UNIVERSIDAD MAYOR DE SAN ANDRÉS

FACULTAD DE CIENCIAS PURAS Y NATURALES

CARRERA DE INFORMATICA

PROYECTO DE GRADO

"SISTEMA INFORMATICO DE CONTROL DE ALMACENES, MAQUINARIAS Y PERSONAL"

CASO: EMPRESA CONSTRUCTORA "INGLOBOL S.A."

PARA OPTAR AL TITULO DE LICENCIATURA EN INFORMATICA

MENCION: INGENIERIA DE SISTEMAS INFORMATICOS

POSTULANTE: JUAN CARLOS MANRIQUEZ CONDORI

TUTOR METODOLOGICO: Lic. GROVER ALEX RODRÍGUEZ RAMÍREZ

ASESOR: Lic. JAVIER REYES PACHECO

LA PAZ-BOLIVIA

2012

AGRADECIMIENTOS

A Dios por permitirme existir y por no dejarme desfallecer en los momentos más difíciles.

A mis padres por el amor y la confianza que me brindan día a día. Gracias por su constante dedicación, por las muestras de cariño y por sobre todo por todo el apoyo y confianza que depositaron en mí.

A mis hermanos y hermanas por el apoyo y comprensión en cada momento y su comprensión por los momentos difíciles y por apoyarme en cada momento durante la elaboración de este proyecto.

Gracias a ti Amalia por todo tu apoyo tu comprensión y la tranquilidad que me brindas en los momentos más difíciles.

A la empresa INGLOBOL S.A. por haberme abierto las puertas para la realización de este proyecto, así como por su entera colaboración durante su ejecución, mil gracias a cada uno de los que integran esta gran familia.

A mis docentes, por haberme enseñado las herramientas necesarias para iniciar mi carrera.

DEDICADO A:

Gracias por haberme dado el mejor regalo que Dios pudo darme, mi familia

Gracias por sobre todo a ti mamá y a ti papá por el gran esfuerzo que hicieron al brindarme una educación y por el sacrificio que significa todo el trayecto que he recorrido y así lograr que llegara lejos, gracias por la confianza que siempre me brindaron, gracias por sobre todo a ustedes.

Gracias por la paciencia brindada por entenderme cuando tenía problemas o dificultades y por darme consejos acertados que me ayudan a crecer y ser una mejor persona.

Resumen

La empresa constructora INGLOBOL S.A. ha sido el objeto de estudio de este presente Proyecto de Grado. El cual busca la forma de optimizar el trabajo y maximizar las ganancias, con el fin de concluir un servicio de calidad.

Para que esto suceda, un factor fundamental es la administración de información, y la interacción entre procesos depende del manejo de los datos, que si es la adecuada dará culminación satisfactoria d los proyectos al igual que los objetivos y metas de la empresa, para alcanzar un grado de credibilidad y posicionamiento en el mercado.

Este Proyecto de grado titulado "Sistema Informático de Control de almacenes, maquinarias y de Personal" para la constructora INGLOBOL S.A. de ha generado en base a la necesidad de poder administrar la información de las obras civiles y recursos involucrados para su ejecución, en los diferentes campamentos abiertos. De tal forma que el sistema ayudara, facilitara y automatizara los procesos realizados, además evitara dificultades antes presentadas por su estructura automatizada, todo esto gracias al desarrollo del sistema, permitiendo el seguimiento y control de los procesos actuales que se deben seguir para efectuar el proyecto.

El Proyecto está desarrollado en PHP y Mysql con sus módulos de control de almacenes, maquinarias, agregados y personal para cada uno de los proyectos de la empresa INGLOBOL S.A.

El presente proyecto ha utilizado una metodología RUP para el análisis y diseño, y con el lenguaje UML para las notaciones de diseño, el cual se efectuó con la utilización de una herramienta case UML Enterprice Architect y el UMLDRAW.

INDICE GENERAL

INDICE DE GRAFICOS Y TABLAS

CAPITULO I

2. ANTECEDENTES GENERALES

1.4. INTRODUCCIÓN

El Proyecto de Grado titulado **"**Sistema Informático de Control de almacenes, maquinarias y de Personal", es una herramienta importante para la toma de decisiones además de ser un soporte eficaz e inmediato de los requerimientos de información en la gestión de la empresa constructora INGLOBOL S.A., mediante la aplicación y desarrollo de un sistema informático a partir de la información brindada por la empresa mediante una computadora; el mismo se brindara facilidades para la toma de decisiones cuando existe incertidumbre e n varios campos como ser , el costo de materiales, rendición de cuentas, control de personal, control de maquinarias, control de agregados, presupuestos, contribuyendo de esta manera a disminuir las divergencias entre informes y así mejorar la productividad de la empresa.

