

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



PROYECTO DE GRADO

**“SISTEMA DE CONTROL Y SEGUIMIENTO A ANALISIS DE MUESTRAS DE
MINERALES”**

CASO: EMPRESA ALEX STEWART ASSAYERS BOLIVIA

PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMATICA

MENCION: INGENIERIA DE SISTEMAS INFORMATICOS

POSTULANTE: Univ. BEATRIZ YOVANA GUACHALLA CALLE

TUTOR: Lic. FATIMA C. DOLZ SALVADOR

REVISOR: Lic. ALDO RAMIRO VALDEZ ALVARADO

LA PAZ - BOLIVIA

DEDICATORIA

Dedico este Proyecto a mi padre Abraham Guachalla a mis hermanos Erika, Mariana y Cristian, quienes me impulsaron a seguir adelante, por estar siempre a mi lado, y hacer posible el logro de uno de mis objetivos.

.....A ellos que son mi razón de existir

Beatriz Guachalla

AGRADECIMIENTOS

Deseo expresar mi profundo agradecimiento a las siguientes personas:

A Dios por darme vida para llegar y disfrutar de la culminación de mi carrera.

A Lic. Fátima Consuelo Dolz de Moreno, por su apoyo, dedicación y guía en el desarrollo y conclusión del presente proyecto.

A Lic. Aldo Ramiro Valdez Alvarado, por dedicar gran parte de su tiempo a la revisión y corrección de este proyecto, por su calidad humana y amabilidad.

A los responsables de Alex Stewart Assayers Ltda. por su colaboración, confianza y apoyo en el transcurso del desarrollo de este proyecto.

A los docentes de la carrera de Informática por los conocimientos transmitidos durante mi ciclo de formación en la universidad.

A mi familia por su constante apoyo y comprensión a las decisiones de mi vida.

A mis amigos y compañeros por el compañerismo brindado

Muchas Gracias....

RESUMEN

Actualmente la Empresa Alex Stewart (Assayers), brinda servicios de control de calidad en todo tipo de establecimiento industrial y lugar donde sea necesario realizar tareas de inspección, pesaje, muestreo y análisis como apoyo al proceso comercial de la cadena de suministros, abarcando todo tipo de mercaderías.

Debido al crecimiento de la empresa, y el incremento de personal, se producen retrasos en el procesamiento de la información, a causa de que algunos procesos se siguen realizando en forma manual.

En este sentido se ha desarrollado el Sistema de control y seguimiento a análisis de muestras de mineral, realizando el mismo a través de la programación orientada a objetos haciendo el uso de la metodología RUP, el cual esta dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental, como una herramienta de apoyo se utilizó el Lenguaje de Modelado Unificado "UML".

El sistema tiene una importancia significativa al constituirse como una herramienta de software de apoyo, en el control y seguimiento a análisis de muestras de minerales. Coadyuvando de forma eficiente y transparente a la elaboración y obtención de información organizada, confiable y oportuna.

El sistema desarrollado beneficiara de gran manera al manejo de la información, de las solicitudes de análisis, resultados de análisis, clientes y usuarios y toda información que se adquiriera durante el proceso de análisis que se ha solicitado a la empresa.

Capítulo I

Marco Introdutorio

1.1 INTRODUCCIÓN

La información que hoy en día se produce en cantidades considerables obliga a muchas entidades tanto gubernamentales como privadas a organizar toda su información, pero estas, no cumplen los objetivos propuestos para satisfacer la demanda de información de los usuarios en cada institución, es por eso que se ven obligadas a establecer agrupaciones con fines y objetivos comunes que gestionen fácilmente la información a través de las consultas y a optimizar la comunicación¹.

La razón principal es la manera en que han evolucionado las computadoras, basadas en las tecnologías de información y sistemas. La mayoría de las organizaciones hacen lo posible por conseguir buena información, pero el logro de ese objetivo depende fundamentalmente de su arquitectura actual, tanto de hardware como de software.

El desarrollo de la informática, su aplicación en distintas áreas del saber humano y su acogida e incorporación en diferentes ámbitos empresariales relacionados con la industria, comercio, educación, gobierno y otros, ha permitido manejar la complejidad de la información existente en éstos ámbitos, a través de la ingeniería de software, ingeniería de sistemas, modelos de calidad total y otros enfoques, enfrentando los problemas emergentes y solucionándolos con el empleo de métodos y herramientas adecuadas.

El uso de los sistemas en el país para los diferentes problemas que afectan a nuestro medio, han sido de poco interés en los problemas que afectan a nuestra sociedad, por eso es que el presente proyecto trata de introducirse a la sociedad

¹ [Fernando Arteaga, 2001]

para resolver problemas que ayuden al profesional, en el área de control y seguimiento de manejo de información de los datos.

El presente proyecto de grado, consiste en desarrollar un sistema que permitirá llevar un control y hacer un seguimiento a la información de los análisis a muestras de minerales, sistema elaborado para la empresa “Alex Stewart Assayers Bolivia” dedicada al control y calidad de minerales. El propósito de este sistema es almacenar toda la información que ingrese en el área de almacenes, en donde se guardan todas las muestras de minerales que el cliente envía para su respectivo análisis, posteriormente estas muestras de minerales son enviadas al laboratorio para su análisis y una vez obtenidos los resultados del análisis se procede a emitir una factura con el detalle de los resultados de dicho análisis. De esta manera la empresa “Alex Stewart Assayers Bolivia”, cuenta con mayor información sobre los análisis realizados en sus laboratorios.

1.2 ANTECEDENTES

En el desarrollo tradicional de un Sistema de Información, se construye toda la arquitectura en función a los datos, datos que fueron identificados ya en una etapa preliminar.

Y al desarrollar un sistema se podría considerar como pilares a los datos, una interactuaran con el usuario, hasta la conclusión de dicho sistema.

1.2.1 ANTECEDENTES INSTITUCIONALES

Actualmente la Empresa Alex Stewart (Assayers) Bolivia, brinda servicios independientes de control de calidad en todo tipo de establecimiento industrial y lugar donde sea necesario realizar tareas de inspección, pesaje, muestreo y análisis como apoyo al proceso comercial de la cadena de suministros, abarcando todo tipo de mercaderías, ya sean materias primas o procesadas, a granel o fraccionadas.

Se cuenta con un equipo de jóvenes profesionales multidisciplinarios, especialistas en los servicios que ofrece la empresa.

Además la empresa cuenta con un laboratorio de preparación de muestras (geoquímicas) local, con el respaldo y libre disponibilidad de los laboratorios especializados en países limítrofes para el análisis de productos relacionados con los servicios que ofrece la empresa.

Los servicios con los que hoy cuenta y pone a disposición son:

- Servicios en Minería
- Servicios de Laboratorio
- Servicios en Agricultura
- Servicios Industriales y Ambientales
- Estudios de Ingeniería de Métodos
- Estudios de Pre – Factibilidad y Factibilidad Económica Financiera
- Diseño de Sistemas de Gestión de Calidad
- Estudios Ambientales

La empresa Alex Stewart (Assayers) registra, organiza y recopila toda la información que el cliente desee obtener al momento de solicitar el análisis a las muestras de mineral, este proceso de análisis consiste en hacer cobros por análisis de calidad, pesaje, muestreo y/o preparación de la muestra. Estos datos son guardados de manera semi-manual, por el hecho de guardar la información en hojas de cálculo en Excel y poseer una gran cantidad de información sobre los trabajos realizados, en folders de palanca. Es por esta razón que la búsqueda de información a cerca de cualquier solicitud o factura emitida por los análisis, que un cliente ha solicitado se hace morosa e imprecisa. Analizando tal contexto y situación, oportunamente se vio la necesidad de efectivizar este proceso y desarrollar este sistema, haciendo más sencilla la búsqueda de información.

1.2.2 ANTECEDENTES DEL PROYECTO

La empresa Alex Stewart (Assayers) no cuenta con ningún sistema automatizado que les permita registrar, controlar y gestionar toda la información, es por esta razón que desarrollaremos un sistema que nos brinde toda la información que se requiera al momento de conocer los resultados de los análisis a las muestras de mineral y conocer el funcionamiento de la empresa.

El sistema registra todos los datos del material, que ha sido enviado por un determinado cliente, se hace el llenado de un formulario con los datos del cliente, muestra del mineral como material y el tipo de análisis que el cliente desee realizar: (pesaje, muestreo, análisis de la calidad y/o preparación de la muestra) en esta actividad se debe considerar la validación y verificación de los datos que se están ingresando a la base de datos.

A su vez se ingresan los datos en almacenes para tener un control de cada material que será analizado en laboratorio, una vez analizados se procede a emitir una factura con todos los datos de la solicitud para que el cliente pueda hacer su debida cancelación. También se tienen almacenados los datos del cliente y el control de personal autorizado al ingreso del sistema, para contar con mayor información dentro de la empresa.

Los resultados de un sistema de información deben de ser precisos, para ello se debe suministrar también información precisa, consistente y coherente con las reglas semánticas propias del mundo real al que ha de representar lo más fielmente posible².

1.3 PLANTEAMIENTO DEL PROBLEMA

Los problemas que se identificaron al momento de realizar el sistema, son los siguientes:

² [Adoración de Miguel, 1999]

- Dificultad al momento del llenado de información, dándose el caso de registrar por algún error dos veces un mismo análisis al mismo cliente.
- Tardanza en el registro y organización de la información en almacenes.
- Falta de información al momento en que el cliente solicita detalles del análisis a las muestras de mineral que se realizo.
- Control del personal no autorizado al sistema.
- Datos no actualizados, cuando el cliente solicita la verificación si ha sido o no cancelado el trabajo realizado en los laboratorios.

1.3.1 FORMULACIÓN DEL PROBLEMA

¿De qué manera se puede realizar un control y seguimiento a análisis de muestras de minerales, cuyas muestras de minerales son enviadas por un cliente para su respectivo análisis y estos resultados a su vez sean registrados en el sistema, para un mayor control de información sobre los análisis que realiza la empresa a un determinado cliente, y así emitir una factura con el detalle y cobro del trabajo que se realizo?

Como obtener información sobre el tipo de análisis que el cliente solicito en los laboratorios de la empresa y obtener los datos almacenados sobre cada muestra de mineral que ingresa en el área de almacenes.

1.4 FORMULACIÓN DEL OBJETIVO

1.4.1 OBJETIVO GENERAL

Implementar un sistema de control y seguimiento a análisis de muestras de mineral para la empresa Alex Stewart Assayers que permita obtener resultados de los análisis realizados en los laboratorios de la empresa y organizar la información

que la empresa adquiere cuando el cliente solicita el tipo de análisis a una determinada muestra de mineral.

1.4.2 OBJETIVOS ESPECÍFICOS

- Registrar los datos de un determinado cliente para obtener más información.
- Registrar las muestras de mineral que ingresan a la empresa para luego ser enviadas a los laboratorios.
- Tener una base de datos donde estén almacenados todos los datos de las muestras de mineral, en caso de requerir información sobre ellos o verificar si han sido o no enviadas a su respectivo análisis
- Manejar datos confiables y seguros al momento de realizar un registro.
- Imprimir informes con la lista de todas las muestras de mineral que ingresaron en almacén.
- Elaborar reportes.

1.5 JUSTIFICACIÓN

El desarrollo del proyecto permite adquirir información detallada y actualizada para el control y seguimiento de las muestras de mineral por cada análisis que realiza la empresa Alex Stewart (Assayers), de esta manera el cliente trabaja con la empresa de manera confiable.

1.5.1 JUSTIFICACIÓN ECONÓMICA

Se justifica económicamente desde el punto de vista en la inversión de material de escritorio para el registro de cada material ingresado en almacenes a través del mismo se minimizara el tiempo de registro de las facturas emitidas por la empresa, mismos que se optimizaran a través del sistema.

Beneficios, a esto se suma el mayor rendimiento, mejor control del seguimiento de manejo de información procesado por la empresa.

1.5.2 JUSTIFICACIÓN SOCIAL

A la conclusión y obtención del sistema, la información generada por ésta servirá para informar de los avances tanto de la empresa, como en la ejecución de los programas a la sociedad, para poner en conocimiento de las tareas ejecutadas por la empresa Alex Stewart (Assayers) enmarcado en la gestión sustentable de las tareas que realiza la misma. Es muy importante que la sociedad así como entidades externas que coadyuvan con la empresa, las cuales puedan evaluar el trabajo de las entidades en este marco; obteniendo los resultados de acuerdo a los planes trazados hasta las fechas previstas para su conclusión.

Al mismo tiempo el sistema está orientado a un conjunto de personas, a través de una estrategia de inversión que produzca por sobretodo ahorros, tomando como prioridad la velocidad y capacidad de absorber y/o adaptar avances tecnológicos en la empresa.

1.5.3 JUSTIFICACIÓN TECNOLÓGICA

Se justifica técnicamente por que en el presente proyecto se aplican diferentes técnicas para el control y seguimiento de la empresa, además que se emplearan técnicas de modelaje y diseño de sistemas. Para ello es necesario la investigación y la evaluación de las nuevas tecnologías dentro de esta área, las cuales nos pueden ser muy útiles y podemos incluir al desarrollo y mejorar la complejidad de nuestro sistema.

Además este proyecto motivará a los clientes a usar y aprovechar la tecnología existente en la empresa con el que actualmente cuenta ya que estos son necesarios para que el presente proyecto se pueda poner en marcha. Por otra parte se aplicaran otras técnicas, que las iremos mencionando más adelante.

1.6 ALCANCES, LÍMITES Y APORTES

1.6.1 ALCANCES

Los alcances una vez obtenido el sistema para la empresa Alex Stewart (Assayers) son:

- Registro de todos los datos obtenidos de cada cliente.
- Desarrollar un sistema orientado al diseño de formularios para el registro de datos, en función a los requerimientos del cliente.
- Control del personal no autorizado al sistema.
- Control de todas las muestras de mineral ingresados en almacenes.
- Registro de Usuario. Se realizará un registro del usuario que esta accediendo al sistema, registrando todos los datos necesarios.
- Registro de un informe detallado del tipo de análisis que se realizo en los laboratorios.
- Generación de reportes del estado de avance en el cumplimiento de las diferentes metas.
- Se brindara un manual de usuario que le permitirá al operador comprender el manejo del sistema.

1.6.2 LÍMITES

Para el desarrollo del sistema se orientará a procesos realizados dentro de la parte de registro, seguimiento y control a las muestras de mineral. Nos limitamos a cumplir con los siguientes puntos:

- No se realizara un estudio exhaustivo de la composición de cada mineral, se trabajara a través de datos proporcionados por los laboratorios de la empresa.

- El desarrollo de este proyecto servirá como una herramienta a los especialistas e instituciones que se encuentran en esta área.
- El establecimiento de una arquitectura para el diseño de formularios a partir de los requerimientos del usuario.
- Realizar un seguimiento a los datos proporcionados por los laboratorios de la empresa.

1.6.3 APORTES

Los aportes del presente trabajo se detallan a continuación:

- El principal será la implementación del Sistema de Control y Seguimiento a análisis de muestras de minerales, para los funcionarios de la empresa “Alex Stewart Assayers”, esto con el fin de ayudar a la empresa para un mejor control.
- Permitir a los funcionarios llevar a cabo un control de las muestras de mineral que el cliente ha solicitado.
- También se hará uso de una interfaz del usuario con el sistema, brindándoles información referida al manejo del sistema, llenado de formularios, registros, reportes y toda la información que solicite el cliente al conocer los resultados del análisis a las muestras de mineral.
- En la parte de la evaluación de calidad de software se hará uso de la norma ISO 9126, esto con el fin de tener medidas reales y prácticas de la calidad del sistema.

El sistema se desarrollará mediante un conjunto de controles de usuario, esta es una técnica que brinda el Visual Studio .NET, esta nueva tecnología cuenta con un grupo de controles más robustos y la posibilidad de agruparlos mediante la utilización de los controles de usuario. En estos controles podemos utilizar controles conocidos como el TextBox, GroupBox, ComboBox, etc entre otros. Y estas a su vez pueden ser reutilizadas.



Capítulo II

Marco Teórico

2.1 SISTEMAS DE INFORMACIÓN

Sistema de información (SI) es un conjunto de elementos orientados al tratamiento y administración de datos e información, organizados y listos para su posterior uso, generados para cubrir una necesidad (objetivo). Dichos elementos formarán parte de alguna de estas categorías:

- Personas
- Datos
- Actividades o técnicas de trabajo.

Recursos materiales en general (típicamente recursos informáticos y de comunicación, aunque no tienen por qué ser de este tipo obligatoriamente).

Todos estos elementos interactúan entre sí para procesar los datos (incluyendo procesos manuales y automáticos) dando lugar a información más elaborada y distribuyéndola de la manera más adecuada posible en una determinada organización en función de sus objetivos.

El término Sistemas de Información hace referencia a un concepto genérico que tiene diferentes significados según el campo del conocimiento al que se aplique dicho concepto, a continuación se enumeran algunos de dichos campos y el sentido concreto que un Sistema de Información tiene en ese campo:

En informática, un sistema de información es cualquier sistema o subsistema de equipo de telecomunicaciones o computacional interconectados y que se utilicen para obtener, almacenar, manipular, administrar, mover, controlar, desplegar, intercambiar, transmitir o recibir voz y/o datos, e incluye tanto los programas de computación ("software" y "firmware") como el equipo de cómputo.

En teoría de sistemas, un sistema de información es un sistema, automatizado o manual, que abarca personas, máquinas, y/o métodos organizados de recolección de datos, procesamiento, transmisión y diseminación de datos que representa información para el usuario.

En seguridad computacional, un sistema de información está descrito por tres componentes:

➤ Estructura:

- Repositorios, que almacenan los datos permanente o temporalmente, tales como "buffers", RAM (memoria de acceso aleatorio), discos duros, caché, etc.
- Interfaces, que permiten el intercambio de información con el mundo no digital, tales como teclados, altavoces, monitores, escáneres, impresoras, etc.

➤ Canales, que conectan los repositorios entre sí, tales como, cables, enlaces inalámbricos, etc. Una red de trabajo es un conjunto de canales físicos y lógicos.

➤ Comportamiento:

- Servicios, los cuales proveen algún valor a los usuarios o a otros servicios mediante el intercambio de mensajes.
- Mensajes, que acarrear un contenido o significado hacia los usuarios internos o servicios.

2.1.1 ACTIVIDADES QUE REALIZA UN SISTEMA DE INFORMACIÓN

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información.

Entrada de Información: Es el proceso mediante el cual el Sistema de Información toma los datos que requiere para procesar la información. Las entradas pueden ser manuales o automáticas. Las manuales son aquellas que se proporcionan en forma directa por el usuario, mientras que las automáticas son datos

o información que provienen o son tomados de otros sistemas o módulos. Esto último se denomina interfaces automáticas.

Las unidades típicas de entrada de datos a las computadoras son las terminales, las cintas magnéticas, las unidades de diskette, los códigos de barras, los escáner, la voz, los monitores sensibles al tacto, el teclado y el mouse, entre otras.

Almacenamiento de información: El almacenamiento es una de las actividades o capacidades más importantes que tiene una computadora, ya que a través de esta propiedad el sistema puede recordar la información guardada en la sección o proceso anterior. Esta información suele ser almacenada en estructuras de información denominadas archivos. La unidad típica de almacenamiento son los discos magnéticos o discos duros, los discos flexibles o diskettes y los discos compactos (CD-ROM).

Procesamiento de Información: Es la capacidad del Sistema de Información para efectuar cálculos de acuerdo con una secuencia de operaciones preestablecidas. Estos cálculos pueden efectuarse con datos introducidos recientemente en el sistema o bien con datos que están almacenados. Esta característica de los sistemas permite la transformación de datos fuente en información que puede ser utilizada para la toma de decisiones, lo que hace posible, entre otras cosas, que un tomador de decisiones genere una proyección financiera a partir de los datos que contiene un estado de resultados o un balance general de un año base.

Salida de Información: La salida es la capacidad de un Sistema de Información para sacar la información procesada o bien datos de entrada al exterior. Las unidades típicas de salida son las impresoras, terminales, diskettes, cintas magnéticas, la voz, los graficadores y los plotters, entre otros. Es importante aclarar que la salida de un Sistema de Información puede constituir la entrada a otro Sistema de Información o módulo.

A continuación se muestran las diferentes actividades que puede realizar un Sistema de Información de Control de Clientes:

Entradas:

- Datos generales del cliente: nombre, dirección, tipo de cliente.
- Políticas de créditos: límite de crédito, plazo de pago, etc.
- Facturas (interface automático).
- Pagos, depuraciones, etc.

Proceso:

- Cálculo de antigüedad de saldos.
- Cálculo de intereses moratorios.
- Cálculo del saldo de un cliente.

Almacenamiento:

- Movimientos del mes (pagos, depuraciones).
- Catálogo de clientes.
- Facturas.

Salidas:

- Reporte de pagos.
- Estados de cuenta.
- Pólizas contables (interface automática)
- Consultas de saldos en pantalla de una terminal.

Las diferentes actividades que realiza un Sistema de Información se pueden observar en la Figura 2.1. Los Sistemas de Información cumplirán tres objetivos básicos dentro de las organizaciones:

- Automatización de procesos operativos.
- Proporcionar información que sirva de apoyo al proceso de toma de decisiones.

- Lograr ventajas competitivas a través de su implantación y uso.

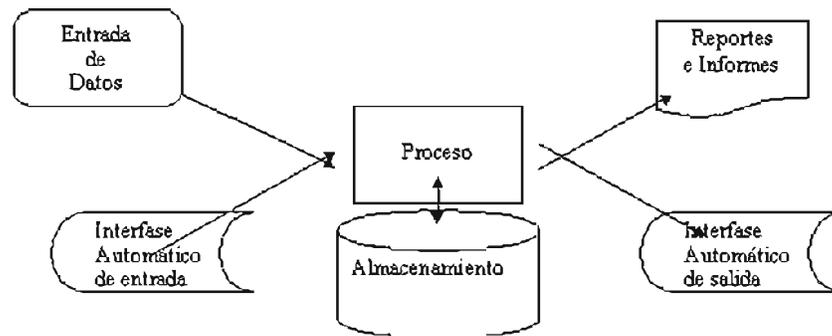


Figura 2.1 Título: Actividades de un sistema de información.

