monografias.com/trabajos89/cliente-servidor-dos-capas/cliente-servidor-dos-capas.shtml>

[COSC95], COCOMO, University of Southern California. (Consultado Marzo, 2015)

http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html

DOCUMENTACIÓN