El sistema abarca los puntos mas importantes en la empresa para poder saber si un proyecto dado marcha bien o tropieza con dificultades en su desarrollo, por tanto la información precedente del sistema mostrara la situación real del proyecto, por tanto esto ayudara

mostrando un panorama general para poder encaminarse de acuerdo a las decisiones de la gerencia.

1.5. **ANTECEDENTES**

Hoy en día existen un sin número de aplicaciones en informática es decir en los campos empresariales, administrativos y comercial. Entre ellas cabe mencionar a las empresas constructoras que tienen en los equipos de computación una de sus principales herramientas para su normal desenvolvimiento diario de sus actividades puesto que no solo es para el registro de datos sino que también es parte importante en mejorar la productividad y así poder ser competitivos en cuanto al uso de tecnologías.

Sin embargo el simple uso de la computadora no cumple todas las necesidades, requerimientos y obligaciones que una empresa constructora exige por tanto se considera que la tecnología de la información puede darle a los procesos de la misma mediante un sistema automatizado resultados que les permita alcanzar ventajas competitivas.

El control de los estados financieros, presupuestos, análisis de costos unitarios, rendición de cuentas, control de personal, control de maquinarias, control de agregados es fundamental para ver en qué estado se encuentra la empresa, por esto un inadecuado control de estas variables afecta la buena toma de decisiones además de causar pérdidas de productividad a la empresa.

A continuación se detallan los principales registros que se realizan en la empresa que son inapropiados puesto que casi todos se manejan en hojas de papel y en el mejor de los casos se utiliza las hojas de cálculo de EXCEL.

- Registro de compras diarias no controladas(rendición de cuentas)
- Retardo en el seguimiento de asistencia de personal (técnico y obreros)
- Registro de agregados (material de construcción como ser arena fina, arena corriente, piedra bruta y grava).
- Registro de maquinarias (propias y alquiladas)
- Registro de materiales que ingresan en almacenes
- Retardo en el envió de solicitud de recursos económicos para el pago quincenal

Además existen otros problemas relacionados con la parte administrativa y organizativa de la empresa como ser:

Un director de obras que es el responsable de un proyecto no cuenta con información oportuna de los datos de sus obreros como es el caso de poder saber que obreros no cuentan con fotocopia de su carnet de identidad en los archivos de obreros.

Un encargado de compras no cuenta con los precios y lugares donde se pueden adquirir los mejores precios para hacer las compras respectivas.

Actualmente la empresa INGLOBOL S.A. realiza sus registros en hojas de cálculo EXCEL la cual no está integrada en sus distintas áreas necesarias para una buena productividad.

1.6. TRABAJOS SIMILARES

- Autor: Herrera Cadena, Yadranka

Tutor: Lic. Freddy Miguel Toledo Paz

Revisor: Lic. Brígida Carvajal Blanco

Título: Sistema Integrado de Gestión de Proyectos para la Constructora COINCIED S.R.L.

Caso: Empresa Constructora COINCIED S.R.L.

Año: 2009

Tesis: T-1751

- Autor: FelixHuayhua Miranda

Tutor: Lic. Nancy Orihuela Sequeiros

Revisor: Lic. AbdiasPatzi Choque

Título: Sistema de Información Integrado de Control Administrativo, Financiero y Proyectos

Caso: Empresa Constructora “ELEKTROPACHA”

Año: 2008

Tesis: T-1704

1.8. FORMULACIÓN DEL PROBLEMA

¿Cómo puede la empresa INGLOBOL S.A. para sus Proyectos Barrios de Verdad hacer que sus controles de personal, materiales en almacenes, compras, maquinarias y agregados sean agiles y de esta forma contar con información actualizada y precisa?

1.9. PLANTEAMIENTO DE OBJETIVOS

1.9.1. OBJETIVO GENERAL

Desarrollar un Sistema Informático Integrado para la empresa INGLOBOL S.A. en sus proyectos Barrios de Verdad para contar con controlesagilese información actualizada y precisa.

1.9.2. OBJETIVOS ESPECÍFICOS

- Diseñar e implementar un sistema integrado con los módulos para el control de almacenes, personal y maquinarias empleando Ingeniería de Software, con un diseño orientado a objetos.
- Diseñar una Base de datos relacional para el sistema automatizado.
- Realizar un entorno grafico amigable al usuario.

1.10. JUSTIFICACIÓN

1.10.1. JUSTIFICACIÓN TECNOLÓGICA

El uso de tecnología es una herramienta indispensable en los más diversos campos de la actividad humana, las constructoras actualmente tienen a la tecnología como una herramienta indispensable ya que requieren guardar información importante para el correcto funcionamiento de la empresa.

1.10.2. JUSTIFICACIÓN ECONÓMICA

La empresa INGLOBOL cuenta con equipos necesarios para implementar el sistema por tanto no es necesario la compra de un equipo en especial, el uso de software gratuito (PHP, MySQL) evitara el pago de Licencias.