[Fuente: <http://www.monografias.com/trabajos7/sisinf/>]

Esta figura muestra como es el funcionamiento de los sistemas de información.

2.1.2 TIPOS DE SISTEMAS DE INFORMACIÓN

Los Sistemas de Información que logran la automatización de procesos operativos dentro de una organización, son llamados frecuentemente Sistemas Transaccionales, ya que su función primordial consiste en procesar transacciones tales como pagos, cobros, pólizas, entradas, salidas, etc. Por otra parte, los Sistemas de Información que apoyan el proceso de toma de decisiones son los Sistemas de Soporte a la Toma de Decisiones, Sistemas para la Toma de Decisión de Grupo, Sistemas Expertos de Soporte a la Toma de Decisiones y Sistema de Información para Ejecutivos. El tercer tipo de sistema, de acuerdo con su uso u objetivos que cumplen, es el de los Sistemas Estratégicos, los cuales se desarrollan en las organizaciones con el fin de lograr ventajas competitivas, a través del uso de la tecnología de información. Los tipos y usos de los Sistemas de Información se muestran en la figura 2.2.

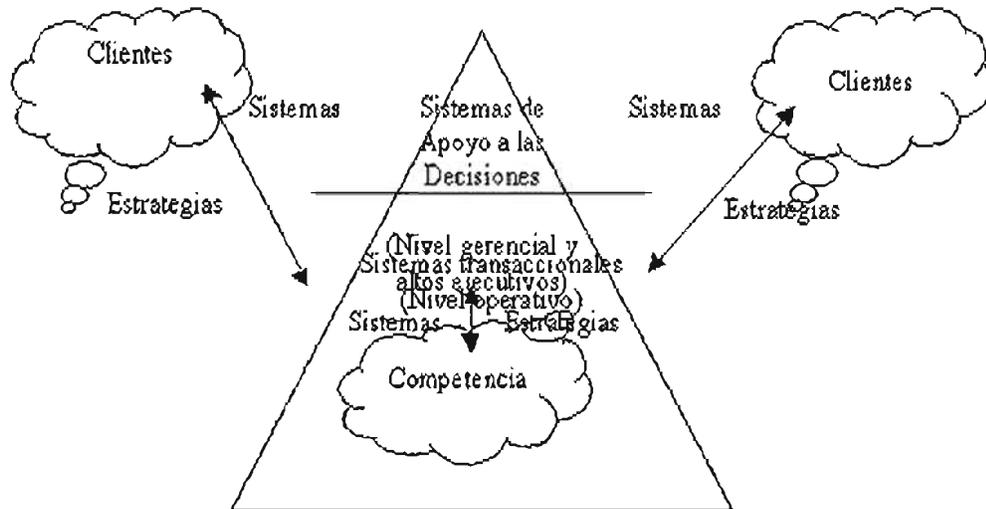


Figura 2.2 Titulo: Tipos de sistemas de información.

Fuente: [<http://www.monografias.com/trabajos7/sisinf>]

A continuación se mencionan las principales características de estos tipos de Sistemas de Información.

Sistemas Transaccionales: Sus principales características son:

- A través de éstos suelen lograrse ahorros significativos de mano de obra, debido a que automatizan tareas operativas de la organización.
- Con frecuencia son el primer tipo de Sistemas de Información que se implanta en las organizaciones. Se empieza apoyando las tareas a nivel operativo de la organización.
- Son intensivos en entrada y salida de información; sus cálculos y procesos suelen ser simples y poco sofisticados.
- Tienen la propiedad de ser recolectores de información, es decir, a través de estos sistemas se cargan las grandes bases de información para su explotación posterior.
- Son fáciles de justificar ante la dirección general, ya que sus beneficios son visibles y palpables.

Sistemas de Apoyo de las Decisiones: Las principales características de estos son:

- Suelen introducirse después de haber implantado los Sistemas Transaccionales más relevantes de la empresa, ya que estos últimos constituyen su plataforma de información.
- La información que generan sirve de apoyo a los mandos intermedios y a la alta administración en el proceso de toma de decisiones.
- Suelen ser intensivos en cálculos y escasos en entradas y salidas de información. Así, por ejemplo, un modelo de planeación financiera requiere poca información de entrada, genera poca información como resultado, pero puede realizar muchos cálculos durante su proceso.
- No suelen ahorrar mano de obra. Debido a ello, la justificación económica para el desarrollo de estos sistemas es difícil, ya que no se conocen los ingresos del proyecto de inversión.
- Suelen ser Sistemas de Información interactivos y amigables, con altos estándares de diseño gráfico y visual, ya que están dirigidos al usuario final.
- Apoyan la toma de decisiones que, por su misma naturaleza son repetitivos y de decisiones no estructuradas que no suelen repetirse. Por ejemplo, un Sistema de Compra de Materiales que indique cuándo debe hacerse un pedido al proveedor o un Sistema de Simulación de Negocios que apoye la decisión de introducir un nuevo producto al mercado.

- Estos sistemas pueden ser desarrollados directamente por el usuario final sin la participación operativa de los analistas y programadores del área de informática.

Este tipo de sistemas puede incluir la programación de la producción, compra de materiales, flujo de fondos, proyecciones financieras, modelos de simulación de negocios, modelos de inventarios, etc.

Sistemas Estratégicos: Sus principales características son:

- Su función primordial no es apoyar la automatización de procesos operativos ni proporcionar información para apoyar la toma de decisiones.
- Suelen desarrollarse en casa es decir, dentro de la organización, por lo tanto no pueden adaptarse fácilmente a paquetes disponibles en el mercado.
- Típicamente su forma de desarrollo es a base de incrementos y a través de su evolución dentro de la organización. Se inicia con un proceso o función en particular y a partir de ahí se van agregando nuevas funciones o procesos.
- Su función es lograr ventajas que los competidores no posean, tales como ventajas en costos y servicios diferenciados con clientes y proveedores. En este contexto, los Sistema Estratégicos son creadores de barreras de entrada al negocio. Por ejemplo, el uso de cajeros automáticos en los bancos en un Sistema Estratégico, ya que brinda ventaja sobre un banco que no posee tal servicio. Si un banco nuevo decide abrir sus puertas al público, tendrá que dar este servicio para tener un nivel similar al de sus competidores.

- Apoyan el proceso de innovación de productos y proceso dentro de la empresa debido a que buscan ventajas respecto a los competidores y una forma de hacerlo en innovando o creando productos y procesos.

Por último, es importante aclarar que algunos autores consideran un cuarto tipo de sistemas de información denominado Sistemas Personales de Información, el cual está enfocado a incrementar la productividad de sus usuarios.

2.1.3 PARÁMETROS DE LOS SISTEMAS DE INFORMACIÓN

El sistema se caracteriza por ciertos parámetros los cuales son constantes arbitrarias que singularizan, por sus propiedades, el valor y la descripción dimensional de un sistema específico o de un componente del sistema.

Los parámetros de los sistemas son:

- Entrada o impulso (input): Es la fuerza de arranque del sistema, que provee el material o la energía para la operación del sistema.
- Salida o resultado (output): Es la finalidad para la cual se reunieron elementos y relaciones del sistema. Los resultados de un proceso son las salidas, las cuales deben ser coherentes con el objetivo del sistema. Los resultados de los sistemas son finales, mientras que los resultados de los subsistemas son intermedios.
- Procesamiento o procesador (throughput): Es el fenómeno que produce cambios, también es el mecanismo de conversión de las entradas en salidas o resultados.

Generalmente es representado como la caja negra, en la que entran los insumos y salen cosas diferentes, que son los productos.

- Retroalimentación o reinformación (feedback): Es la función de retorno del sistema que tiende a comparar la salida con un criterio preestablecido, manteniéndola controlada dentro de aquel estándar o criterio.
- Ambiente: Es el medio que envuelve externamente el sistema. Está en constante interacción con este último que es el que recibe entradas, las procesa y efectúa salidas. La supervivencia de un sistema depende de su capacidad de adaptarse, cambiar y responder a las exigencias y demandas del ambiente externo. Aunque el ambiente puede ser un recurso para el sistema, también puede ser una amenaza.

2.1.4. Aplicación de los sistemas de información

Los sistemas de información tratan el desarrollo, uso y administración de la infraestructura de la tecnología de la información en una organización.

En la era post-industrial, la era de la información, el enfoque de las compañías ha cambiado de la orientación hacia el producto a la orientación hacia el conocimiento, en este sentido el mercado compite hoy en día en términos del proceso y la innovación, en lugar del producto. El énfasis ha cambiado de la calidad y cantidad de producción hacia el proceso de producción en sí mismo, y los servicios que acompañan este proceso.

El mayor de los activos de una compañía hoy en día es su información, representada en su personal, experiencia, conocimiento, innovaciones (patentes, derechos de autor, secreto comercial). Para poder competir, las organizaciones deben poseer una fuerte infraestructura de información, en cuyo corazón se sitúa la infraestructura de la tecnología de información. De tal manera que el sistema de información se centre en estudiar las formas para mejorar el uso de la tecnología que soporta el flujo de información dentro de la organización.

2.1.5. EVOLUCIÓN DE LOS SISTEMAS DE INFORMACIÓN

De la sección anterior se desprende la evolución que tienen los Sistemas de Información en las organizaciones. Con frecuencia se implantan en forma inicial los Sistemas Transaccionales y, posteriormente, se introducen los Sistemas de Apoyo a las Decisiones. Por último, se desarrollan los Sistemas Estratégicos que dan forma a la estructura competitiva de la empresa.

En la década de los setenta, Richard Nolan, un conocido autor y profesor de la Escuela de Negocios de Harvard, desarrolló una teoría que impactó el proceso de planeación de los recursos y las actividades de la informática.

2.2 METODOLOGÍA

Para el desarrollo del presente proyecto se aplicará la metodología RUP y para su modelado el UML, herramientas que contribuyan al mejor entendimiento y desarrollo del mismo. La metodología utilizada para este trabajo está sujeta a la modelización y abstracción de datos, adquiriendo información necesaria, segura y confiable que el cliente requiera y para el funcionamiento de la empresa, para este efecto se realiza las siguientes fases genéricas:

- Selección de un conjunto de datos.
- Combinación de datos en una estructura.
- Refinación de la estructura hasta que la misma refleje la situación ideal deseada.

En el análisis de la situación actual y análisis de la situación futura deseada como marco introductorio del cumplimiento del avance en la construcción del sistema son guiados en una primera etapa por las especificaciones realizadas dentro del sistema de marco aplicativo que ayuda a elaborar un proyecto.

2.2.1 METODOLOGÍA RUP

El Proceso Unificado de Desarrollo Software o simplemente Proceso Unificado es un marco de desarrollo de software que se caracteriza por estar dirigido por casos

de uso, centrado en la arquitectura y por ser iterativo e incremental. El refinamiento más conocido y documentado del Proceso Unificado es el Proceso Unificado de Rational o simplemente RUP.³

El Proceso Unificado no es simplemente un proceso, sino un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. De la misma forma, el Proceso Unificado de Rational, también es un marco de trabajo extensible, por lo que muchas veces resulta imposible decir si un refinamiento particular del proceso ha sido derivado del Proceso Unificado o del RUP. Por dicho motivo, los dos nombres suelen utilizarse para referirse a un mismo concepto.

El nombre Proceso Unificado se usa para describir el proceso genérico que incluye aquellos elementos que son comunes a la mayoría de los refinamientos existentes. También permite evitar problemas legales ya que Proceso Unificado de Rational o RUP son marcas registradas por IBM (desde su compra de Rational Software Corporation en 2003).

El primer libro sobre el tema se denominó, en su versión española, El Proceso Unificado de Desarrollo de Software (ISBN 84-7829-036-2) y fue publicado en 1999 por Ivar Jacobson, Grady Booch y James Rumbaugh, conocidos también por ser los desarrolladores del UML, el Lenguaje Unificado de Modelado. Desde entonces los autores que publican libros sobre el tema y que no están afiliados a Rational utilizan el término Proceso Unificado, mientras que los autores que pertenecen a Rational favorecen el nombre de Proceso Unificado de Rational.

El RUP mejora la productividad del equipo ya que permite que cada miembro del grupo sin importar su responsabilidad específica acceda a la misma base de datos de conocimiento. Esto hace que todos compartan el mismo lenguaje, la misma visión y el mismo proceso acerca de cómo desarrollar software.

³ G. Booch, El Lenguaje Unificado de Modelado, AddisonWesley, 1999.

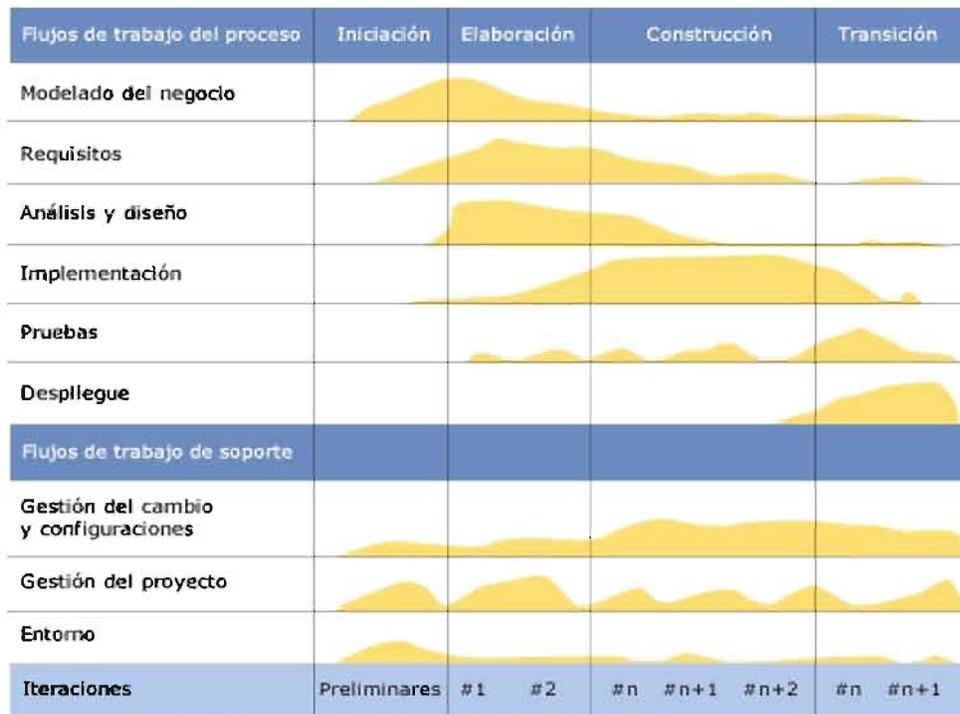


Figura 2.3 Título: Ciclo de vida del RUP

[Fuente: Metodología RUP]

Fases del RUP

- Establece oportunidad y alcance
- Identifica las entidades externas o actores con las que se trata
- Identifica los casos de uso

RUP comprende 2 aspectos importantes por los cuales se establecen las disciplinas:

Proceso: Las etapas de esta sección son:

- Modelado de negocio
- Requisitos
- Análisis y Diseño
- Implementación

- Pruebas
- Despliegue

Soporte: En esta parte nos encontramos con las siguientes etapas:

- Gestión del cambio y configuraciones
- Gestión del proyecto
- Entorno

La estructura dinámica de RUP es la que permite que éste sea un proceso de desarrollo fundamentalmente iterativo, y en esta parte se ven inmersas las 4 fases descritas anteriormente:

- Inicio (También llamado Incepción o Concepción)
- Elaboración
- Desarrollo (También llamado Implementación, Construcción)
- Cierre (También llamado Transición)

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una base de inicio

Fase de Inicio: Esta fase tiene como propósito definir y acordar el alcance del proyecto con los patrocinadores, identificar los riesgos asociados al proyecto, proponer una visión muy general de la arquitectura de software y producir el plan de las fases y el de iteraciones posteriores. Durante esta fase de inicio las iteraciones se centran con mayor énfasis en las actividades de modelado del sistema y en sus requerimientos

Fase de elaboración: En la fase de elaboración se seleccionan los casos de uso que permiten definir la arquitectura base del sistema y se desarrollaran en esta fase, se realiza la especificación de los casos de uso seleccionados y el primer análisis del dominio del problema, se diseña la solución preliminar.

Fase de Construcción: El propósito de esta fase es completar la funcionalidad del sistema, para ello se deben clarificar los requisitos pendientes, administrar los cambios de acuerdo a las evaluaciones realizados por los usuarios y se realizan las mejoras para el proyecto. Además se lleva a cabo la construcción del producto por medio de una serie de iteraciones las cuales se seleccionan algunos Casos de Uso, se redefine su análisis y diseño y se procede a su implantación y pruebas. En esta fase se realiza una pequeña cascada para cada ciclo, se realizan tantas iteraciones hasta que se termine la nueva implementación del producto.

Fase de Transición: El propósito de esta fase es asegurar que el software esté disponible para los usuarios finales, ajustar los errores y defectos encontrados en las pruebas de aceptación, capacitar a los usuarios y proveer el soporte técnico necesario. Se debe verificar que el producto cumpla con las especificaciones entregadas por las personas involucradas en el proyecto.

Flujos de trabajo: La mera enumeración de roles, actividades y artefactos no define un proceso, necesitamos definir la secuencia de actividades realizadas por los diferentes roles, así como la relación entre los mismos, que nos producen unos resultados observables. El RUP define varios flujos de trabajo distintos, entre los que distingue entre dos grupos, los de proceso, y los de apoyo. Las distintas iteraciones a realizar consistirá en la ejecución de estos flujos de trabajo con una mayor o menos intensidad dependiendo de la fase e iteración en la que nos encontremos.

Especificación de las Fases

RUP en cada una de sus fases (pertenecientes a la estructura estática) realiza una serie de artefactos que sirven para comprender mejor tanto el análisis como el diseño del sistema estos artefactos son los siguientes:

Inicio:

- Documento Visión
- Especificación de Requerimientos

Elaboración:

- Diagramas de caso de uso

Construcción:

- Documento Arquitectura que trabaja con las siguientes vistas:

Vista Lógica:

- Diagrama de clases
- Modelo E-R (Si el sistema así lo requiere)

Vista de Implementación:

- Diagrama de Secuencia
- Diagrama de estados
- Diagrama de Colaboración

Vista Conceptual:

- Modelo de dominio

Vista física:

- Mapa de comportamiento a nivel de hardware.

La metodología RUP es más apropiada para proyectos grandes (Aunque también pequeños), dado que requiere un equipo de trabajo capaz de administrar un proceso complejo en varias etapas. En proyectos pequeños, es posible que no se puedan cubrir los costos de dedicación del equipo de profesionales necesarios.

2.3 MODELADO UML

Lenguaje Unificado de Modelado (LUM) o (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software

más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group).⁴ Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.⁵

El lenguaje UML se expresa con símbolos y/o agrupaciones de estos llamadas diagramas. Nos sirve fundamentalmente para crear diferentes tipos de ellos permitiéndonos ver desde diferentes perspectivas un sistema software.

⁴ Estándar Federal 1037C, Sistemas de Información Glosario Nacional.

⁵ En informática es un área de memoria destinada para el almacenamiento temporal de datos, y se utilizan típicamente entre dispositivos con diferente velocidad de procesamiento.

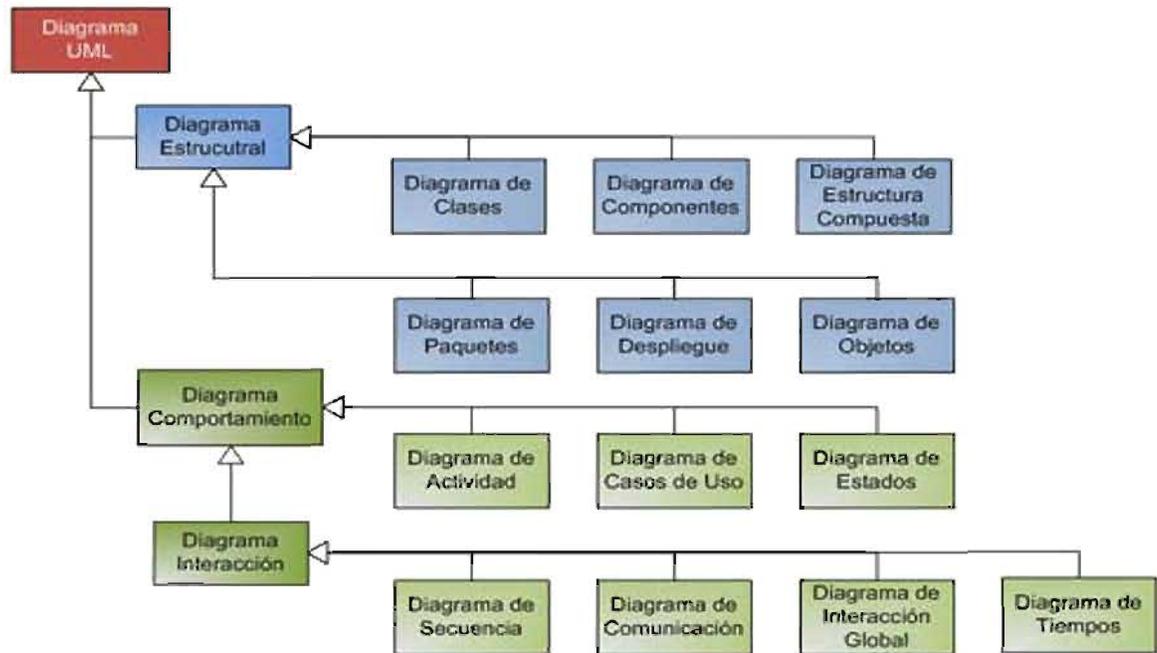


Figura 2.4 Título: Diagrama UML

[Fuente: <http://www.monografias.com/trabajos5/insof/insof.shtml>]

Diagramas de estructura: Muestran los elementos de una especificación que sean independientes del tiempo. Incluyen clase, estructura de componentes, componente, despliegue, objeto y diagramas de paquetes.

Diagramas de comportamiento: Permiten exhibir comportamientos de un sistema o de los procesos de las organizaciones. Incluyen los diagramas de actividad, estado, caso típico y de interacción.

Diagramas de interacción: Es un subconjunto de los diagramas de comportamiento que permiten enfatizar las interacciones entre los objetos. Incluyen comunicación, vista general de interacciones, secuencia y diagrama de tiempo.

En el estándar UML se nos habla de un grupo de diagramas determinado que son los más comunes y habituales. Podemos establecer una clasificación de estos dividiéndolos en Estructurales y de Comportamiento, y dentro de los de comportamiento tendríamos a los de Interacción.

Existe un tipo primordial que es “Diagrama UML” del cual heredan “Diagrama Estructural” y “Diagrama Comportamiento”, de estos a su vez heredan los trece

diferentes tipos de diagramas más comunes, existiendo un tipo intermedio que serían los “Diagramas de Interacción”.

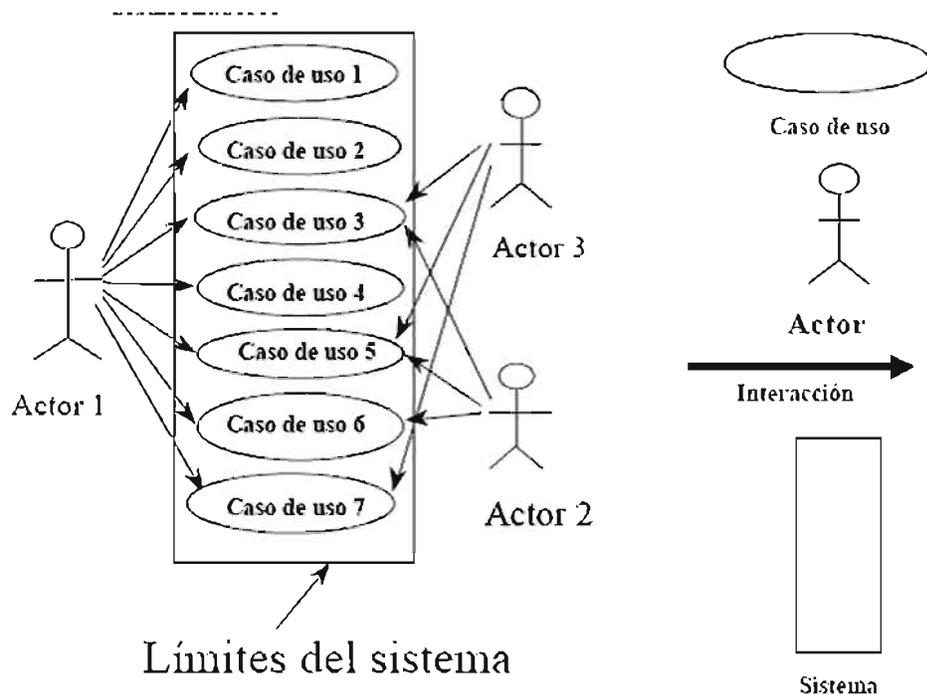


Figura 2.5 Título: Diagrama de casos de uso

[Fuente: <http://www.que-informatica.com/index.php/tag/uml/>]

Los diagramas son de gran utilidad para trabajar en los requisitos, en el análisis del sistema, en la construcción del mismo y en su posterior despliegue. Nos permitirán conocer ese concepto del que tanto se habla y que parece tan difícil de determinar que es la Arquitectura del Sistema. El UML hace que esta sea algo tangible. Siendo el resultado de agrupar los diferentes diagramas en lo que llamamos vistas. Estas vistas forman la Arquitectura del Sistema.

Cada una de ellas nos ofrece diferente información sobre el sistema software:

- Vista de Casos de Uso: Nos facilita información sobre el comportamiento y funcionalidad del sistema.
- Vista de Diseño: Nos proporciona información del vocabulario y la funcionalidad del sistema.

- Vista de Interacción: Nos da información sobre el rendimiento del sistema, la escalabilidad del mismo y la capacidad de procesamiento necesaria.
- Vista de Implementación: Establece el ensamblado del sistema y la gestión de la configuración.
- Vista de Despliegue: Nos permite establecer la topología del sistema, su distribución y las pautas para su instalación.



Figura 2.6 Título: Arquitectura del UML

[Fuente: <http://www.que-informatica.com/index.php/tag/uml/>]

Por lo tanto, este será el objetivo a la hora de crear software, conseguir plantear la arquitectura del sistema mediante diagramas UML. Conviene advertir aquí que el proceso de creación de software es un proceso iterativo e incremental. Así podemos apreciar que UML nos servirá para reflejar la arquitectura del sistema en un momento del tiempo dado. Lo normal será que en versiones o iteraciones posteriores del software que estamos creando esta arquitectura evolucione, quedando cada vez más definida. Lo que significará que nuestros diagramas irán cambiando, ya que son estos los que la definen. Además no es

obligatorio usar todos los diagramas, ni todos los símbolos. Es posible que el sistema que estamos desarrollando quede definido con solo algunos de ellos.

Por otro lado, comentar que el lenguaje UML, como tal lenguaje que es, se usa para definirse a sí mismo. Cada símbolo o concepto del UML y sus reglas de aplicación están definidos con UML en lo que se conoce como meta-modelo de UML.

2.4 TECNOLOGÍA Y HERRAMIENTAS DE DESARROLLO

2.4.1 VISUAL STUDIO.NET

Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, 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 intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles. Visual Studio 2008 fue publicado (RTM) el 17 de Noviembre de 2007 en inglés, mientras que la versión en castellano no fue publicada hasta el 2 de Febrero de 2008.

A las mejoras de desempeño, escalabilidad y seguridad con respecto a la versión anterior, se agregan entre otras, las siguientes novedades.

La mejora en las capacidades de Pruebas Unitarias permiten ejecutarlas más rápido independientemente de si lo hacen en el entorno IDE o desde la línea de comandos. Se incluye además un nuevo soporte para diagnosticar y optimizar el sistema a través de las herramientas de pruebas de Visual Studio. Con ellas se podrán ejecutar perfiles durante las pruebas para que ejecuten cargas, prueben

procedimientos contra un sistema y registren su comportamiento; y utilizar herramientas integradas para depurar y optimizar.

Visual C++ (también conocido como MSVC, Microsoft Visual C++) es un entorno de desarrollo integrado (IDE) para lenguajes de programación C, C++ y C++/CLI. Está especialmente diseñado para el desarrollo y depuración de código escrito para las API's de Microsoft Windows, DirectX y la tecnología Microsoft .NET Framework.

Visual C++ hace uso extensivo del framework Microsoft Foundation Classes (o simplemente MFC), el cual es un conjunto de clases C++ para el desarrollo de aplicaciones en Windows.

El IDE cuenta con herramientas poderosas como el IntelliSense, RemoteDebuging, Editar y Continuar, y Texto Resaltado.

Cuenta con una versión Express, llamada Microsoft Visual C++ Express Edition, la cual es gratuita y se puede descargar desde el sitio de Microsoft.

El lenguaje de programación utilizado por esta herramienta, de igual nombre, está basado en C++ y es compatible en la mayor parte de su código con este lenguaje, a la vez que su sintaxis es exactamente igual. En algunas ocasiones esta incompatibilidad impide que otros compiladores, sobre todo en otros sistemas operativos, funcionen bien con código desarrollado en este lenguaje.

2.4.2 MICROSOFT SQL SERVER

Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o PostgreSQL o MySQL.

Características de Microsoft SQL Server

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasa a ser el SQL Express Edition, que se distribuye en forma gratuita.

Es común desarrollar completos proyectos complementando Microsoft SQL Server y Microsoft Access a través de los llamados ADP (Access Data Project). De esta forma se completa la base de datos (Microsoft SQL Server), con el entorno de desarrollo (VBA Access), a través de la implementación de aplicaciones de dos capas mediante el uso de formularios Windows. En el manejo de SQL mediante líneas de comando se utiliza el SQLCMD

Para el desarrollo de aplicaciones más complejas (tres o más capas), Microsoft SQL Server incluye interfaces de acceso para varias plataformas de desarrollo, entre ellas .NET, pero el servidor sólo está disponible para Sistemas Operativos Windows.

En SQL Server 2008 la nueva base de datos contiene mayor seguridad, integración con PowerShell, remueve la configuración del área expuesta (consola para configurar seguridad), cifrado transparente de datos, auditoría de datos,

compresión de datos, tiene correctores de sintaxis del lenguaje Transact-SQL e IntelliSense (una característica del visual studio que permite a la base de datos sugerir objetos existentes mientras uno escribe la mitad de la palabra).

Así mismo incluye nuevos tipos de datos y funciones. Entre ellos, datos espaciales, nuevos datos de tiempo (datetime2 y Datetimeoffset), tipos de datos jerárquicos.

2.5 CALIDAD DE SOFTWARE

Es la aptitud de un producto o servicio para satisfacer las necesidades del usuario. Es la calidad de todos los productos, no solamente de equipos sino también de programas.

En el desarrollo de software, la calidad de diseño acompaña a la calidad de los requisitos, especificaciones y diseño del sistema. La calidad de concordancia es un aspecto centrado principalmente en la implementación; Si la implementación sigue al diseño, y el sistema resultante cumple con los objetivos de requisitos y de rendimiento, la calidad de concordancia es alta.

La calidad del software es una preocupación a la que se dedican muchos esfuerzos. Sin embargo, el software casi nunca es perfecto. Todo proyecto tiene como objetivo producir software de la mejor calidad posible, que cumpla, y si puede supere las expectativas de los usuarios.

Características propias del software son aquellas que se quiere controlar y asegurar, el software es un producto inmaterial que no se fabrica, tampoco se degradan físicamente, sino que se desarrolla; El software puede tener errores, incidencias pero no son similares a lo que cualquier equipo de carácter físico.

La calidad del software se encuentra a la par con la calidad tradicional, pero un paso atrás, debido a que la calidad tradicional tiene varias décadas de historia, mientras que la calidad de software tiene 50 a 30 años. Adicionalmente se puede seguir los siguientes aspectos para evaluar la calidad del software:

- Funcionalidad
- Confiabilidad
- Usabilidad
- Eficiencia
- Mantenibilidad
- Portabilidad
- Escalabilidad (actualización)

La calidad de concordancia es el grado de cumplimiento de las especificaciones de diseño durante su realización. En el desarrollo del software la calidad del diseño comprende los requisitos, especificaciones y el diseño del sistema, la calidad de concordancia es un aspecto centrado principalmente en la implementación. El control de calidad es una serie de inspecciones, revisiones y pruebas utilizados a lo largo del proceso del software para asegurar que cada producto cumple con los requisitos que le han sido asignados.

2.5.1 CERTIFICACIÓN DEL SOFTWARE

Consecuencia de un proceso que es asegurar la calidad pero nunca es el objetivo final. Los procedimientos para construir un software de calidad, deben ser correctos y estar en función de la normalización (ISO 9000).

NORMATIVA ISO 9000

Pone a disposición de un auditor o certificador los procesos internos, de forma que este indique si cumple o no la normativa al 100%, audita el sistema; Si los resultados son positivos se emite la certificación y cada cierto tiempo se tiene que renovar; La certificación es costosa, a consecuencia de costes que ocasionan la lejanía y el tiempo de duración de proceso (aprox. 6 meses). Se certifica la empresa y la metodología para el desarrollo de la aplicación.

2.5.2 MEDICIÓN DEL SOFTWARE

En el software lo que se mide son atributos propios del mismo, se descompone un atributo general en otros más simples de medir, a veces se mide bien o mal ya que la descomposición del atributo genérico de calidad en otros sub- atributos se torna irreal, se mide con datos estadísticos no avalados, es imposible decir que la medición se hace en forma correcta.

El concepto de medida va de más a menos, va de lo general a lo concreto y lo concreto es asociado a la métrica, cuya combinación te daría el nivel de calidad o seguridad de tu producto. Las ciencias bien estructuradas se basan en medidas bien hechas, se basan en la matemática.

2.5.2.1 TIPOS DE MEDIDA

- Número de errores durante un periodo determinado.
- Fallo en la codificación o diseño de un sistema que causa que el programa no funcione correctamente o falle.
 - Tamaño de un producto informático (líneas de código)
 - Métrica de punto función (IBM): relaciona funcionalidades que ofrecía.
 - Estimación de costes y esfuerzos.

COCOMO

El Modelo Constructivo de Costes (o COCOMO, por su acrónimo del inglés COConstructive COSt MOdel) es un modelo matemático de base empírica utilizado para estimación de costes¹ de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada

vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado.

Pertenece a la categoría de modelos de subestimaciones basados en estimaciones matemáticas. Está orientado a la magnitud del producto final, midiendo el "tamaño" del proyecto, en líneas de código principalmente.

Las ecuaciones de estimación del esfuerzo de desarrollo tienen la forma:

$$E = a_i S^{b_i} m(X)$$

con

- S el número de miles de líneas de código fuente.
- $m(X)$ es un multiplicador que depende de 15 atributos.

En la siguiente tabla se muestran los coeficientes para los diferentes modos.

Tabla 2.1

Proyecto de Software	a_b	b_b	c_b	d_b
Orgánico	2,4	1,05	2,5	0,38
Semi - acoplado	3,0	1,12	2,5	0,35
Empotrado	3,6	1,20	2,5	0,32

Título: Coeficientes del proyecto software

[Fuente: Pressman, 2002]

Modelo Básico

Este modelo trata de estimar, de una manera rápida y más o menos burda, la mayoría de proyectos pequeños y medianos. Se consideran tres modos de desarrollo en este modelo: orgánico, semi-acoplado y empotrado.

Modelo Intermedio

En este modelo se introducen 15 atributos de coste para tener en cuenta el entorno de trabajo. Estos atributos se utilizan para ajustar el coste nominal del proyecto al entorno real, incrementando la precisión de la estimación.

Modelo detallado

Este modelo puede procesar todas las características del proyecto para construir una estimación. Introduce dos características principales

- Multiplicadores de esfuerzo sensitivos a la fase: Algunas fases se ven más afectadas que otras por los atributos. El modelo detallado proporciona un conjunto de multiplicadores de esfuerzo para cada atributo. Esto ayuda a determinar la asignación del personal para cada fase del proyecto.
- Jerarquía del producto a tres niveles: Se definen tres niveles de producto. Estos son módulo, subsistema y sistema. La cuantificación se realiza al nivel apropiado, esto es, al nivel al que es más susceptible la variación.

2.5.3 UTILIDAD DE LA MEDIDA DEL SOFTWARE

En el proceso de asegurar la calidad, cuantificar los atributos que constituyen la calidad para el usuario final, ahí tenemos los resultados cuantitativos.

El usuario final mide la calidad del software según lo que tenga o no, es en ese sentido de que la calidad del software depende de quien la juzgue. El hecho de que una empresa tenga certificación en calidad de software no garantiza que su software sea de calidad igualdad al mismo nivel.

La medida del software es aplicada mayormente para favorecer a quien elabora el sistema y poder de este modo calcular costo de desarrollo del software.



Capítulo III

Marco Aplicativo

3.1 DESARROLLO DEL PROYECTO EMPLEANDO LA METODOLOGÍA RUP Y MODELADO UML

El primer paso hacia la división del proceso de desarrollo de software consiste en separar las partes en cuatro fases atendiendo el momento en que se realizan: inicio, elaboración, construcción y transición. Cada una de las fases se divide entonces en una o más iteraciones.

Luego de la investigación sobre la situación actual de la unidad de almacenes y de las unidades que requieren información de la misma, se realiza la puesta en práctica del método del proceso unificado para desarrollo de software. En ese sentido se planificó llevar a cabo 2 iteraciones, 1 en cada una de las fases, donde en cada fase se realizó las actividades de los cinco flujos de trabajo iterativos como son: modelo de negocio, requisitos, análisis, diseño, implementación, lo que llevo finalmente al desarrollo y construcción del sistema.

A continuación desarrollaremos las cuatro fases del Proceso Unificado abordando en la segunda iteración en el desarrollo del sistema a partir de las fases del Proceso Unificado.

UML presenta varios diagramas donde cada uno representa un aspecto del sistema. Es por eso que vamos a mencionar algunos de estos diagramas que emplearemos en el desarrollo de nuestro sistema. Haremos el uso de los diagramas necesarios para cada etapa según la complejidad de nuestro sistema.

No es necesario emplear todos los diagramas; para sistemas sencillos un diagrama de clases junto con un par de diagramas de actividades e interacción sería suficiente, asimismo si los sistemas son complejos requieren de la utilización de más diagramas, debido a que requieren de etapas incrementales e iterativas (ciclos de

desarrollo) en el análisis, diseño e implementación, por ello es que vamos a especificar la etapa de desarrollo y los diagramas necesarios en cada etapa.

3.2 FASE DE INICIO

La empresa “Alex Stewart Assayers Bolivia”, encargada del control y calidad de minerales cumple con la función principal de hacer un análisis a muestras de mineral.

Por la problemática que se tiene en el capítulo 1 y 2; en referencia a sus procesos de salida e ingresos de muestras de mineral a almacenes, se establece y se identifica la necesidad principal de contar con un software para la unidad de almacenes, que realice las tareas específicas de control y seguimiento. En este sentido se pretende llevar adelante el desarrollo del sistema propuesto.

3.2.1 MODELADO DEL NEGOCIO

El modelado del negocio se define como un proceso de representación de uno o más aspectos o elementos de una empresa, tales como su propósito, su estructura, su funcionalidad, su dinámica, su lógica de negocios y sus componentes.

Habiendo hecho un estudio preliminar al funcionamiento de la empresa “Alex Stewart Assayers Bolivia”, se pasa a identificar y describir cada uno de los procesos de negocio, determinando las informaciones, actividades, roles y reglas del negocio implicadas. Con esto lo que se pretende es comprender toda la actividad de la organización relacionada con el sistema a implantar.

El modelado del negocio se basa en dos diagramas importantes; el modelo de casos de uso del negocio y los modelos del objeto del negocio, que los describiremos a continuación:

3.2.1.1 MODELO DE CASOS DE USO DEL NEGOCIO

La empresa interactúa con distintos elementos externos, entre los que se identifican el cliente (persona o entidad que solicita el análisis a las muestras de mineral a los laboratorios de la empresa), el operador (persona que se encarga de

controlar el acceso al sistema), encargado de almacén (persona que se encarga de registrar el ingreso y salida de las muestras de mineral a ser analizadas y controlar las existencias en almacén) y por último encargado de laboratorio (persona que se encarga de la realización del análisis a la muestra de mineral y entrega de resultados).

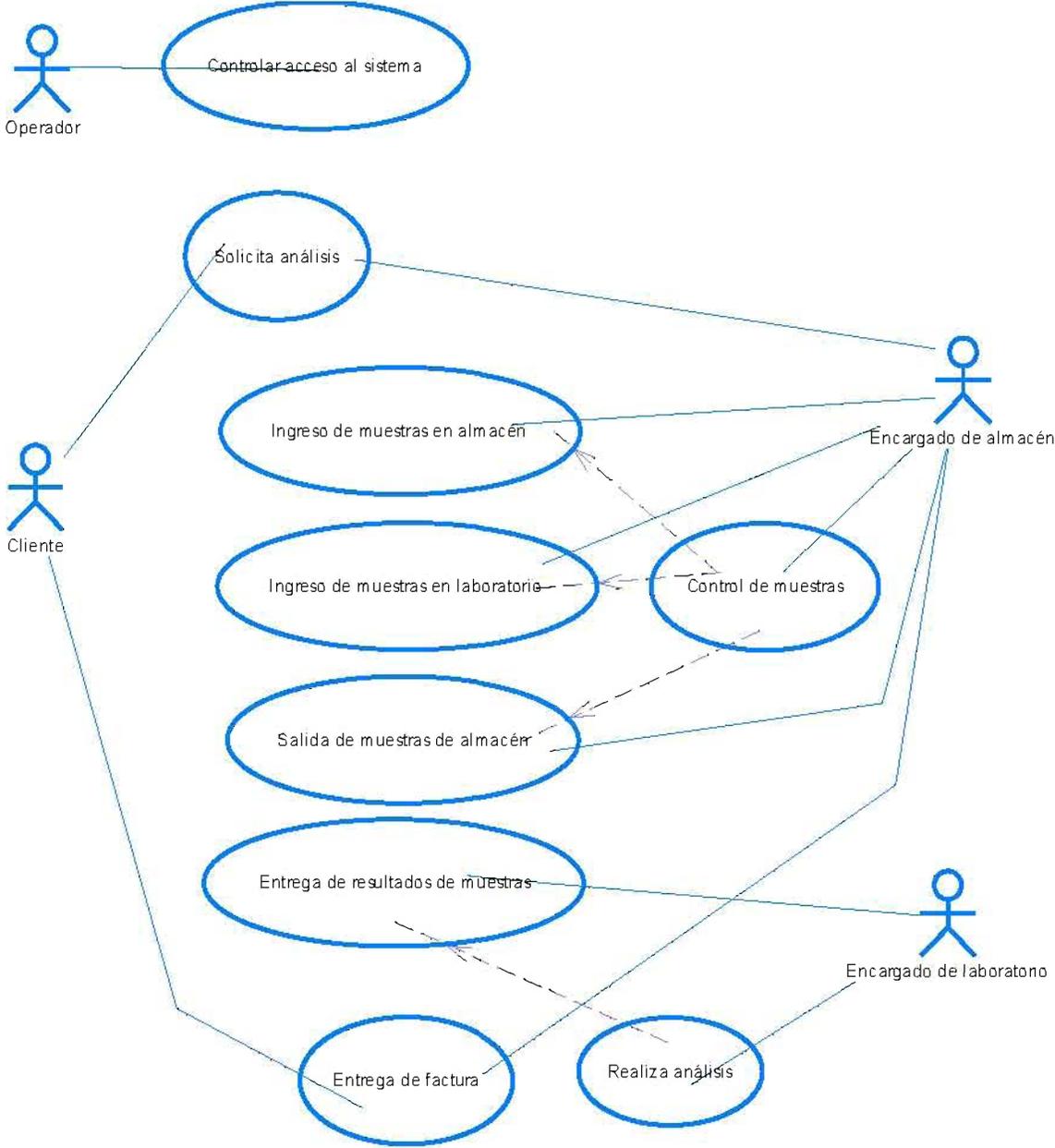


Figura 3.1 Título: Caso de uso del negocio.

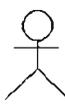
[Fuente: Elaboración Propia.]

En el diagrama de casos de uso del negocio elaborado anteriormente se observa como es el funcionamiento de la empresa. El cliente hace la solicitud de análisis a una determina muestra de mineral, el encargado del almacén realiza los registros de solicitud, ingresos de muestras en almacén, salidas de almacén, ingresos en laboratorio y elabora los respectivos informes. Mientras el encargado de laboratorio realiza los análisis a las muestras de mineral y entrega resultados. El clienta pasa la fecha que se le indico y se le hace la entrega de su factura con el detalle del análisis realizado.

a) DESCRIPCIÓN DE LOS ACTORES DEL MODELO DEL NEGOCIO

A continuación se muestra a los principales actores que se identificaron en el modelo de negocio y otros que son necesarios para el sistema propuesto.

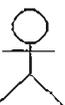
Tabla 3.1

Actor	Descripción
 Operador	Encargado de dar altas, bajas, modificaciones, búsquedas, listados y reportes, registros a los usuarios del sistema y verificar el acceso al sistema.

Título: Actor (Operador) y su descripción.

[Fuente: Elaboración Propia.]

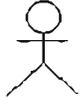
Tabla 3.2

Actor	Descripción
 Cliente	Hace la solicitud de análisis. Una vez realizado el análisis procede a cancelar por el trabajo que se realizo y recibe una factura por su pago.

Título: Actor (Cliente) y su descripción.

[Fuente: Elaboración Propia.]

Tabla 3.3

Actor	Descripción
 Encargado de almacén	Encargado de registrar, ingresos y salidas de nuevas muestras de mineral. Elabora los informes y facturas con los resultados del análisis. Encargado de llevar todo el control de los datos que se ingresan .

Titulo: Actor (Encargado de almacén) y su descripción.

[Fuente: Elaboración Propia.]

Tabla 3.4

Actor	Descripción
Encargado del laboratorio	Recibe las muestras de mineral que deben ser analizadas. Con la ayuda de otras personas, equipos y materiales necesarios, realiza el análisis que el cliente solicitó. Posteriormente pasa a elaborar un informe detallado del análisis solicitado.

Titulo: Actor (Encargado del laboratorio) y su descripción.

[Fuente: Elaboración Propia.]

b) DESCRIPCIÓN DE LOS CASOS DE USO DEL MODELO DEL NEGOCIO

A continuación se muestra los principales casos de uso identificados a partir del modelo del negocio y los requisitos funcionales que debe satisfacer el sistema. Cada uno de los casos de uso aporta un resultado de valor para sus actores.

Tabla 3.5

Caso de uso	Control de acceso al sistema
Actor	Operador
Tipo	Primario
Descripción	Este caso de uso interactúa el Encargado de almacenes, Encargado del laboratorio, Operador y Personal autorizado que deseen acceder al sistema, se encarga de controlar la validez del usuario y contraseña ingresada.

Título: Caso de uso Controlar acceso al sistema

[Fuente: Elaboración Propia.]

Tabla 3.6

Caso de uso	Control de muestras
Actor	Encargado de almacén
Tipo	Primario
Descripción	Este caso de uso interactúa con el Encargado de almacén, este registra todos los datos de las muestras de mineral a ser analizadas. Así también registra la salida de las muestras a laboratorio.

Título: Control de existencias

[Fuente: Elaboración Propia.]

Tabla 3.7

Caso de uso	Solicita análisis
Actor	Cliente
Tipo	Primario
Descripción	Esta caso de uso interactúa con el Cliente, porque es él quien solicita el análisis a sus muestras de mineral y pide un informe detallado de los resultados que dieron, por su análisis.

Título: Caso de uso Solicita análisis.

[Fuente: Elaboración Propia.]

Tabla 3.8

Caso de uso	Ingreso de muestras a almacén
Actor	Encargado de almacén
Tipo	Primario
Descripción	Este caso de uso interactúa con el Encargado de Almacenes, le permite registrar cada ingreso de muestras de mineral de los clientes al área de almacén.

Título: Caso de uso Ingreso de muestras a almacén.

[Fuente: Elaboración Propia.]

Tabla 3.9

Caso de uso	Salida de muestras de almacén
Actor	Encargado de almacén
Tipo	Primario
Descripción	Este caso de uso interactúa con el Encargado de Almacenes, le permite registrar cada una de las salidas de muestras de mineral que son enviadas al laboratorio, por solicitud del cliente.

Titulo: Caso de uso Salida de muestras de almacén.

[Fuente: Elaboración Propia.]

Tabla 3.10

Caso de uso	Ingreso de muestras a laboratorio
Actor	Encargado de laboratorio
Tipo	Primario
Descripción	Este caso de uso interactúa con el Encargado de almacén quien hace el registro de salida de almacén y es enviada a laboratorio para ser analizada, según la solicitud.

Titulo: Caso de uso Ingreso de muestras a laboratorio

[Fuente: Elaboración Propia.]

Tabla 3.11

Caso de uso	Entrega de resultados del análisis
Actor	Encargado de almacén
Tipo	Primario
Descripción	Este caso de uso interactúa con el Encargado de laboratorio, porque este entregara un informe detallado de los resultados a las muestras de mineral analizadas.

Titulo: Caso de uso Entrega de resultados del análisis

[Fuente: Elaboración Propia.]

Tabla 3.12

Caso de uso	Realiza análisis
Actor	Encargado de laboratorio
Tipo	Primario
Descripción	Este caso de uso interactúa con el Encargado de laboratorio, es este actor quien realiza el análisis a la muestra de mineral, dependiendo del tipo de servicio que solicito el cliente. Una vez analizado entrega a almacén su informe correspondiente.

Titulo: Caso de uso Realiza análisis

[Fuente: Elaboración Propia.]

Tabla 3.13

Caso de uso	Entrega de factura
Actor	Encargado de almacén y Cliente
Tipo	Primario
Descripción	Este caso de uso interactúa con el Encargado de almacén y el cliente. El cliente recibe por parte de la empresa una factura por el cobro del servicio solicitado.

Título: Caso de uso Entrega de factura

[Fuente: Elaboración Propia.]

3.2.2 REQUISITOS

A continuación mencionaremos algunos requisitos que son necesarios para el desarrollo del sistema.

Tabla 3.14

Requisitos del sistema	<ul style="list-style-type: none"> ○ Tener registros ordenados de los ingresos y salidas de las muestras de mineral. ○ Realizar un informe de las muestras que ya fueron analizadas. ○ Registrar solicitud de análisis a muestras de minerales. ○ Contar con información oportuna, actualizada de los ingresos, salidas de las muestras de minerales y reportes de estos. ○ Generar consulta sobre la existencia de muestras de minerales en almacén.
------------------------	--

	<ul style="list-style-type: none"> ○ Generar lista de solicitudes. ○ Consultar y verificar la información registrada en la base de datos. ○ Examinar las nuevas muestras de minerales que ingresan en almacén. ○ Generar copias de seguridad de la base de datos. ○ Realizar un informe de los trabajos que se realizaron.
--	---

Título: Tabla de requisitos del sistema

[Fuente: Elaboración Propia.]

Los requisitos del sistema que mencionamos en la tabla 3.14 nos ayuda a tener una idea de los puntos que se deben considerar para el desarrollo del sistema y que serán cumplidos a la finalización del mismo.

Tabla 3.15

Requerimiento del software	<p>Para la implementación del sistema es necesario contar con los siguientes recursos:</p> <ul style="list-style-type: none"> ➤ Sistema operativo Windows XP o superior. ➤ Lenguaje de programación Visual Basic.Net 2008. ➤ Motor de base de datos SQL Server 2008.
----------------------------	---

Título: Tabla de requisitos del software

[Fuente: Elaboración Propia.]

El uso de los recursos de software para el hardware tiene como base el entorno del sistema operativo Windows. Y para su implementación también hacemos el uso de la gama de programas y lenguajes que ofrece Windows, haciendo compatibles todos los recursos.

Tabla 3.16

<p>Requerimiento del hardware</p>	<ul style="list-style-type: none"> ➤ Procesador mínimo Pentium IV, mejor si es superior. ➤ Memoria RAM 512 Mb como mínimo. ➤ Disco duro de almacenamiento mayor a 50 Gigabyte.
-----------------------------------	---

Titulo: Tabla de requisitos del hardware

[Fuente: Elaboración Propia.]

Para la implantación de este proyecto se requerirá componentes de hardware, los cuales deberán ser elegidos de manera cuidadosa, componentes mínimos que necesita un sistema convencional.

Tabla 3.17

<p>Requerimiento de interfaz de usuario</p>	<ul style="list-style-type: none"> • Menús, colores. • Ingreso de datos por medio del teclado y el mouse como dispositivos de entrada de datos. • Ingreso de datos en los formularios correspondientes. • Salida de datos plasmados en formatos prediseñados, a través del monitor y la impresora.
---	--

Titulo: Tabla de requerimientos de interfaz de usuario

[Fuente: Elaboración Propia.]

El nuevo sistema informático debe tener un formato amigable que permita una interacción fácil con los usuarios. Es decir un formato agradable a la vista, por esa razón que vemos la necesidad de aplicar los requerimientos de interfaz que mencionamos en la tabla 3.17

3.3 FASE DE ELABORACIÓN

Como ya se ha visto en la fase anterior se cuenta con un modelo preliminar de los requerimientos del sistema, un diagrama de casos de uso general del sistema y la información necesaria para empezar a desarrollar el software. Es necesario, en esta fase, representar la mayor parte de los requisitos en casos de uso más detallados y en otros diagramas necesarios para entender mejor el funcionamiento del sistema.

3.3.1 MODELO DE CASOS DE USO

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se presentan mediante Diagramas de Casos de Uso, los diagramas de casos de uso definen conjuntos de funcionalidades afines que el sistema debe cumplir para satisfacer los requerimientos que tiene a su cargo.

EL propósito de estos casos de uso es:

- Determinar las necesidades de los usuarios desde el punto de vista del desarrollador, es decir, los servicios que debe proporcionar el sistema.
- Ser un punto de partida para descubrir las clases a implementar.
- Ser un punto de partida para analizar descubrir las operaciones de cada clase.

El diagrama de casos de uso simboliza la forma en cómo un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan (operaciones o casos de uso). Un diagrama de casos de uso consta de los siguientes elementos: actor, casos de uso, relaciones de uso, herencia y comunicación.

La descripción general del modelo de casos de uso explica la relación de los casos de uso entre sí mismo y con los actores. Tal como sigue a continuación:

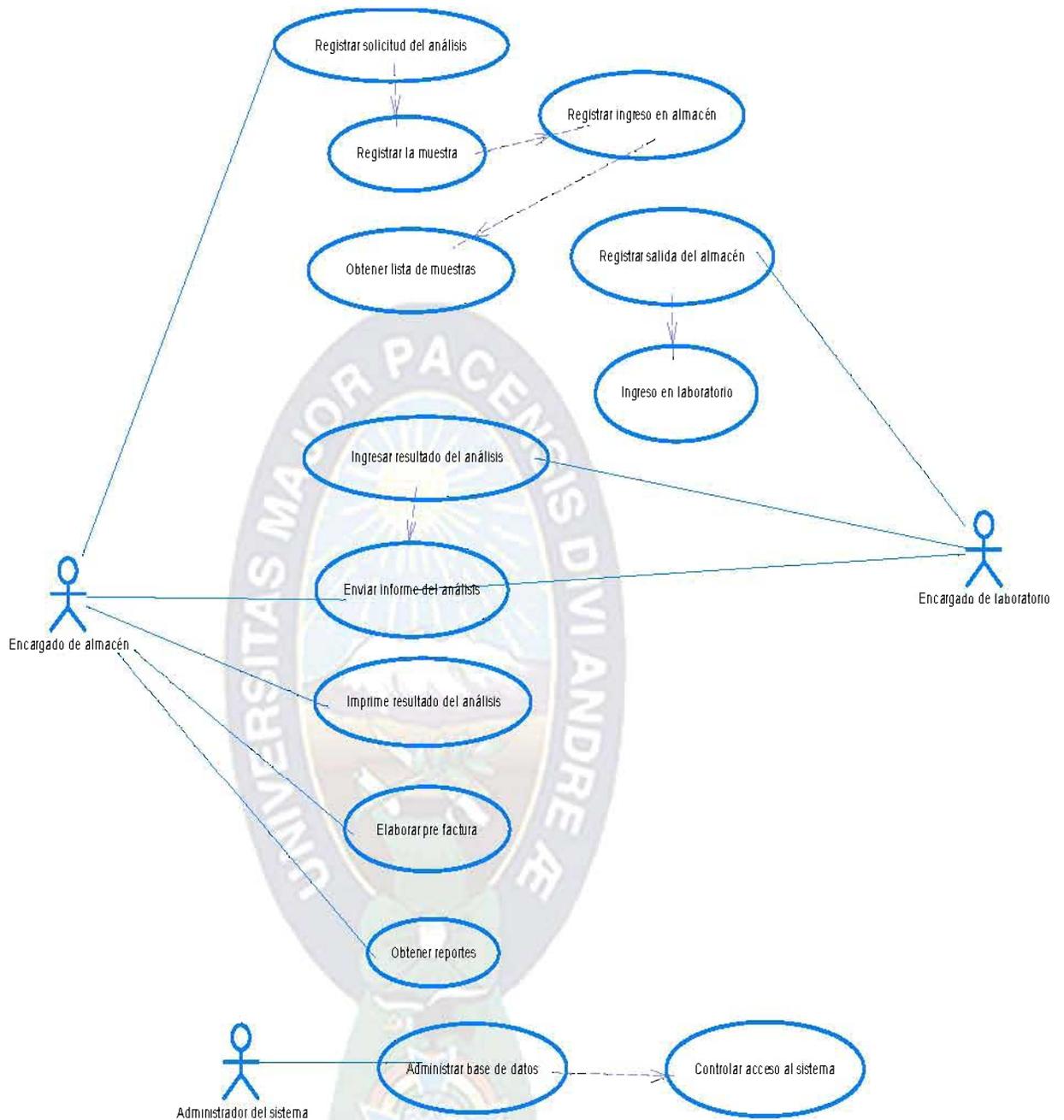


Figura 3.2 Titulo: Diagrama de caso de uso del sistema

[Fuente: Elaboración Propia]

En la figura 3.1 podemos observar el funcionamiento general del sistema, mostrando como es la interacción de los actores con el sistema.

Algunos casos de uso del diagrama de casos de uso general se desglosan en otros casos de uso específicos como se observa continuación.



Figura 3.3 Titulo: Controlar acceso al sistema.

[Fuente: Elaboración Propia]

En la figura 3.3 observamos cómo es la interacción del usuario con el sistema. En esta figura también vemos cuál es el proceso para controlar de acceso al sistema.

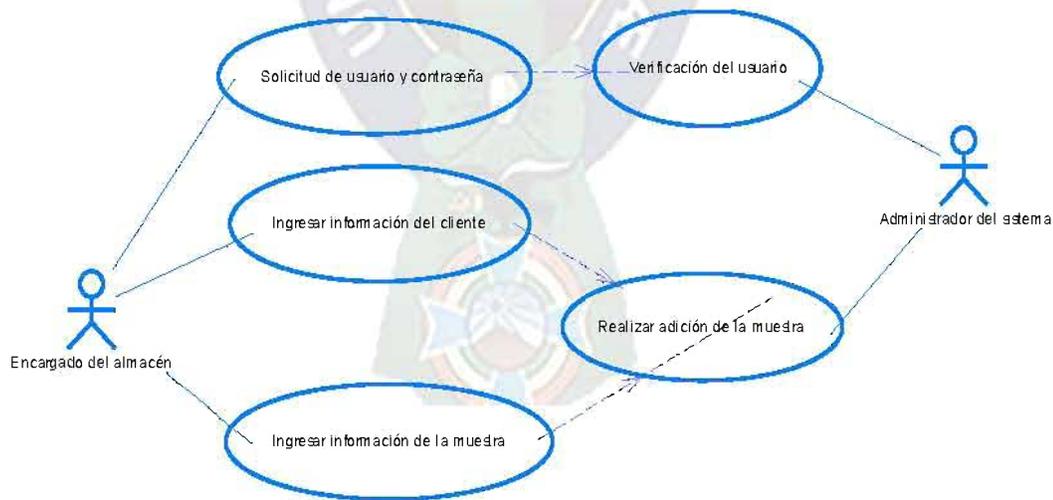


Figura 3.4 Titulo: Registrar solicitud de análisis

[Fuente: Elaboración Propia]

En la figura 3.4 el encargado del almacén interactúa con el sistema para que este pueda hacer la solicitud de análisis que el cliente solicite.

3.3.1.1 DESCRIPCIÓN DE LOS CASOS DE USO EN FORMATO EXPANDIDO

Los casos de usos expandidos detallan la conformación de los procesos paso a paso, ahora desarrollaremos cada caso de uso y los mostraremos en las siguientes tablas.

Tabla 3.18

Caso de uso	Controlar acceso al sistema	
Actor	Operador	
Propósito	Registrar al usuario, controlar o verificar a los usuarios que ingresen al sistema.	
Resumen	El operador debe estar registrado en el sistema, sino no está, se procede a registrar al nuevo usuario.	
Tipo	Primario	
Curso principal	Acción del actor	Respuesta del sistema
	<ul style="list-style-type: none"> ➤ El operador ingresa los datos del usuario. ➤ El operador llena los datos del usuario y se adiciona. 	<ul style="list-style-type: none"> ➤ Si esta registrado se muestra en pantalla la página principal. ➤ Si el usuario no está registrado, se muestra un mensaje de error. ➤ Muestra en pantalla el formulario de datos para ser registrado. ➤ Almacena los datos correspondientes.

Título: Descripción de caso de uso: Controlar acceso al sistema

[Fuente: Elaboración Propia.]

Con la tabla 3.18 controlamos el acceso al sistema considerando que el operador debe estar registrado en el sistema, sino no está, se procede a registrar al nuevo usuario.

Tabla 3.19

Caso de uso	Solicita análisis	
Actores	Cliente y Encargado de almacén	
Propósito	Hacer la solicitud de análisis a una determina muestra de mineral.	
Resumen	Se usa el caso de uso Solicita análisis, para que en los laboratorios de la empresa se pueda hacer un análisis a la muestra de mineral, con el tipo de análisis que se requiera.	
Tipo	Primario	
Curso principal	Acción del actor	Respuesta del sistema
	<ul style="list-style-type: none"> ➤ Encargado de almacén ingresa al sistema. ➤ Introduce datos de la muestra 	<ul style="list-style-type: none"> ➤ Despliega formulario de solicitud de análisis. ➤ Solicita nombre del cliente. (Sino esta registrado el cliente, despliega formulario de registro de cliente ➤ Almacena datos del cliente y las muestras.
Precondición	El sistema tiene almacenada las listas de clientes que generalmente hacen el análisis a sus muestras. Y si no registra	

	al cliente para tener un control de clientes.
Pos condición	Ninguno

Título: Descripción de caso de uso: Solicita análisis

[Fuente: Elaboración Propia.]

El cliente llena todos los datos de la muestra a ser analizada.

Tabla 3.20

Caso de uso	Registro de ingreso de muestras a almacén	
Actores	Encargado de almacén	
Propósito	Registrar todas las muestras de mineral que ingresen en almacenes.	
Resumen	Este caso de uso permite registrar todas las muestras de mineral, para introducirlas en el área de almacenes. Para su registro se deben llenar todos los datos del formulario de registro	
Tipo	Primario	
Curso principal	Acción del actor	Respuesta del sistema
	<ul style="list-style-type: none"> ➤ Ingresa al formulario de registro de muestras. ➤ Se registran los datos y se procede a 	<ul style="list-style-type: none"> ➤ El sistema solicita datos de la muestras de mineral (nombre del cliente, tipo de análisis, cantidad de muestras, asignar código a la muestra, etc.

	verificarlos.	➤ Registra las muestras.
Precondición	El sistema tiene almacenada las listas de clientes que generalmente hacen el análisis a sus muestras. Y si no registra al cliente para tener un control de clientes.	
Pos condición	Ninguno	

Título: Descripción de caso de uso: Registro de ingreso de muestras a almacén.

[Fuente: Elaboración Propia.]

En tabla 3.20 se muestra como se procede a hacer el registro de la muestra de mineral que será ingresada en el área de almacenes para luego ser analizadas.

Tabla 3.21

Caso de uso	Registro de salida de muestras a laboratorio	
Actores	Encargado de almacén	
Propósito	Registrar todas las muestras de mineral que han sido enviadas a laboratorio.	
Resumen	Este caso de uso interactúa con el Encargado de Almacenes, le permite registrar cada una de las salidas de muestras de mineral que son enviadas a laboratorio, por solicitud del cliente.	
Tipo	Primario	
Curso principal	Acción del actor	Respuesta del sistema
	<ul style="list-style-type: none"> ➤ Ingresa al sistema. ➤ Envía datos de las muestras. 	<ul style="list-style-type: none"> ➤ El sistema solicita datos de la muestras de mineral (tipo de análisis, cantidad de muestras, etc. Envía datos a

	<ul style="list-style-type: none"> ➤ Se registran los datos y se procede a verificarlos. Y así los envía a laboratorio. 	<p>laboratorio.</p> <ul style="list-style-type: none"> ➤ Verifica si llegaron los datos.
Precondición	El sistema tiene almacenado todos los datos de la muestra de mineral que serán enviadas al laboratorio.	
Pos condición	Ninguno	

Titulo: Descripción de caso de uso: Registro de salida de muestras a laboratorio.

[Fuente: Elaboración Propia.]

En la tabla 3.21 muestra como es el registro de salida de las muestras de mineral que son enviadas luego a laboratorio para ser analizadas. No se hace un doble registro sino solo se envía los datos a laboratorio.

Tabla 3.22

Caso de uso	Control de muestras	
Actor	Encargado de almacén	
Propósito	Verificar la situación actual de almacenes.	
Resumen	Obtener la cantidad de muestras de minerales ingresadas o enviadas al laboratorio, para tener un control de todas las muestras de mineral que estén en almacenes.	
Tipo	Primario	
Curso principal	Acción del actor	Respuesta del sistema
	<ul style="list-style-type: none"> ➤ El encargado de almacén ingresa al sistema para verificar la cantidad de 	<ul style="list-style-type: none"> ➤ Se muestra en pantalla

	<p>muestras en almacén.</p> <p>➤ Entonces el encargado envía las muestras a ser analizadas a laboratorio.</p>	<p>la lista de todas las muestras ingresadas o enviadas al laboratorio.</p> <p>➤ Muestra un listado de todas las muestras que deben ser enviadas al laboratorio.</p>
Precondición	El sistema tiene almacenado todos los datos de la muestra de mineral, registros de ingreso y salidas de dichas muestras.	
Pos condición	Ninguno	

Título: Descripción de caso de uso: Control de muestras

[Fuente: Elaboración Propia.]

En la tabla 3.22 observamos que accedemos al sistema para poder desplegar una lista de todas las muestras, analizadas o no.

Tabla 3.23

Caso de uso	Obtener reportes
Actores	Encargado de almacén
Propósito	Imprimir informes de usuarios, clientes y listas de todas las muestras de mineral que se encuentren en almacén o si ya fueron enviadas a laboratorio.
Resumen	Este caso de uso interactúa con el área de Almacenes le permite conocer la situación real de cada movimiento a las muestras de mineral, (saber si están aun en almacenes o ya fueron enviadas a laboratorio) a través de generación de reportes impresos de las salidas e ingresos de muestras de

	mineral, control de existencias en almacén, etc.	
Tipo	Primario	
Curso principal	Acción del actor	Respuesta del sistema
	<ul style="list-style-type: none"> ➤ El encargado de almacén ingresa al sistema. ➤ El encargado del sistema elige el reporte deseado. 	<ul style="list-style-type: none"> ➤ El sistema despliega la ventana de reportes. ➤ Muestra ventana de reportes
Precondición	El sistema tiene almacenado todas las listas de la muestra de mineral, usuarios, clientes y personal autorizado al sistema.	
Pos condición	Ninguno	

Título: Descripción de caso de uso: Obtener reportes

[Fuente: Elaboración Propia.]

En esta tabla 3.23 observamos que podemos desplegar los reportes que el usuario deseo visualizar.

Tabla 3.24

Caso de uso	Registro de muestras analizadas	
Actores	Encargado de almacén	
Propósito	Registrar todos los datos del resultado a las muestras de mineral analizadas.	
Resumen	Este caso de uso interactúa con el Encargado de Almacenes, porque es él quien registrara todas las muestras de mineral que ya fueron analizadas, a su vez registrara todos los datos que dieron como resultado del análisis.	
Tipo	Primario	
Curso principal	Acción del actor	Respuesta del sistema
	<ul style="list-style-type: none"> ➤ Ingresa al sistema. 	

	<ul style="list-style-type: none"> ➤ Formulario de registro de muestras analizadas. ➤ Llena los resultados del análisis. 	<ul style="list-style-type: none"> ➤ Muestra página principal. ➤ El sistema muestra el formulario de registro de muestras analizadas. ➤ Almacena los datos de muestras analizadas.
Precondición	El sistema tiene almacenado todos los datos de la muestra de mineral cuando se las registro para el ingreso de almacén.	
Pos condición	Ninguno	

Título: Descripción de caso de uso: Registro de muestras analizadas

[Fuente: Elaboración Propia.]

En la tabla 3.24 se realiza el llenado de los resultados de las muestras que ya fueron analizadas en los laboratorios y ahora solo se espera registrar esos datos e imprimirlos.

Tabla 3.25

Caso de uso	Elaboración de facturas
Actores	Encargado de almacén
Propósito	Elaborar una factura con el detalle respectivo del análisis, para un mejor control.
Resumen	Este caso de uso interactúa con el área de Almacenes, para ejecutar el debido cobro por el análisis que se realizó en los laboratorios. Permite emitir una factura de acuerdo al tipo de

	análisis y un detalle del mismo.	
Tipo	Primario	
Curso principal	Acción del actor	Respuesta del sistema
	<ul style="list-style-type: none"> ➤ Ingresa al sistema. ➤ Introduce los datos para llenar factura. 	<ul style="list-style-type: none"> ➤ El sistema muestra el formulario de facturación. ➤ Guarda datos. ➤ Imprime Factura. ➤ Imprime resultado del análisis.
Precondición	El sistema debe tener almacenado los datos que dieron como resultado del análisis a las muestras de mineral.	
Pos condición	Ninguno	

Título: Descripción de caso de uso: Elaboración de facturas

[Fuente: Elaboración Propia.]

En la tabla 3.25 vemos que se ha hecho el llenado de la factura con los datos del cliente y la muestra y además se imprimirá un detalle de los resultados de los análisis a las muestras de mineral.

3.3.2. DIAGRAMAS DE ESTADO DEL SISTEMA

Este diagrama se utiliza para describir el comportamiento de un sistema, representa los diferentes estados que puede adquirir una clase, como representarla a diferentes etapas de su vida. El estado de un objeto se puede caracterizar por el valor de uno varios de los atributos de su clase, así también se caracteriza por la existencia de un enlace con otro objeto.

A continuación mostraremos los diagramas de estado del sistema que se emplearon para el desarrollo del sistema.

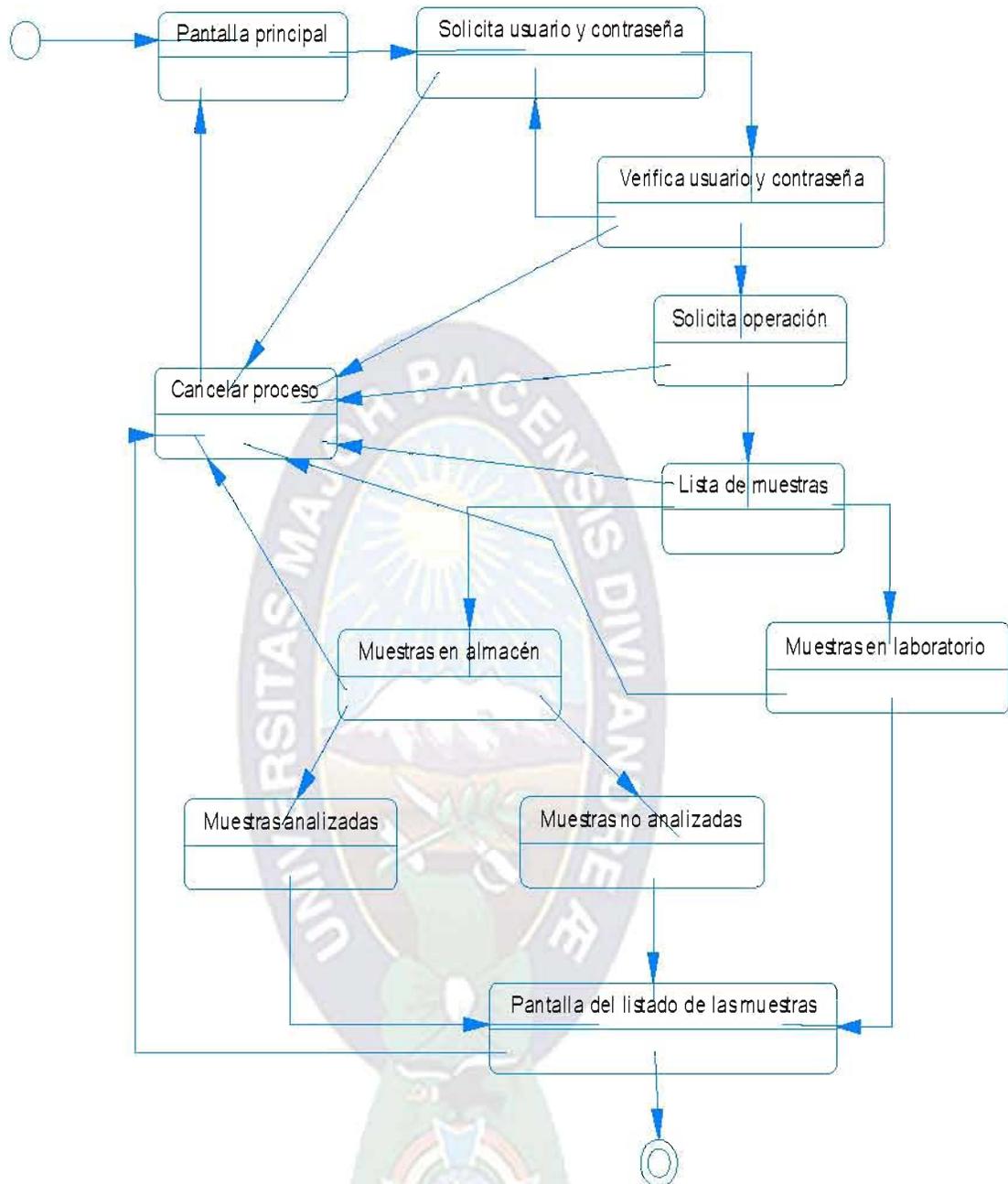


Figura 3.6 Título: Listado de muestras

[Fuente: Elaboración Propia]

En la figura 3.6 muestra el listado de las muestras que ingresan en almacén, se puede desplegar lista de muestras analizadas, como de las que no hay sido analizadas. Así mismo podemos mostrar aquellas muestras de mineral que ya fueron enviadas a laboratorio para ser analizadas.

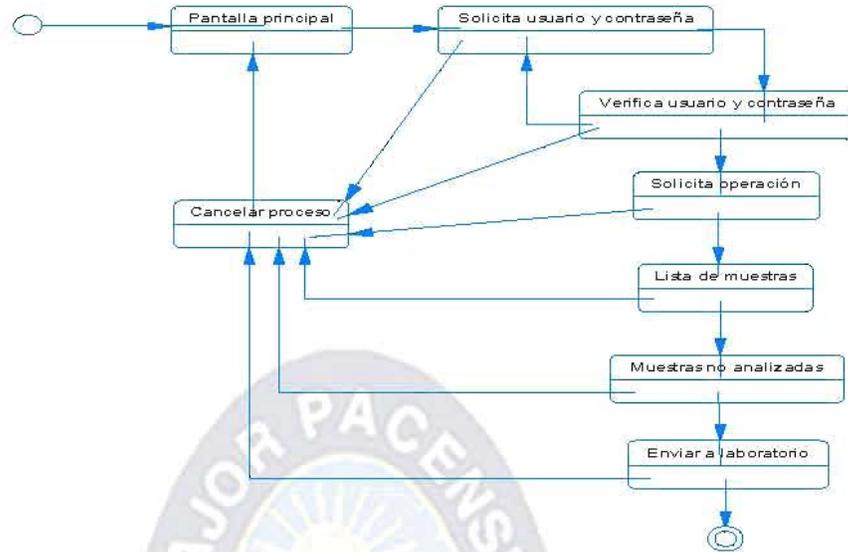


Figura 3.7 Titulo: Registro de la muestra en laboratorio

[Fuente: Elaboración Propia]

En la figura 3.7 hacemos el registro de todas las muestras que no han sido analizadas, al no haber sido analizadas pasan a laboratorio para su respectivo análisis.

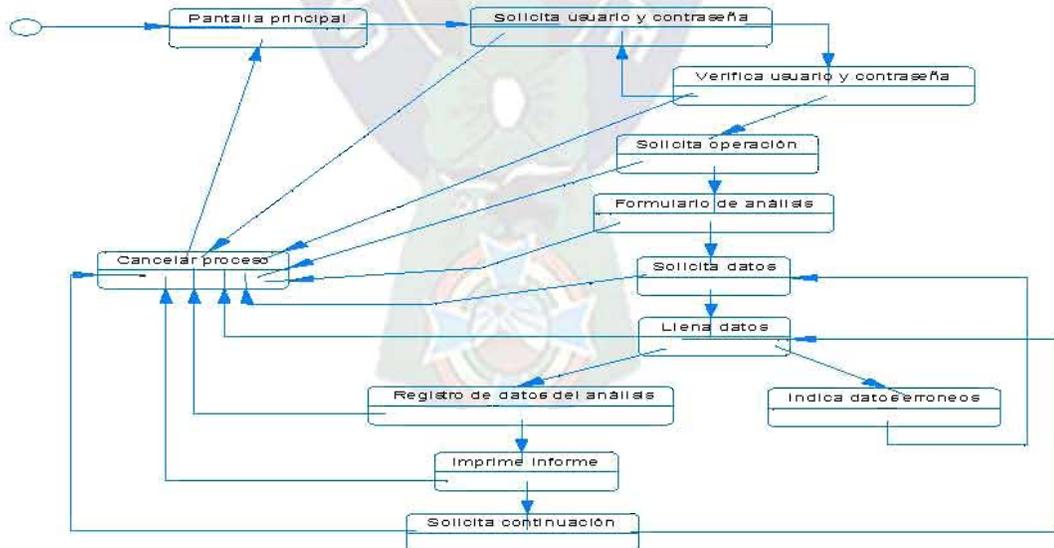


Figura 3.8 Titulo: Registro de datos de la muestra analizada

[Fuente: Elaboración Propia]

En la figura 3.8 pasamos a registrar los datos que dieron como resultado de los análisis a las muestra de mineral y esos mismos datos son impresos luego para adjuntar a la factura el detalle del servicio realizado.

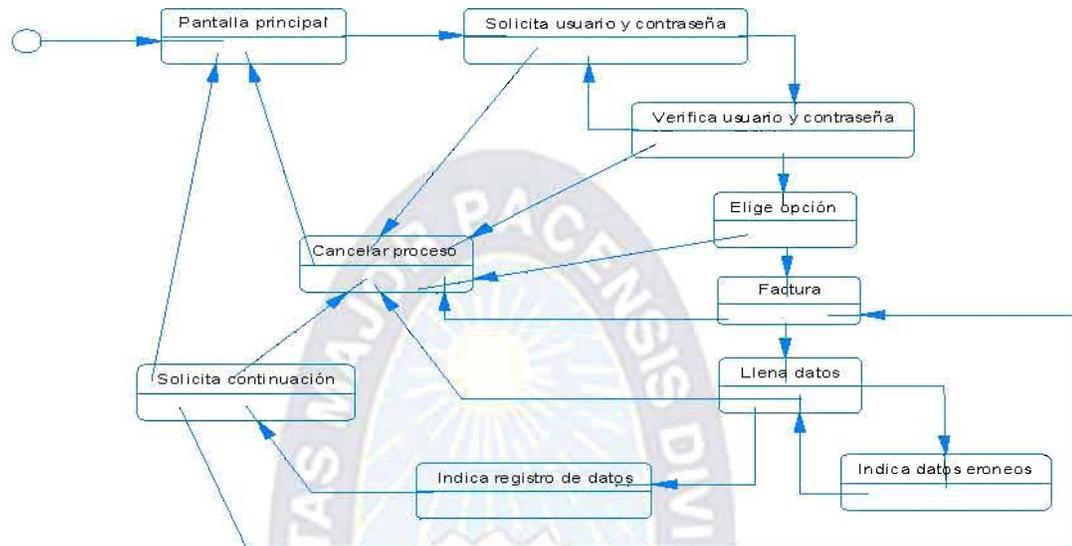


Figura 3.9 Título: Elaboración de factura

[Fuente: Elaboración Propia]

En la figura 3.9 se observa que se procede al llenado de la factura, datos que son introducidos por el encargado del almacén. Luego de introducir los datos necesarios se procede a imprimir dicha factura para el cliente.

3.3.3 DIAGRAMAS DE SECUENCIA DEL SISTEMA

En los diagramas de secuencia de un sistema se muestra gráficamente los eventos que fluyen de los actores al sistema, este debe prepararse para el curso normal de los eventos de un caso de uso, teniendo en cuenta los cursos opcionales más interesantes.

Mediante estos diagramas de secuencia se explicara los pasos a seguir en forma secuencial de los diferentes procesos que existen en el sistema actual, comenzando de la acción del actor y la respuesta del sistema mediante las clases existentes.

A continuación se observaran los diagramas de secuencia más importantes para el desarrollo del sistema.



Figura 3.10 Titulo: Diagrama de secuencia: Controlar acceso al sistema

[Fuente: Elaboración Propia]

En esta figura observamos al operador haciendo su ingreso al sistema y el sistema verifica si el usuario ingresado puede acceder al sistema.

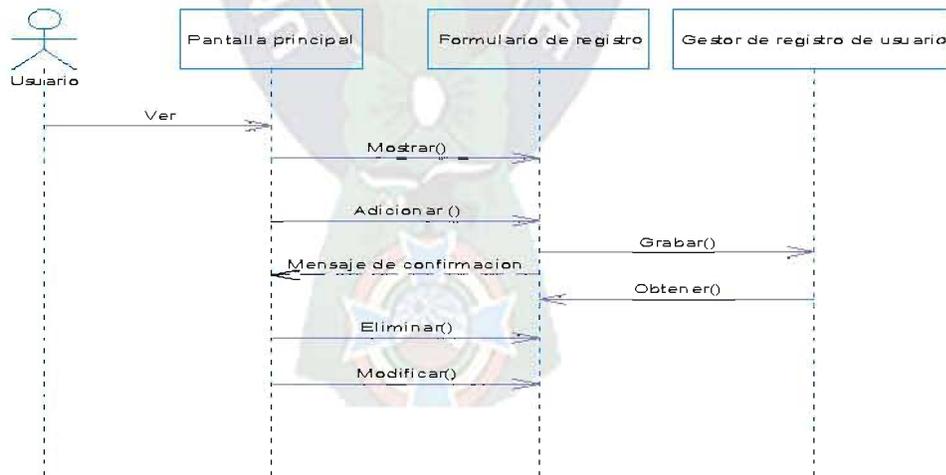


Figura 3.11 Titulo: Diagrama de secuencia: Registro de usuarios

[Fuente: Elaboración Propia]

Esta figura muestra como se puede registrar un usuario para poder acceder al sistema.



Figura 3.12 Titulo: Diagrama de secuencia: Verificar usuario

[Fuente: Elaboración Propia]

La figura 3.12 verifica si el usuario que dese ingresar al sistema está registrado en la base de datos.

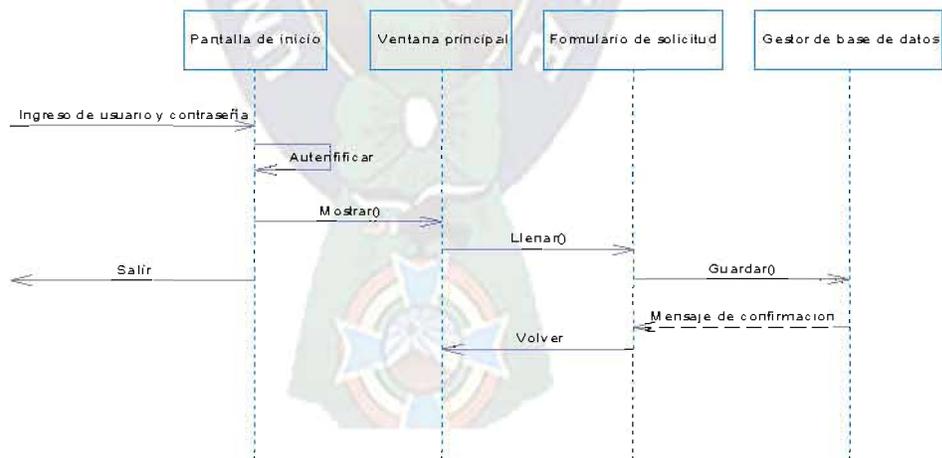


Figura 3.13 Titulo: Diagrama de secuencia: Solicitud de análisis

[Fuente: Elaboración Propia]

La figura 3.13 muestra cómo puede hacer la solicitud de análisis, ingresando por la pantalla principal y luego accediendo al formulario de solicitud.

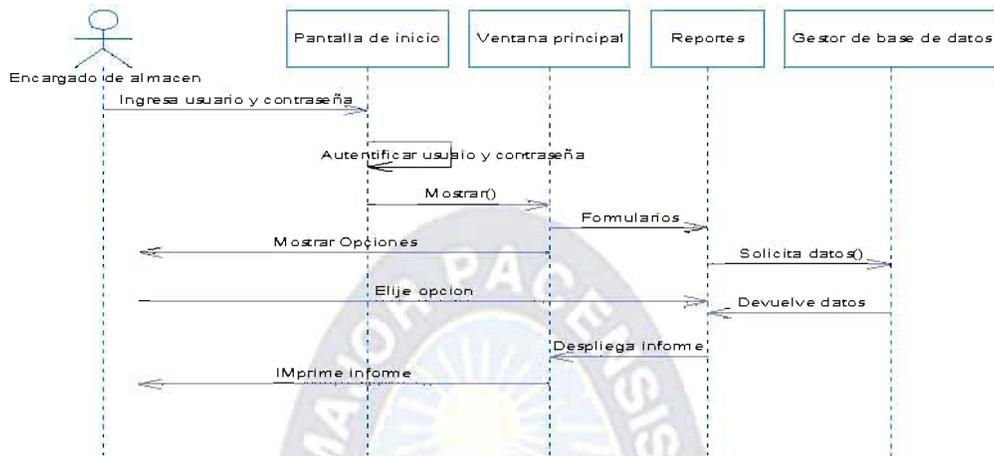


Figura 3.14 Título: Diagrama de secuencia: Obtener reportes

[Fuente: Elaboración Propia]

La figura 3.14 permite generar reportes de un determinado formulario, imprime los datos que hayan seleccionado.



Figura 3.15 Título: Diagrama de secuencia: Imprimir factura

[Fuente: Elaboración Propia]

La figura 3.15 muestra cómo obtener e imprimir la factura que se tiene que entregar al cliente que solicitó el análisis.

Los diagramas generados anteriormente son producto de la realización constante que se llevó a cabo durante el desarrollo del sistema, las historias que son documentos utilizados en la adquisición de requerimientos son pieza importante para conseguir y seguir con el desarrollo y un factor clave es la de estar constantemente con el usuario en este caso el sistema con el cual se trabajó en la especificación de los requerimientos y la búsqueda de la funcionalidad deseada. La realización no es solo aplicada en el diseño procedimental del software sino también en el diseño de la documentación que se amolda a los cambios establecidos entre el usuario y desarrollador.

3.3.4 DIAGRAMA DE COMPONENTES

El diagrama de componentes nos permite ver la estructura de los módulos ejecutables del sistema.

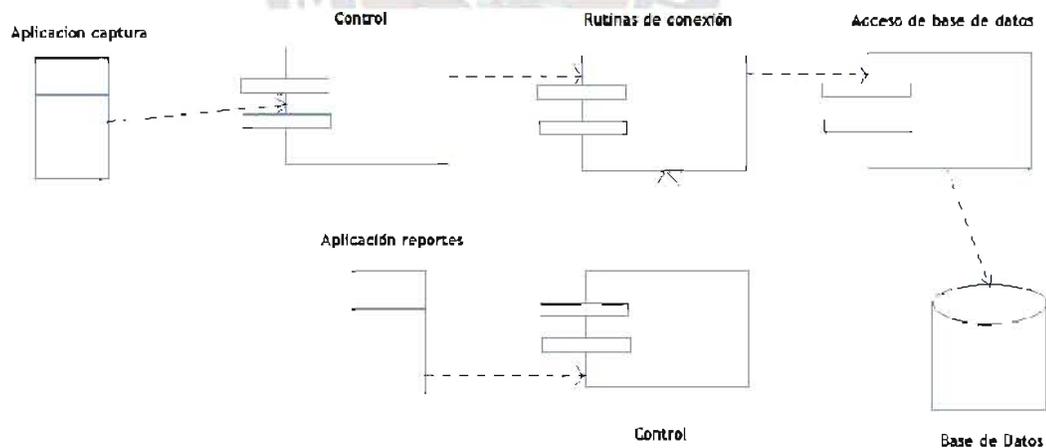


Figura 3.16 Titulo: Diagrama de componentes del sistema: de control

[Fuente: Elaboración Propia]

El diagrama de componentes de control interactúa con la base de datos para la aplicación de reportes.

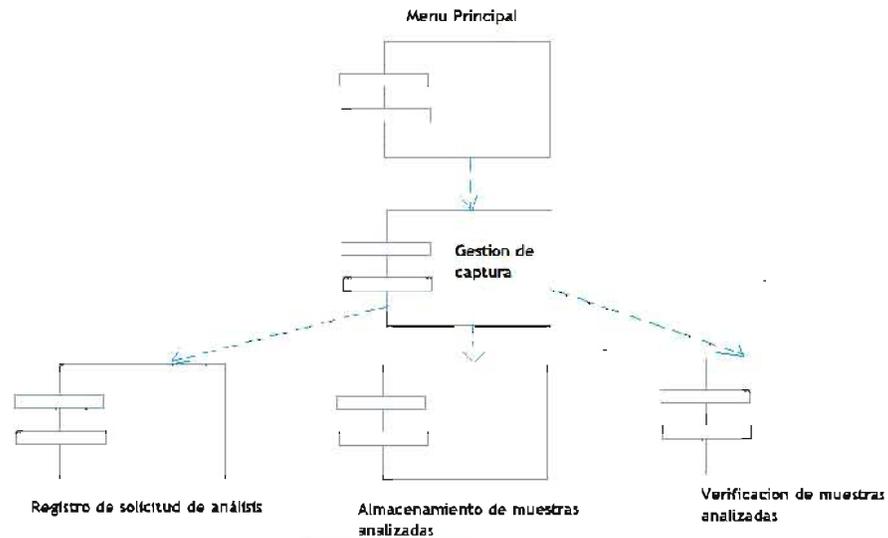


Figura 3.17 Titulo: Diagrama de de componentes del sistema: Gestión de captura

[Fuente: Elaboración Propia]

El diagrama de componentes de gestión de captura permite mostrar los registros, almacenamiento de las muestras analizadas y su verificación.

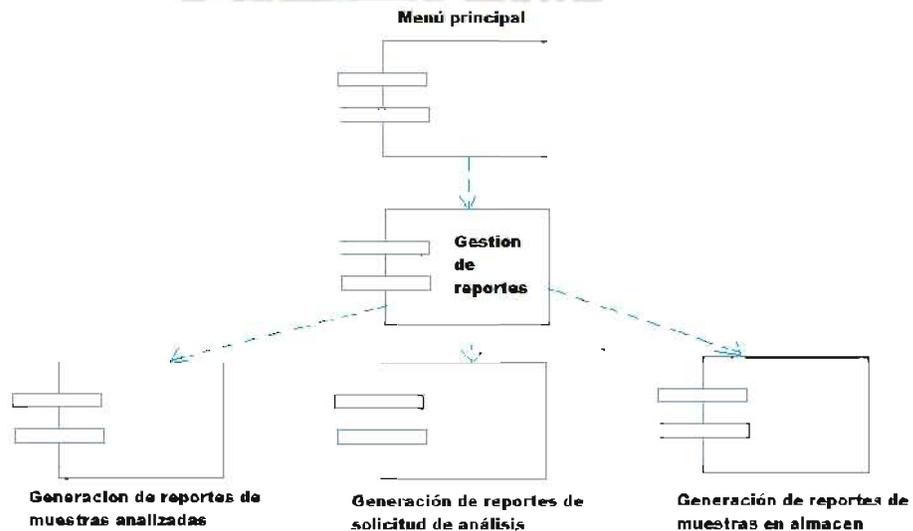


Figura 3.18 Titulo: Diagrama de de componentes del sistema: Gestión de reportes

[Fuente: Elaboración Propia]

Este diagrama para gestión de reportes obtiene listas e informes de los resultados de análisis de muestras de minerales.

3.3.5 CONTRATO DE OPERACIONES

Los contratos de operaciones ayudan a describir el comportamiento del sistema. Un contrato de operación describe el cambio de un estado a otro de los objetos del dominio como resultado de la ejecución de una operación. Su creación depende de la creación previa del modelo de dominio, los diagramas de secuencia de sistema, y la identificación de las operaciones de sistema. A continuación veremos los principales contratos de operaciones para el sistema.

Tabla 3.26

Nombre:	Ingresar usuario
Referencias cruzadas:	Registro de usuarios
Precondiciones:	Registro en curso.
Post-condiciones:	<ul style="list-style-type: none"> • Se creó un nuevo registro para un determinado usuario. • Al ingresar a un usuario se permite tener acceso al sistema. • El registro de usuarios permitirá que el usuario no tenga problemas con el control de acceso al sistema, porque primero verificara que el usuario este registrado en el sistema.

Título: Contrato de operaciones para: Ingresar usuario.

[Fuente: Elaboración propia]

La tabla 3.26 nos muestra como el usuario ingresa al sistema, siempre y cuando el usuario esta registrado en el sistema.

Tabla 3.27

Nombre:	Verificar usuario
Referencias cruzadas:	Verificar usuarios
Precondiciones:	Poder verificar el ingreso a cualquier usuario
Post-condiciones:	<ul style="list-style-type: none"> • Se creara un registro del usuario que ya ingreso al sistema. • Ya verificados los datos del usuario que son correctos, se ingresa a la página principal del sistema.

Titulo: Contrato de operaciones para: Verificar usuarios

[Fuente: Elaboración propia]

La tabla 3.27 hace la verificación del usuario, permitiendo controlar que personas ajenas no ingresen al sistema. Con este procedimiento hacemos el control a todo el personal que trabaja en la empresa.

Tabla 3.28

Nombre:	Imprimir informes
Referencias cruzadas:	Obtener reportes
Precondiciones:	Registro de todos los datos de las muestras de mineral al sistema.
Post-condiciones:	<ul style="list-style-type: none"> • Imprimir un listado de todas las muestras de minerales que se ingresaron o salieron de almacén. • Imprimir informes del los resultados de los

	<p>análisis a las muestras de mineral.</p> <ul style="list-style-type: none"> • Actualizar los datos del control existencias de las muestras en almacén.
--	---

Titulo: Contrato de operaciones para: Obtener reportes

[Fuente: Elaboración propia]

La tabla 3.28 obtiene reportes de las muestras analizadas, muestras en laboratorio y muestras que esperan ser analizadas. Con estos reportes se tiene un control y seguimiento a las muestras de mineral.

3.3.6 ANÁLISIS Y DISEÑO

Análisis y diseño orientado a objetos (ADOO) es un enfoque de la ingeniería de software que modela un sistema como un grupo de objetos que interactúan entre sí. Este enfoque representa un dominio en términos de conceptos compuestos por verbos y sustantivos, clasificados de acuerdo a su dependencia funcional.

En este método de análisis y diseño se crea un conjunto de modelos utilizando una notación acordada es por eso que para el desarrollo del sistema hemos empleado el lenguaje unificado de modelado (UML).

La función del Análisis es ser dar soporte a las actividades de un negocio, o desarrollar un producto que pueda venderse para generar beneficios, es por eso que se deben tomar en cuenta estos elementos fundamentales:

- Software
- Hardware
- Personal
- Base de datos
- Documentación

- Procedimientos

El lenguaje unificado de modelado se ha vuelto el lenguaje de modelado estándar usado en análisis y diseño orientado a objetos.

3.3.6.1 MODELO DE ANALISIS Y DISEÑO

El sistema se ha logrado modelar de acuerdo al siguiente Diagrama de clases que se muestra.

Del Diagrama de clases de la Figura 3.19 se ha logrado identificar las entidades, atributos y relaciones, esto considerando sólo las clases más importantes.

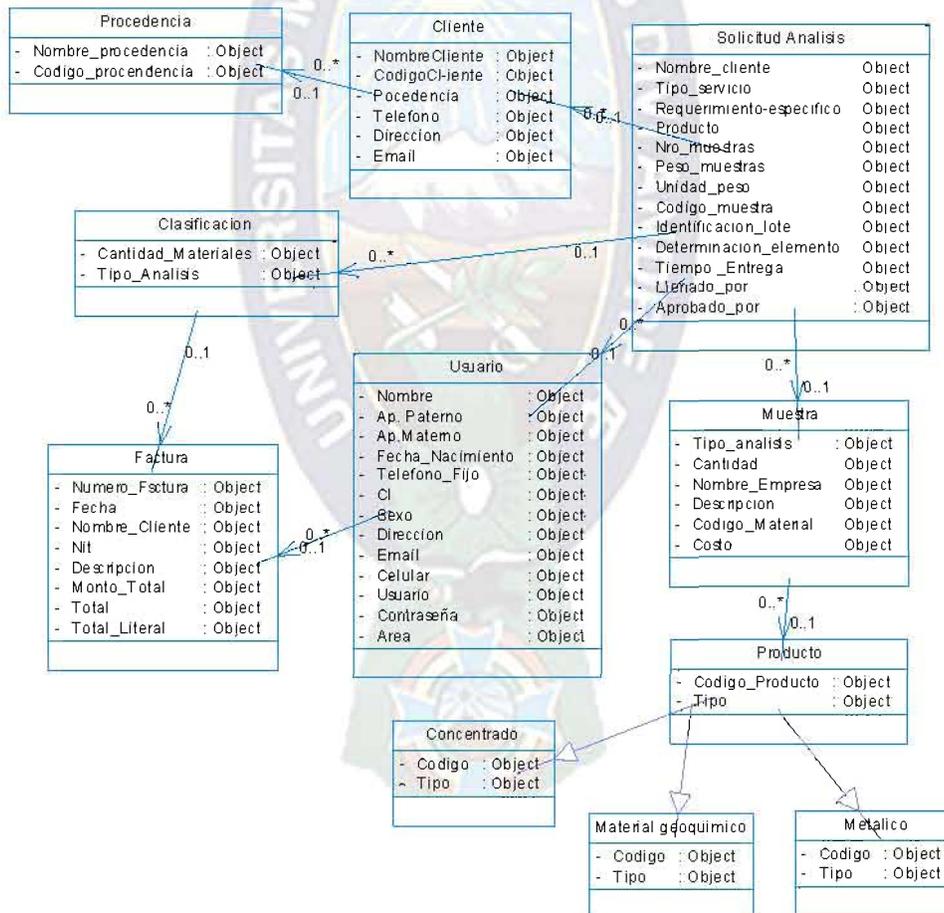


Figura 3.19 Titulo: Diagrama de clases del sistema

[Fuente: Elaboración Propia]

Se logro obtener una estructura de la base de datos, en la cual se busca que puedan efectuarse combinaciones de consultas dependiendo de la necesidad del usuario, además tomando en cuenta que los datos deben ser precisos y consistentes, obteniendo un almacenamiento, una recuperación y actualización de datos eficiente.

3.3.6.2 MODELO DE DATOS

Del Diagrama de clases de la Figura 3.19 se ha logrado identificar las entidades, Atributos y relaciones, esto considerando sólo las clases más importantes.

De donde se ha logrado elaborar el Diagrama Entidad- Relación que nos proporciona una representación grafica más clara de nuestra base de datos la cual se puede observar en la Figura 3.20.

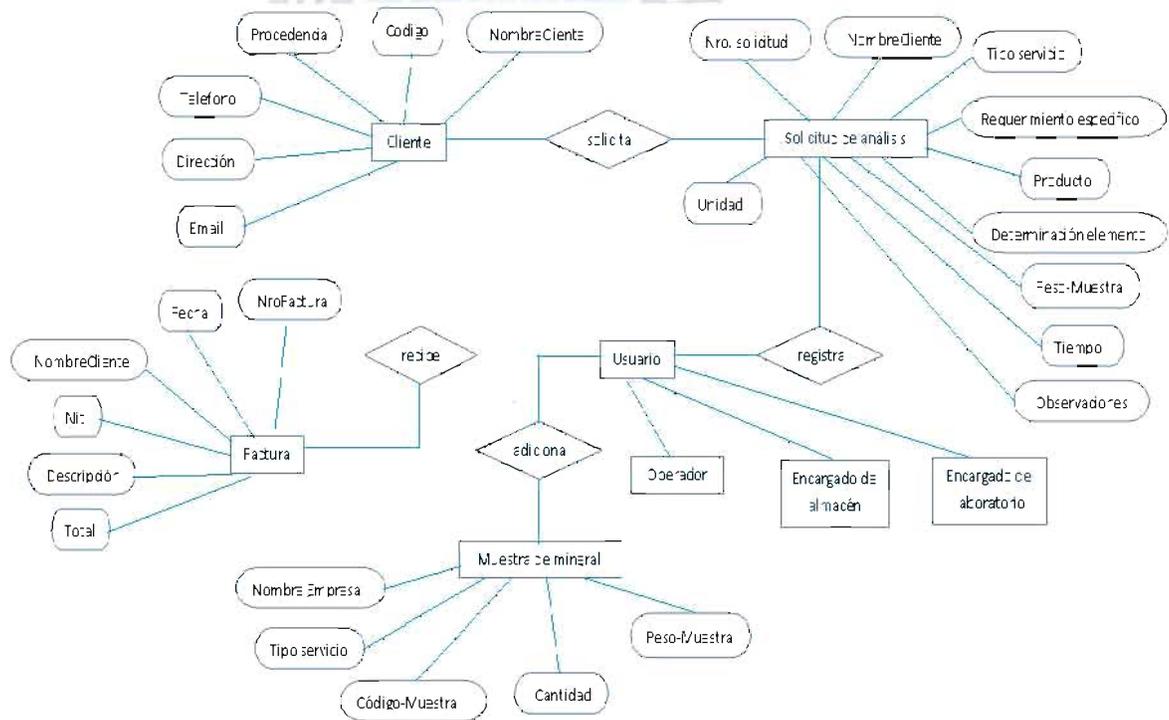


Figura 3.20 Titulo: Modelo entidad-relación

[Fuente: Elaboración Propia]

3.4. FASE DE CONSTRUCCIÓN

3.4.1. IMPLEMENTACIÓN DEL SISTEMA

Partiendo del modelado realizado, se desarrollara en el lenguaje de programación C# con código de aplicación en Visual Basic.Net pero la implementación de un sistema de software es algo más que únicamente escribir y compilar código. Debemos considerar aspectos de tecnologías utilizadas como la interacción con base de datos en nuestro caso el SQL Server 2008.

3.4.1.1. INTERFACES DEL SISTEMA

En función del modelado realizado, se generara la interfaz del sistema en un lenguaje de programación concreto, C# el código de la aplicación bajo Visual Basic.NET lenguaje de programación orientado a objetos.

A continuación especificamos algunos de los módulos de la aplicación.

- Autenticación del Sistema

Formulario de Ingreso

Alex Stewart Assayers

ARA EN RPLIMA

Usuario:
Enrique

Contraseña:

Ingresar

Nota:
Si aun no esta registrado en el sistema, proceda a registrarse.

Registrarse

Figura 3.21 Titulo: Formulario de ingreso al sistema

[Fuente: Elaboración Propia]

- **Menú Principal:** Ingreso a los formularios donde se encuentran los registros, listas, etc.



Figura 3.22 Titulo: Pantalla principal

[Fuente: Elaboración Propia]

- **Registro de usuarios:** Este formulario permite que un usuario ajeno al sistema, pueda hacer su registro, y de este modo poder acceder al sistema.

Datos Personales	
Nombre: Manuel	C.I. 1985687
Apellido Paterno: Álvarez	Dirección: Obreros calle N° 213
Apellido Materno: López	Email: manuel-alvarez@hotmail.com
Fecha de Nacimiento: 25/03/1987	Cellular: 72589659
* Teléfono fijo:	Sexo:

Datos Usuario	
Usuario: manuel	Registrar
Contraseña: *****	Nuevo
Año: Atención	

Figura 3.23 Titulo: Formulario de registro de usuario

[Fuente: Elaboración Propia]

- **Registro de clientes:** Este formulario permite registrar a todos los clientes que deseen hacer un respectivo análisis a una determinada muestra de mineral. Son registrados para tener mayor información sobre ellos.



Figura 3.24 Titulo: Formulario de registro de cliente

[Fuente: Elaboración Propia]

- **Solicitud de análisis:** En este formulario se registran los datos de las muestras a ser analizadas para tener un control de cuantas solicitudes se han realizado.

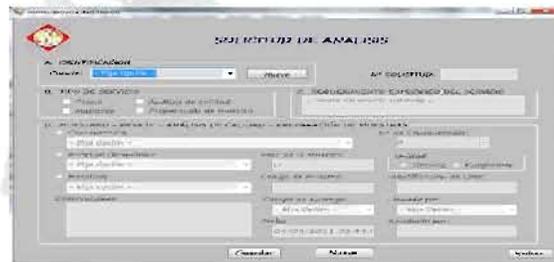


Figura 3.25 Titulo: Formulario de solicitud de análisis

[Fuente: Elaboración Propia]

- **Factura:** Este formulario, registra los datos obtenidos por el análisis de una determinada muestra de mineral, y es llenado para que el cliente pueda tener un comprobante del pago por el servicio realizado.



Figura 3.26 Titulo: Formulario de registro de factura

[Fuente: Elaboración Propia]

- **Lista de solicitudes:** Este formulario, muestra los datos de todas las solicitudes que se han realizado.



Figura 3.27 Titulo: Formulario de lista de solicitudes

[Fuente: Elaboración Propia]



Capítulo IV

Calidad de Software

4.1. INTRODUCCIÓN

Un elemento clave para cualquier proceso es la medición. Empleamos medidas para entender mejor los atributos de los modelos que creamos pero fundamentalmente empleamos las medidas para valorar la calidad de los productos de ingeniería o de los sistemas que construimos.⁶

El American Heritage Dictionary define la calidad como “una característica o atributo de algo”, como un atributo de un elemento la calidad se refiere a las características mensurables cosas que se pueden comparar con estándares conocidos¹.

Cuando se examina un elemento según sus características mensurables se pueden encontrar dos tipos de calidad: calidad del diseño y calidad de concordancia.

En el desarrollo de software, la calidad de diseño acompaña a la calidad de los requisitos, especificaciones y diseño del sistema. La calidad de concordancia es un aspecto centrado principalmente en la implementación; Si la implementación sigue al diseño, y el sistema resultante cumple con los objetivos de requisitos y de rendimiento, la calidad de concordancia es alta.

Se puede seguir los siguientes aspectos para evaluar la calidad del software:

- Funcionalidad
- Confiabilidad
- Usabilidad
- Eficiencia
- Mantenibilidad

⁶ [Pressman, 2005]

- Portabilidad
- Escalabilidad

Estos aspectos ayudaran a establecer un sistema de mayor calidad. La calidad de software no se certifica, lo que se certifica son los procedimientos para construir un software de calidad, los procedimientos deben ser correctos y estar en función de la normalización.

4.2. TIPOS DE MÉTODOS DE CALIDAD

A continuación se mencionan algunas métricas técnicas del software y métodos existentes para realizar la medición de la calidad de software:

- ❖ Factores de Calidad de McCall.
- ❖ FURPS.
- ❖ ISO9126.
- ❖ Métricas Bang.
- ❖ Métricas de Diseño a Nivel de Componentes.
- ❖ Web Site QEM.

4.2.1. ESTÁNDAR ISO 9126

El estándar ISO 9126 ha sido desarrollado en un intento de identificar los atributos claves de calidad para el software. El estándar identifica seis atributos claves de calidad: funcionalidad, confiabilidad, usabilidad, eficiencia, facilidad de mantenimiento y portabilidad.

Los factores ISO 9126 no necesariamente son utilizados para medidas directas, en cualquier caso facilitan una valiosa base para medidas indirectas y un excelente lista para determinar calidad de un sistema. Esta norma se orienta a dos

áreas para la calidad del producto software, calidad interna - externa, y calidad en uso del producto software ⁷

Modelo de Calidad Interna y Externa: este modelo determina características y sub características, que sirven para evaluar la calidad interna y la calidad externa de un producto software como sigue: (1) Funcionalidad, (2) Confiabilidad, (3) Usabilidad, (4) Eficiencia, (5) Mantenimiento, (6) Portabilidad.

Modelo de Calidad en Uso: este modelo propone métricas de calidad en uso categorizadas en las cuatro características siguientes: (1) Efectividad, (2) Productividad, (3) Seguridad Física, (4) Satisfacción.

Dentro del presente proyecto de grado para la empresa “Alex Stewart Assayers” se ve por conveniente aplicar los estándares ISO 9126 en relación a la confiabilidad y portabilidad, debido a que el sistema debe enviar y recibir datos confiables, además de ser portable de acuerdo a las necesidades de cada área de trabajo.

4.2.1.1. CARACTERÍSTICAS DEL ESTÁNDAR ISO/IEC 9126

a) FUNCIONALIDAD

Conjunto de atributos que relacionan la existencia de un conjunto de funciones con sus propiedades especificadas. Las funciones satisfacen necesidades especificadas o implícitas.

- Adecuación: atributos que determinan si el conjunto de funciones son apropiadas para las tareas especificadas.

⁷ Norma ISO/IEC 9126-1: 2001 – Software engineering – Product quality – Part 1: Quality model, June 2001; Part 2: External metrics; Part 3: Internal metrics.

- Exactitud: atributos que determinan que los efectos sean los correctos o los esperados.
- Seguridad: atributos que miden la habilidad para prevenir accesos no autorizados, ya sea accidentales o deliberados, tanto a programas como a datos.
- Interoperabilidad: atributos que miden la habilidad de interactuar con sistemas especificados.
- Cumplimiento: atributos que hacen que el software adhiera a estándares relacionados con la aplicación, y convenciones o regulaciones legales.

El punto función es una métrica orientada a la función del software y del proceso por el cual se desarrolla. Se centra en la funcionalidad o utilidad del programa, los puntos de función se calculan realizando una serie de actividades, comenzando por determinar los siguientes números:

- Número de entradas de usuario: Se cuenta cada entrada de usuario que proporciona al software diferentes datos orientados a la aplicación.
- Número de salidas de usuarios: Estas se refieren a informes, mensaje de error, etc. Es decir, salidas que proporcionan al usuario información orientada a la aplicación.
- Número de peticiones de usuario: Esta petición está definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida interactiva.
- Número de archivos: Se cuenta cada archivo maestro lógico.

- Número de interfaces externas: Se cuenta todas las interfaces legibles por el ordenador que son solicitados para transmitir información a otro sistema.

De acuerdo a lo mencionado es que se tiene los resultados en la siguiente tabla:

Tabla 4.1

Descripción	Cantidad
Número de entradas de usuario	25
Número de salidas de usuarios	17
Número de peticiones de usuario	15
Número de archivos	19
Número de interfaces externas	0

Titulo: Parámetros de medición.
[Fuente: Elaboración Propia]

Esta tabla muestra las diferentes cantidades que se hacen uso al momento de acceder al sistema, cuyos datos nos ayudan a determinar el punto de función.

Con los datos obtenidos, considerando un factor de ponderación medio.

Tabla 4.2

Parámetros de medición	Cuenta	Factor de ponderación (Medio)	Totales
Número de entradas de usuario.	25	4	100
Número de salidas de usuario.	17	5	85
Número de peticiones de usuario.	15	4	60
Número de archivos.	19	10	190
Número de interfaces externas.	0	7	0
Cuenta Total			435

Titulo: Calculo de métricas de punto función.
[Fuente: Elaboración Propia]

Para el cálculo de métricas del punto función, se consideran los valores calculados anteriormente y así obtener los totales.

La relación que permite calcular los puntos de función es la siguiente:

$$PF = Cuenta\ Total * (Grado\ de\ Confiabilidad + Tasa\ de\ error * \sum Fi)$$

Donde:

$$PF = Medida\ de\ Funcionalidad$$

Cuenta Total: Es la suma del valor de las entradas, salidas, peticiones, interfaces externas y archivos.

Grado de confiabilidad: Es la confiabilidad estimada del sistema.

Tasa de error: Probabilidad subjetiva estimada del dominio de la información, este error estimado es del 1 %.

Fi: son valores de ajuste de complejidad que toman los valores de la tabla 4.3 y que dan respuesta a las preguntas de la tabla 4.4

Tabla 4.3

Sin Importancia	0
Incidental	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

Titulo: Valores de ajuste de complejidad.
[Fuente: Elaboración Propia]

Tabla 4.4

Factor / Escala	Sin Importancia (0)	Incidental (1)	Moderado (2)	Medio (3)	Significativo (4)	Esencial (5)
1 ¿Requiere el sistema copias de seguridad y de recuperación fiable?						X
2 ¿Se requiere comunicación de datos?				X		
3 ¿Existen funciones de procesos distribuidos?			X			
4 ¿Es crítico el rendimiento?		X				
5 ¿Será ejecutado el sistema en S.O. existente?					X	
6 ¿Requiere el sistema de entrada interactiva en línea?				X		
7 ¿Requiere el sistema entrada de datos interactiva sobre múltiples ventanas?					X	
8 ¿Se actualizan los archivos maestros de manera interactiva?				X		
9 ¿Son complejas las entradas, las Salidas, los archivos o las peticiones?			X			
10 ¿Es complejo el procesamiento interno?				X		
11 ¿Se ha diseñado el código para ser reutilizable?					X	
12 ¿Están incluidas en el diseño la conversión y la instalación?				X		
13 ¿Se ha diseñado el sistema para soportar múltiples instalaciones?				X		
14 ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?						X
Total (ΣFi)				45		

Título: Tabla de preguntas del punto función.

[Fuente: Elaboración Propia]

Con la obtención de los anteriores datos y considerando un grado de confiabilidad mínimo del % es que a continuación calculamos el valor de PF:

$$PF = Cuenta\ Total * (Grado\ de\ Confiabilidad + Tasa\ de\ error * \sum Fi)$$

$$PF = 435 * (0.65 + 0.01 * 45)$$

$$PF = 478.5$$

Si consideramos el máximo valor de ajuste de complejidad como $\sum Fi = 70$, se tiene:

$$PF_{max} = 435 * (0.65 + 0.01 * 70)$$

$$PF_{max} = 587.25$$

Entonces si $\sum Fi$ es considerada como el 100%, la relación obtenida entre los puntos será:

Así:

$$\frac{PF}{PF_{max}} = \frac{478.5}{587.25} = 0.81$$

Por lo tanto la funcionalidad del sistema es del 81 % tomando en cuenta el punto de función máximo.

Conversión de los Puntos de Función a KLDC

Ahora se debe convertir los Puntos de Función a miles de líneas de código. Tomar en cuenta la siguiente tabla.

Tabla 4.5

LENGUAJE	NIVEL	FACTOR LDC/PF
C	2,5	128
Ansi Basic	5	64
Java	6	53
PL/I	4	80
Ansi Cobol 74	3	107
Visual Basic	7	46
ASP	9	36
PHP	11	29
Visual C+ +	9,5	34

Título: Tabla de Conversión de Puntos de Función a KLDC

Fuente: [JONE,S96]

$$LDC = PF * \frac{\text{Factor LCD}}{PF}$$

$$LDC = (478.5 * 34)$$

$$LDC = 16269$$

$$KLDC = \frac{16269}{1000} = 16.27$$

b) CONFIABILIDAD

Conjunto de atributos que se relacionan con la capacidad del software de mantener su nivel de performance bajo las condiciones establecidas por un período de tiempo.

Madurez: atributos que se relacionan con la frecuencia de fallas por defectos en el software.

Tolerancia a las fallas: atributos que miden la habilidad de mantener el nivel especificado de performance en caso de fallas del software.

Recuperación: atributos que miden la capacidad de restablecer el nivel de performance y recuperar datos en caso de falla, y el tiempo y esfuerzo necesario para ello.

Cumplimiento: Capacidad del producto software para adherirse a normas, convenciones o regulaciones relacionadas con la funcionalidad.

Para determinar la confiabilidad del sistema especificamos el instante en el que este comienza a funcionar, determinado por

$$t_0 = 0$$

A partir de este momento se observa el trabajo del sistema hasta que se produzca una falla en el instante t que se va aproximando a una variable aleatoria continua, que nos determina la confiabilidad en términos probabilísticos.

Entonces tenemos las siguientes probabilidades:

$$P(T \leq t) = F(t) \quad (1): \text{Probabilidad de fallas.}$$

$$P(T > t) = 1 - F(t) \quad (2): \text{Probabilidad de trabajo sin fallas.}$$

Debido a que se tiene tiempos de inicio como de fin, para el cálculo de estas probabilidades se utiliza la distribución exponencial. Entonces la función $F(t)$ está dada por:

$$F(t) = 0.81 e^{-\lambda t} \quad \text{Ecuación de confiabilidad:}$$

Para el cálculo de error se consideran 10 ejecuciones en un mes en un periodo de 12 meses para obtener una probabilidad de fallas. En este caso, el margen de error λ será de $\lambda = \frac{1}{10}$. Además se utiliza el resultado obtenido del punto función igual a 0,81 para la distribución en un periodo $t = 12$ meses. Obteniendo la probabilidad de fallas se tiene:

$$P(T \leq t) = F(t)$$

$$F(t) = 0.81 * e^{-(1/10) \cdot 12}$$

$$F(t) = 0.24 \quad \text{Probabilidad de fallo}$$

Calculando el valor de la probabilidad de que el sistema trabaje sin fallos a partir del anterior resultado tenemos que: El software tendrá una confiabilidad de 81 % en una gestión, lo cual indica que se mantendrá estable y en funcionamiento.

c) USABILIDAD

El estándar ISO9126 define la usabilidad como:

“La capacidad de un producto de software de facilitar a usuarios específicos alcanzar metas específicas con eficacia, productividad, seguridad y satisfacción en un contexto específico de uso”. Añade que “Calidad en uso es la visión de calidad de los usuarios de un ambiente conteniendo software, y es medida sobre los resultados de usar el software en el ambiente, antes que sobre las propiedades del software en sí mismo”.

El sistema de control y seguimiento análisis de muestras de minerales en cuanto a la usabilidad es comprensible y dispone de la facilidad de uso para los usuarios destinados a los encargados del área de almacén, laboratorio y al mismo operador, esto debido a que durante la entrega de software se han venido haciendo las modificaciones necesarias a las peticiones del usuario en cuanto a la interfaz, al manejo del sistema, esto para facilitarle al usuario a alcanzar sus objetivos y metas específicas para realizar el control y seguimiento de muestras de minerales.

d) MANTENIBILIDAD

Para calcular la estabilidad del sistema, es decir índice de madurez del software (IMS), se establecerá los cambios que ocurrieron con cada versión del producto.

Para esto el Índice de Madures del Software se determina con la siguiente información:

MT= Numero de módulos en la versión actual.

Fc = Numero de módulos en la versión actual que se han cambiado

Fa = Numero de módulos en la versión actual que se han añadido

Fe = Numero de módulos en la versión actual que se han eliminado.

El índice de madurez del software se calcula de la siguiente manera:

$$IMS = \frac{[MT - (Fc + Fa + Fe)]}{MT}$$

En el sistema se obtuvieron los siguientes valores como muestra la tabla 4.6 para la información requerida para el IMS:

Tabla 4.6

Información	Valores Obtenidos
MT	5
Fc	0
Fa	1
Fe	1

Título: Calculo del IMS

Fuente: [Elaboración propia]

Ahora calculamos el índice de madurez del software sustituyendo los valores de la tabla los cuales son resultados obtenidos del sistema.

$$IMS = \frac{5 - (0 + 1 + 1)}{5}$$

$$IMS = 0.60$$

Por lo tanto tomando en cuenta la escala siguiente se concluye que el IMS obtenido tiene una estabilidad alta.

Tabla 4.7

75% <= IMS <=100%	Optima
50% <= IMS <=75%	Buena
25% <= IMS <=50%	Suficiente
0% <= IMS <=25%	Deficiente

Título: Escala de aceptación

Fuente: [Elaboración propia]

e) PORTABILIDAD

Mediante la métrica de facilidad de instalación se calcula el factor de portabilidad, mediante este se obtiene el porcentaje de éxitos de instalación de los usuarios responsables de realizar la instalación del sistema.

Facilidad de instalación

La facilidad de instalación viene dada de la siguiente relación:

$$X = \frac{A}{B}$$

A = Número de casos de éxitos de la operación instalación por parte del usuario.

B = Número total de operaciones de instalación que realizó el usuario.

Entonces se tiene que:

$$X = \frac{8}{10} = 0.8$$

$$X = 0.8$$

Por tanto existe un 80% de probabilidad que el usuario instale exitosamente el sistema y de acuerdo a la escala de aceptación mostrada en la tabla 4.7 que se muestra en la parte superior se concluye que el sistema es portable en un 80%.

Ahora haciendo una sumatoria total de cada característica del estándar ISO 9126, se considera que el sistema tiene una funcionabilidad del 74%, y se concluye que el sistema es de buena calidad.



Capítulo V

Costo Beneficio

5.1 INTRODUCCIÓN

Como se conoce, una de las tareas de mayor importancia en la planificación de proyectos de software es la estimación, la cual consiste en determinar, con cierto grado de certeza, los recursos de hardware y software, costo, tiempo y esfuerzo necesarios para el desarrollo de los mismos.

La estimación de costos de software tiene dos usos en la administración de proyectos:

- Durante la etapa de planeamiento: Permite decidir cuantas personas son necesarias para llevar a cabo el proyecto y establecer el cronograma adecuado.
- Para controlar el progreso del proyecto: Es esencial evaluar si el proyecto está evolucionando de acuerdo al cronograma y tomar las acciones correctivas si fuera necesario. Para esto se requiere contar con métricas que permitan medir el nivel de cumplimiento del desarrollo del software.

En el ámbito de la ingeniería de software, la estimación de costos radica básicamente en estimar la cantidad de personas necesarias para desarrollar el producto. A diferencia de otras disciplinas de la ingeniería, en las cuales, el costo de los materiales es el principal componente a ser estimado.

La estimación de costos de software posibilita relacionar conceptos generales y técnicas del análisis económico en el mundo particular de la ingeniería de software. Aunque no es una ciencia exacta no podemos prescindir de ella puesto que hoy en día un error en las predicciones puede conducir a resultados adversos.

Es importante reconocer la fuerte relación entre costo, cronograma y calidad. Estos tres aspectos están íntimamente relacionados y confrontados entre sí. De esta manera, se hace difícil incrementar la calidad sin aumentar el costo y/o el cronograma del software a desarrollar.

Similarmente, el cronograma de desarrollo no puede reducirse dramáticamente sin deteriorar la calidad del producto de software y/o incrementar el costo de desarrollo.

Muchas organizaciones desarrollan sistemas simplemente para satisfacer requerimientos obligatorios de gobierno (por ejemplo, sistemas de reporte para oficinas de trabajo, o nuevos sistemas para manejar la cambiante legislación de impuestos). Desde luego, incluso en estos casos cuando no existen beneficios derivables del sistema (más que el lujo de evitarse multas o poder permanecer en el negocio), la administración usualmente desea saber cuál será el costo del sistema; pero esto se debe poder realizar como parte de las actividades de estimación.

Como analista no está en condición de insistir en que se realice un estudio de costo/beneficio; después de todo, es el sistema del usuario y si él quiere construirlo sin justificación alguna, es su decisión. Sin embargo es buena idea si se ha hecho un estudio de costo /beneficio para el proyecto y, de ser así, si es razonable. Si no lo hay, o si los beneficios son muy difusos (o si se subestimaron exageradamente los costos), debe estar consistente de que el proyecto es vulnerable.

5.2. ANÁLISIS DE COSTO Y BENEFICIO

El método de Análisis Costo Beneficio está basado en la razón de los beneficios a los costos asociada con un proyecto en particular. Se considera que un proyecto es atractivo, cuando los beneficios derivados de su implementación exceden sus costos asociados. Por tanto, el primer paso en un análisis de costo/beneficio es determinar cuáles de los elementos son beneficios y costos.

Para la identificación de los costos y beneficios del proyecto que son convenientes para su evaluación, es necesario definir una situación base o situación sin proyecto; la comparación de lo que sucede con proyecto versus lo que hubiera sucedido sin proyecto, definirá los costos y beneficios del mismo.

¿QUÉ ES?

El análisis costo/beneficio es el proceso de colocar cifras en dólares en los diferentes costos y beneficios de una actividad. Al utilizarlo, podemos estimar el impacto financiero acumulado de lo que queremos lograr.

¿CUÁNDO SE UTILIZA?

Se debe utilizar el análisis costo/beneficio al comparas los costos y beneficios de las diferentes decisiones. Un análisis de costo/beneficio por si solo puede no ser una guía clara para tomar una buena decisión. Existen otros puntos que deben ser tomados en cuenta por ejemplo: la moral de los empleados, seguridad, obligaciones legales y satisfacción del cliente.

¿CÓMO SE UTILIZA?

El análisis de costo beneficio involucra los siguientes seis pasos:

- Reunir todos los datos provenientes de factores importantes relacionados con cada una de sus decisiones.
- Determinar los costos relacionados con cada factor. Algunos costos como la mano de obra, serán exactos mientras que otros serán estimados.
- Sumar los costos totales para cada decisión propuesta.
- Determinar los benéficos en dólares para cada decisión.

- Poner las cifras de los costos y benéficos totales de manera que haya una relación donde los benéficos son el numerador y lo costos el denominador.
- Compara las relaciones beneficios a costos para las diferentes decisiones.

Para determinar el costo total del proyecto se tomará en cuenta los siguientes costos:

- Costo del software desarrollado.
- Costo de la Implementación del sistema.
- Costo de la Elaboración del Proyecto.

5.3 MÉTODOS PARA EL ANÁLISIS COSTOS/BENEFICIOS

Diferentes métodos pueden ser utilizados para calcular la relación costo/beneficio. Los métodos más sofisticados consideran el tiempo-valor del dinero como parte del análisis costo/beneficio. El tiempo-valor del dinero conocido también como el factor de descuento, es simplemente un método utilizado para convertir el valor futuro del dinero en valor presente (dólares futuros a dólares presentes). Se basa sobre la premisa de que el dólar de hoy tiene más valor que un dólar en unos años en el futuro debido a los intereses o a la ganancia que se pueda obtener. Incluir el tiempo-valor del dinero puede ser crucial para la salud financiera de una organización, ya que los esfuerzos por mejorar pueden requerir de compromisos de capital por un periodo de tiempo prolongado.

Los métodos comunes para el análisis del costo /beneficio incluyen:

- Punto de equilibrio
- Periodo de devolución
- Valor presente neto (VAN)
- Tasa interna de retorno (TIR)

Uno de los métodos más comunes para el cálculo de costos es el Cocomo, es por esta razón que emplearemos este modelo.

5.3.1 MODELO COCOMO

Entre los distintos métodos de estimación de costes de desarrollo de software, el modelo COCOMO (CONstructive COst MOdel) desarrollado por Barry M. Boehm, se engloba en el grupo de los modelos algorítmicos que tratan de establecer una relación matemática la cual permite estimar el esfuerzo y tiempo requerido para desarrollar un producto.

Por un lado COCOMO define tres modos de desarrollo o tipos de proyectos:

- Orgánico: proyectos relativamente sencillos, menores de 50 KDLC líneas de código, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables.
- Semi-acoplado: proyectos intermedios en complejidad y tamaño (menores de 300 KDLC), donde la experiencia en este tipo de proyectos es variable, y las restricciones intermedias.
- Empotrado: proyectos bastantes complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica. Además se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

Y por otro lado existen diferentes modelos que define COCOMO:

- Modelo básico: Se basa exclusivamente en el tamaño expresado en LDC.
- Modelo intermedio: Además del tamaño del programa incluye un conjunto de medidas subjetivas llamadas conductores de costes.

- Modelo avanzado: Incluye todo lo del modelo intermedio además del impacto de cada conductor de coste en las distintas fases de desarrollo.

Para nuestro caso el modelo intermedio será el que usaremos, dado que realiza las estimaciones con bastante precisión.

1) COSTO DEL SOFTWARE

Entonces la formulas a utilizar serán las siguientes:

$$E = a_b(KLDC)^{b_b} * FAE$$

$$T = c_b(E)^{d_b}$$

Donde:

E: Esfuerzo aplicado en personas por mes.

T: Tiempo de desarrollo en meses cronológicos

KLDC: Número estimado de línea de código distribuidas (en miles)

Y tomando en cuenta, que en el anterior capitulo de calidad de software ya obtuvimos algunos resultados como las líneas de código KLDC=16.27, ahora pasamos a calcular los valores restantes.

Tabla 5.1

Proyecto de Software	a_b	b_b	c_b	d_b
Orgánico	2,4	1,05	2,5	0,38
Semi - acoplado	3,0	1,12	2,5	0,35
Empotrado	3,6	1,20	2,5	0,32

Título: Coeficientes a_b , b_b y exponentes c_b y d_b

[Fuente: Pressman, 2002]

En la tabla 5.1 se muestra los tipos de proyectos de software; del cual para nuestro caso elegiremos el proyecto de software orgánico, ya que las líneas de código obtenidos son menores a 50 KLDC.

Y por otro lado también debemos hallar la variable FAE(Factor de Ajuste del Esfuerzo), la cual se obtiene mediante la multiplicación de los valores evaluados en los diferentes 15 conductores de coste que se observan en la siguiente tabla:

Tabla 5.2

Conductores de coste	Valoración					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de Software						
Fiabilidad requerida del software	0,75	0,88	0	1,15	1,4	0
Tamaño de la base de datos	0	0,94	1	1,08	1,16	0
Complejidad del producto	0,7	0,85	1	1,15	1,3	1,65
Atributos del Hardware						
Restricciones del tiempo de ejecución	0	0	1	1,11	1,3	1,66
Restricciones del almacenamiento principal	0	0	1	1,06	1,21	1,56
Volatilidad de la máquina virtual	0	0,87	1	1,15	1,3	0
Tiempo de respuesta del ordenador	0	0,87	1	1,07	1,15	0
Atributos de personal						
Capacidad del analista	1,46	1,19	1	0,86	0,71	0
Experiencia en la aplicación	1,29	1,13	1	0,91	0,82	0
Capacidad de los programadores	1,42	1,17	1	0,86	0,7	0
Experiencia en S.O. utilizado	1,21	1,1	1	0,9	0	0
Experiencia en el lenguaje de programación	1,14	1,07	1	0,95	0	0
Atributos del proyecto						
Prácticas de programación modernas	1,24	1,1	1	0,91	0,82	0
Utilización de herramientas software	1,24	1,1	1	0,91	0,83	0
Limitaciones de planificación del proyecto	1,23	1,08	1	1,04	1,1	0

Título: Coeficientes a_b , b_b y exponentes c_b y d_b

[Fuente: Introducción al COCOMO]

$$FAE = 1.15 * 1.08 * 1 * 1.11 * 1.06 * 1 * 1.07 * 0.86 * 0.91 * 0.86 * 1 * 0.95 * 0.91 * 0.91 * 1$$
$$FAE = 0.827901944$$

Justificación de los valores

Atributos de software

- Fiabilidad requerida del software: Si se produce un fallo por el pago de un pedido, o fallo en alguna reserva, etc. Puede ocasionar grandes pérdidas a la empresa (Valoración Alta).
- Tamaño de la base de datos: La base de datos de nuestro sistema contiene varios datos. (Valoración Alta).
- Complejidad del producto: La aplicación va a realizar varios cálculos y almacenamientos de datos (Valoración Nominal).

Atributos de hardware

- Restricciones del tiempo de ejecución: En los requerimientos se exige alto rendimiento (Valoración Alta).
- Restricciones del almacenamiento principal: Porque existen pocas restricciones para el uso del sistema (Valoración Alta).
- Volatilidad de la máquina virtual: Se usarán sistemas de la "Familia Windows" (Valoración Nominal).
- Tiempo de respuesta del ordenador: Deberá ser interactivo con el usuario (Valoración Alta).

Atributos del personal

- Capacidad del analista: Capacidad alta relativamente, debido a la experiencia en análisis en proyecto similar (Valoración Alta)
- Experiencia en la aplicación: Se tiene cierta experiencia en aplicaciones de esta envergadura (Valoración alta).
- Capacidad de los programadores: Teóricamente deberá tenerse una capacidad alta por la experiencia en anteriores proyectos similares (Valoración alta).
- Experiencia en S.O. utilizado: Con Windows XP la experiencia es a nivel usuario (Valoración Nominal).
- Experiencia en el lenguaje de programación: Es relativamente alta, dado que se controlan las nociones básicas y las propias del proyecto (Valoración Alta).

Atributos del proyecto

- Prácticas de programación modernas: Se usarán prácticas de programación mayormente convencional (Valoración Alta).
- Utilización de herramientas software: Se usarán herramientas estándar que no exigirán apenas formación, de las cuales se tiene cierta experiencia (Valoración Alta).
- Limitaciones de planificación del proyecto: Existen ciertos límites de planificación. (Valoración Nominal).

Calculo del esfuerzo de desarrollo

$$E = 3.2 (16.27)^{1.05} * 0.827901944$$

$$E = 15.91 \text{ personas/mes}$$

Calculo del tiempo de desarrollo

$$T = 2.5(15.91)^{0.38}$$

$$T = 7.15 \text{ meses}$$

Productividad

$$PR = LDC/\text{Esfuerzo}$$

$$PR = \frac{1627 \text{ LDC}}{15,91}$$

$$PR = 102.26 \text{ LDC/personas mes}$$

Personal promedio:

$$P = E/T$$

$$P = \frac{15,91}{7,15} = 2,22 \text{ personas}$$

Según estas cifras será necesario un equipo de 2 personas trabajando alrededor de 7 meses, pero puesto que el desarrollo del proyecto debe realizarse en un plazo de 6 meses, incrementaremos a 3 personas el número de personas del equipo de proyecto.

Así pues tendremos un equipo formado por 1 Jefe de proyecto, 1 Analista y 1 programador. El salario de un programador aproximadamente oscila entre los 3000 Bs, cifra que tomaremos en cuenta para la estimación siguiente:

Costo del software = Numero de programadores * Salario de un programador

$$\text{Costo del software} = 3 * 3000 = 9000 \text{ Bs.}$$

2) COSTO DE IMPLEMENTACIÓN

Tabla 5.3

Descripción	Costo Total (Bs)
Instalación Visual Basic.Net 2008	80
Instalación SQL Server	105
Total	185

Titulo: Costo de implementación

[Fuente: Elaboración propia]

En la tabla se muestran los gastos de instalación de los programas requeridos para el desarrollo del sistema.

3) COSTO DE ELABORACIÓN DEL PROYECTO

Tabla 5.4

Descripción	Costo Total (Bs)
Análisis y diseño del proyecto	700
Bibliografía	70
Material de escritorio	150
Investigación en Internet	75
Otros	80
Total	1075

Titulo: Costo de elaboración del proyecto

[Fuente: Elaboración propia]

Los costos de elaboración del proyecto se refieren a los costos del estudio del sistema, en la etapa de recopilación y análisis.

4) COSTO TOTAL

Tabla 5.5

Descripción	Costo Total (Bs)
Costo del Software	9000
Costo de implementación	185
Costo de elaboración del proyecto	1075
Total	10260

Titulo: Costo Total

[Fuente: Elaboración propia]

Así tenemos que el costo total de todo el desarrollo del proyecto es de Bs 10260, costo que se hace aceptable al beneficio de la empresa.

5.3.2 VALOR ACTUAL NETO (VAN)

El Valor actual neto también conocido como valor actualizado neto, cuyo acrónimo es VAN (en inglés NPV), es un procedimiento que permite calcular el valor presente de un determinado número de flujos de caja futuros, originados por una inversión. La metodología consiste en descontar al momento actual (es decir, actualizar mediante una tasa) todos los flujos de caja futuros del proyecto. A este valor se le resta la inversión inicial, de tal modo que el valor obtenido es el valor actual neto del proyecto.

El método de valor presente es uno de los criterios económicos más ampliamente utilizados en la evaluación de proyectos de inversión. Consiste en determinar la equivalencia en el tiempo 0 de los flujos de efectivo futuros que genera un proyecto y comparar esta equivalencia con el desembolso inicial. Cuando dicha equivalencia es mayor que el desembolso inicial, entonces, es recomendable que el proyecto sea aceptado.

La fórmula que nos permite calcular el Valor Actual Neto es:

$$VAN = \sum_{t=1}^n \frac{V_t}{(1+k)^t} - I_0$$

V_t : representa los flujos de caja en cada periodo t.

I_0 : es el valor del desembolso inicial de la inversión.

n : es el número de períodos considerado.

El tipo de interés es k. Si el proyecto no tiene riesgo, se tomará como referencia el tipo de la renta fija, de tal manera que con el VAN se estimará si la inversión es mejor que invertir en algo seguro, sin riesgo específico. En otros casos, se utilizará el coste de oportunidad.

Cuando el VAN toma un valor igual a 0, k pasa a llamarse TIR (tasa interna de retorno). La TIR es la rentabilidad que nos está proporcionando el proyecto.

Tabla 5.6

Valor	Significado	Decisión a tomar
$VAN > 0$	La inversión produciría ganancias por encima de la rentabilidad exigida(r).	El proyecto puede aceptarse.

VAN < 0	La inversión produciría ganancias por debajo de la rentabilidad exigida(r).	El proyecto debería rechazarse.
VAN = 0	La inversión no produciría ni ganancias ni pérdidas.	Dado que el proyecto no agrega valor monetario por encima de la rentabilidad exigida (r), la decisión debería basarse en otros criterios, como la obtención de un mejor posicionamiento en el mercado u otros factores.

Titulo: Valores a considerar del VAN

[Fuente: VAN]

El valor actual neto es muy importante para la valoración de inversiones en activos fijos, a pesar de sus limitaciones en considerar circunstancias imprevistas o excepcionales de mercado. Si su valor es mayor a cero, el proyecto es rentable, considerándose el valor mínimo de rendimiento para la inversión.

Una empresa suele comparar diferentes alternativas para comprobar si un proyecto le conviene o no. Normalmente la alternativa con el VAN más alto suele ser la mejor para la entidad; pero no siempre tiene que ser así. Hay ocasiones en las que una empresa elige un proyecto con un VAN más bajo debido a diversas razones como podrían ser la imagen que le aportará a la empresa, por motivos estratégicos u otros motivos que en ese momento interesen a dicha entidad.

Puede considerarse también la interpretación del VAN, en función de la Creación de Valor para la Empresa:

- Si el VAN de un proyecto es Positivo, el proyecto Crea Valor.
- Si el VAN de un proyecto es Negativo, el proyecto Destruye Valor.
- Si el VAN de un proyecto es Cero, el Proyecto No Crea ni Destruye Valor.
-

Ahora calculamos el VAN:

$$VAN = \sum \frac{Ganancias}{(1+i)^n} - \sum \frac{Costos}{(1+i)^n}$$

$$VAN = \sum \frac{20000}{(1+12\%)^5} - \sum \frac{10260}{(1+12\%)^5}$$

$$VAN = 239.386$$

Entonces:

$$C/B = \frac{\sum \text{Ganancias}}{\sum \text{Costos}}$$

$$C/B = \frac{20000}{10260}$$

$$C/B = 1.94$$

De este resultado podemos concluir que, en la realización de este proyecto; por cada boliviano invertido ganamos 94 Bs.

5.3.3 TASA INTERNA DE RETORNO (TIR)

La tasa interna de retorno o tasa interna de rentabilidad (TIR) de una inversión, está definida como el promedio geométrico de los rendimientos futuros esperados de dicha inversión, y que implica por cierto el supuesto de una oportunidad para "reinvertir". En términos simples en tanto, diversos autores la conceptualizan como la tasa de interés (o la tasa de descuento) con la cual el valor actual neto o valor presente neto (VAN o VPN) es igual a cero. El VAN o VPN es calculado a partir del flujo de caja anual, trasladando todas las cantidades futuras al presente. Es un indicador de la rentabilidad de un proyecto, a mayor TIR, mayor rentabilidad.

Se utiliza para decidir sobre la aceptación o rechazo de un proyecto de inversión. Para ello, la TIR se compara con una tasa mínima o tasa de corte, el coste de oportunidad de la inversión (si la inversión no tiene riesgo, el coste de oportunidad utilizado para comparar la TIR será la tasa de rentabilidad libre de riesgo). Si la tasa de rendimiento del proyecto expresada por la TIR- supera la tasa de corte, se acepta la inversión; en caso contrario, se rechaza.

$$TIR = \sum \frac{\text{Ganancias} - \text{Costos}}{(1 + x)^n}$$

$$TIR = \sum \frac{20000 - 10260}{(1 + 3\%)^5}$$

$$TIR = 20350$$

Por lo tanto la tasa de rentabilidad que nos está proporcionando el proyecto es de Bs 20350.

CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

Terminando el proceso de construcción del sistema, se concluye que:

- A la culminación de la elaboración, análisis, diseño e implementación, se concluyó satisfactoriamente, llegando a los objetivos planteados.
- Se obtuvo un producto con las particularidades de un sistema de información, que contribuye como un aporte tecnológico. El sistema ha logrado adaptarse satisfactoriamente a los requerimientos de la Empresa.
- La automatización de la información, facilita la organización de la misma, agilizando los procesos de control de personal de toda la empresa. En especial en el control de muestras de mineral que ingresan en almacén ya que anteriormente se realizaba el registro en planillas lo que demora más o menos en un rango de 3 a 5 días.
- Con la implementación del Sistema de Información “Control y Seguimiento a análisis de muestras de minerales se logro cumplir con el objetivo general que es el de “Desarrollar e implementar el Modulo de Control y Seguimiento de las muestras

Los requisitos del software son la base de las medidas de calidad. La falta de concordancia con los requisitos es una falta de calidad, los estándares o metodologías definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se sigue ninguna metodología siempre habrá falta de calidad.

6.2. RECOMENDACIONES

- El sistema ha tomado en cuenta el área de almacén y laboratorio pero no el área contable ni el área de presidencia se recomienda diseñar los módulos faltantes.
- Se recomienda desarrollar un sistema en Windows Forms (Aplicación de Escritorio) como un añadido con sus respectivas Base de Datos Centralizada que se ejecute en cada uno de las pc's , esto para subsanar la posible falla de conexión con el proyecto, haciendo todo esto posible mediante el uso del concepto de Aplicaciones distribuidas.
- En cuanto al sistema desarrollado se recomienda realizar la capacitación al usuario respectivo encargado del manejo del sistema, esto para el buen manejo de la información y del propio sistema. Además es recomendable realizar una evaluación constante con el fin de garantizar la fiabilidad de la misma, realizar un mantenimiento preventivo y de adaptación del sistema con el fin de mejoras funcionales y adaptación del software a su entorno.

BIBLIOGRAFÍA

REFERENCIAS BIBLIOGRÁFICAS

- [PRESSMAN, 2005]

Pressman, R.: "Ingeniería De Software", 2005, 5ta. Edición, Editorial. McGraw Hill.

- [PRESSMAN,2000]

Roger S. Pressman, R. 2000: INGENIERIA DEL SOFTWARE UN ENFOQUE PRACTICO, Qusnta Edición, Editorial mcgraw-HILL, México.

- [LARMAN, 1999]

Larman Craig.: "UML y Patrones", 1999 Primera Edición, Editorial. Prentice Hall, México 1999.

- [OMG, 1999]

Object Management Group. 1999. OMG Unified Modeling Language Specification (Draft). Versión 1.3. alfa R5, marzo de 1999. Disponible en: <http://www.rational.com/uml>.

- [BORUMJA, 2000]

Jacobson, B., R.: "El Proceso Unificado de Desarrollo de Software", 2000, Segunda Edición, Editorial. Pearson Educación S.A./Madrid 2000.

- [MSUML, 2006]

Modelado de Sistemas con UML. Extraído el 3 de Octubre de 2006.

- [CGAP, 2007]

"Sistema integrado de control y administración de personal ", Biblioteca de la carrera de informática.

REFERENCIAS WEB

- <http://es.tldp.org/Tutoriales/doc-modelado-sistemas-UML/multiple-html/index.html>.
- <http://www.slideshare.net/msch/utilizando-metodologia-rup-parte1>.
- <http://www.monografias.com/trabajos59/calidad-software/calidad-software.shtml>.



INDICE DE FIGURAS

Figura 2.1: Actividades de un sistema de información.

Figura 2.2: Tipos de sistemas de información.

Figura 2.3: Ciclo de vida del RUP.

Figura 2.4: Diagrama UML.

Figura 2.5: Diagrama de casos de uso.

Figura 2.6: Arquitectura del uml.

Figura 3.1: Caso de uso del negocio.

Figura 3.2: Diagrama de caso de uso del sistema.

Figura 3.3: Controlar acceso al sistema.

Figura 3.4: Registrar solicitud de análisis

Figura 3.5: Solicitud de análisis.

Figura 3.6: Listado de muestras.

Figura 3.7: Registro de la muestra en laboratorio.

Figura 3.8: Registro de datos de la muestra analizada.

Figura 3.9: Elaboración de factura.

Figura 3.10: Diagrama de secuencia: Controlar acceso al sistema

Figura 3.11: Diagrama de secuencia: Registro de usuarios.

Figura 3.12: Diagrama de secuencia: Verificar usuario

Figura 3.13: Diagrama de secuencia: Solicitud de análisis

Figura 3.14 Diagrama de secuencia: Obtener reportes

Figura 3.15: Diagrama de secuencia: Imprimir factura

Figura 3.16: Diagrama de componentes del sistema de control

Figura 3.17: Diagrama de componentes del sistema; Gestión de captura

Figura 3.18: Diagrama de componentes del sistema: Gestión de reportes

Figura 3.19: Diagrama de clases del sistema

Figura 3.20: Modelo Entidad-relación

Figura 3.21: Formulario de ingreso al sistema

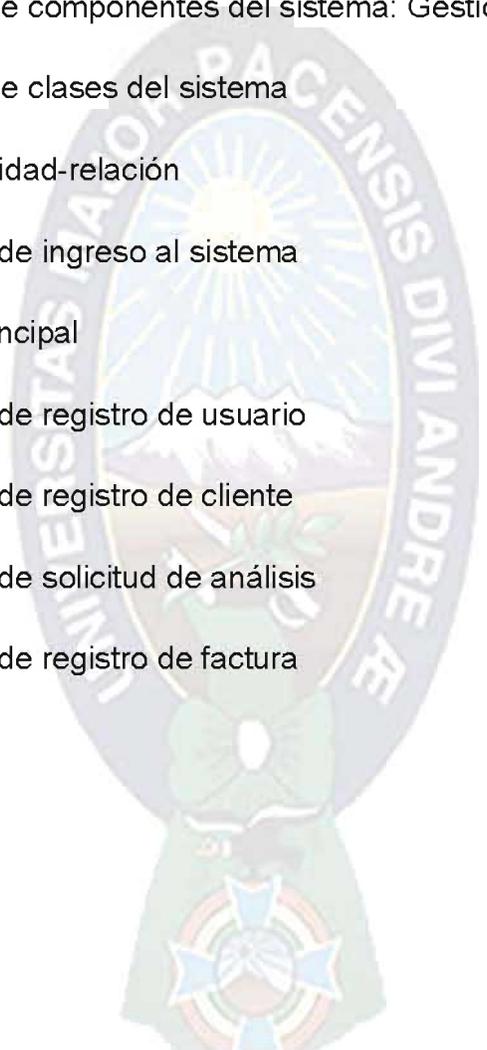
Figura 3.22: Pantalla principal

Figura 3.23: Formulario de registro de usuario

Figura 3.24: Formulario de registro de cliente

Figura 3.25: Formulario de solicitud de análisis

Figura 3.26: Formulario de registro de factura



INDICE DE TABLAS

Tabla 2.1: Coeficientes del proyecto software.

Tabla 3.1: Actor (Operador) y su descripción.

Tabla 3.2: Actor (Cliente) y su descripción.

Tabla 3.3: Actor (Encargado de almacén) y su descripción.

Tabla 3.4: Actor (Encargado del laboratorio) y su descripción.

Tabla 3.5: Caso de uso Controlar acceso al sistema.

Tabla 3.6: Control de existencias.

Tabla 3.7: Caso de uso Solicita análisis.

Tabla 3.8: Caso de uso Ingreso de muestras a almacén.

Tabla 3.9: Caso de uso Salida de muestras de almacén.

Tabla 3.10 Caso de uso Ingreso de muestras a laboratorio

Tabla 3.11: Caso de uso Entrega de resultados de análisis

Tabla 3.12 Caso de uso Realiza análisis

Tabla 3.13 Caso de uso Entrega de factura

Tabla 3.14: Tabla de requisitos del sistema.

Tabla 3.15: Tabla de requisitos del software.

Tabla 3.16: Tabla de requisitos del hardware.

Tabla 3.17: Tabla de requerimientos de interfaz de usuario.

Tabla 3.18: Descripción de caso de uso Controlar acceso al sistema

Tabla 3.19: Descripción de caso de uso Solicita análisis

Tabla 3.20: Descripción del caso de uso: Registro de ingreso de muestras a almacén.

Tabla 3.21: Descripción del caso de uso: Registro de salida de muestras a laboratorio.

Tabla 3.22: Descripción del caso de uso: Control de muestras

Tabla 3.23: Descripción de caso de uso: Obtener reportes

Tabla 3.24: Descripción de caso de uso: Registro de muestras analizadas

Tabla 3.25 Descripción de caso de uso: Elaboración de facturas

Tabla 3.26: Contrato de operaciones para: Ingresar usuario

Tabla 3.27: Contrato de operación para verificar usuarios

Tabla 3.28: Contrato de operaciones para: Obtener reportes

Tabla 4.1: Parámetros de medición.

Tabla 4.2: Calculo de métricas de punto función

Tabla 4.3: Valores de ajuste de complejidad.

Tabla 4.4: Tabla de preguntas del punto función.

Tabla 4.5: Tabla de Conversión de Puntos de Función a KLDC.

Tabla 4.6: Calculo del IMS.

Tabla 4.7: Escala de aceptación.

Tabla 5.1: Coeficientes a_b , b_b y exponentes c_b y d_b .

Tabla 5.2: Coeficientes a_b , b_b y exponentes c_b y d_b .

Tabla 5.3: Costo de implementación.

Tabla 5.4: Costo de elaboración del proyecto.

Tabla 5.5: Costo Total.