No se requiere de una gran inversión para la implementación del sistema por tanto los beneficios que dará el sistema serán en beneficio de la empresa lográndose mayores controles en las actividades de sus proyectos evitándose las pérdidas en materiales y recursos destinados a dichos proyectos.

1.10.3. JUSTIFICACIÓN SOCIAL

El principal beneficiario es la clase obrera puesto mientras la empresa genere utilidades en su beneficio supondrá una correcta inversión en el desarrollo de mas proyectos por tanto se generaran nuevos empleos para mas personas como ser albañiles (ayudantes, contramaestres y maestros), contratistas, choferes, mecánicos, etc.

1.11. ALCANCES Y LIMITES

El presente proyecto tiene como alcance y limite la construcción de un sistema que comprende lo siguiente:

Modulo de almacenes

Los almacenes cuentan con una gran variedad de materiales los cuales serán controlados mediante ingresos y egresos de los mismos tomando en cuenta que están los materiales perecederos y los de larga duración.

Se registraran materiales como ser cemento, gaviones, herramientas (picotas, palas, combos, martillos, etc.), de los cuales los mas críticos en su control son los gaviones y cemento, se controlara la cantidad de cemento que se destina en los diferentes frentes de trabajo.

Modulo de Registro y seguimiento del personal

Un registro de obreros (almacenero, capataz, maestro, contramaestre y ayudante, choferes), personal técnico (Director de obra, residente de obra, administrador y topógrafo)

Queda fuera de los alcances del proyecto el mejorar la calidad de producción individual de los obreros y del personal técnico.

Modulo Agregados

Los agregados (arena fina, arena corriente, grava ¾, grava de 1, piedra manzana, piedra cortada, piedra bruta), resultan ser los de mayores gastos para la productividad de los proyectos por tanto se registraran los nombres de los proveedores, sus operadores, las marcas de las volquetas y los cubos por volqueta.

CAPITULO II

MARCO TEÓRICO

4. MARCO INSTITUCIONAL

4.1. DESCRIPCIÓN DE LA EMPRESA

Fundada el mes de marzo de 1984, Ingeniería Global Boliviana "INGLOBOL Ltda" como una firma jurídicamente establecida como Responsabilidad Limitada, de carácter privado, dedicada a la prestación de servicios en todos los campos de la Construcción civil y montajes industriales. En marzo de 2006 tiene un cambio en la conformación de los socios, y establece la prestación de servicios al campo de la construcción de edificios completos o parte de edificios; obras de ingeniería civil. Asimismo, la Junta de Socios de la empresa designa como Gerente General al Ing. Eduardo Gálvez Valdivia.

Para el mejor desenvolvimiento de las labores de sus técnicos, dispone de una oficina central ubicada en la zona norte de la ciudad de La Paz, la misma que está completamente equipada y dispone de un taller de mantenimiento para su equipo pesado en la ciudad de El Alto.

Los diversos sectores en los que se desarrolla su actividad y los certificados de los clientes avalan la experiencia recogida en los campos de: mantenimiento de carreteras, obras civiles, represas, sistemas de riego u otros hidráulicos, obras viales y, por supuesto, en todas las disciplinas auxiliares o complementarias a las anteriores.

4.1.1. MISIÓN Y VISIÓN

- **MISION DE LA EMPRESA:**

"Concebir y ejecutar proyectos de construcción y mantenimiento de obras viales y civiles de toda naturaleza y envergadura en el ámbito de la ingeniería civil".

- **VISION DE LA EMPRESA:**

"Constituirnos en una de las empresas de mayor prestigio y tamaño del país, reconocida por su eficacia y eficiencia y su alta responsabilidad técnica"

4.1.2. OBJETIVOS DE LA EMPRESA

Satisfacer plenamente a nuestros clientes, para ello asumimos el compromiso de:

- ✓ Cumplir con las especificaciones técnicas de las obras, los requisitos legales y otros que la organización suscriba.
- ✓ Ejecutar las obras dentro de los plazos establecidos y a un costo que asegure la competitividad de la empresa.

Ser sinónimo de prestigio y excelencia, para ello nos comprometemos a:

- ✓ Mejorar continuamente nuestros productos, servicios y procesos.
- ✓ Emplear eficientemente nuestros recursos humanos, físicos y económicos.
- ✓ Desarrollar nuestra competencia personal y mantener nuestra motivación con base en el reconocimiento de un trabajo bien realizado.
- ✓ Promover y convivir en un ambiente de trabajo con base en los principios de lealtad y honradez.
- ✓ Colaborar con los proveedores y subcontratistas.

4.1.3. ORGANIGRAMA DE LA EMPRESA

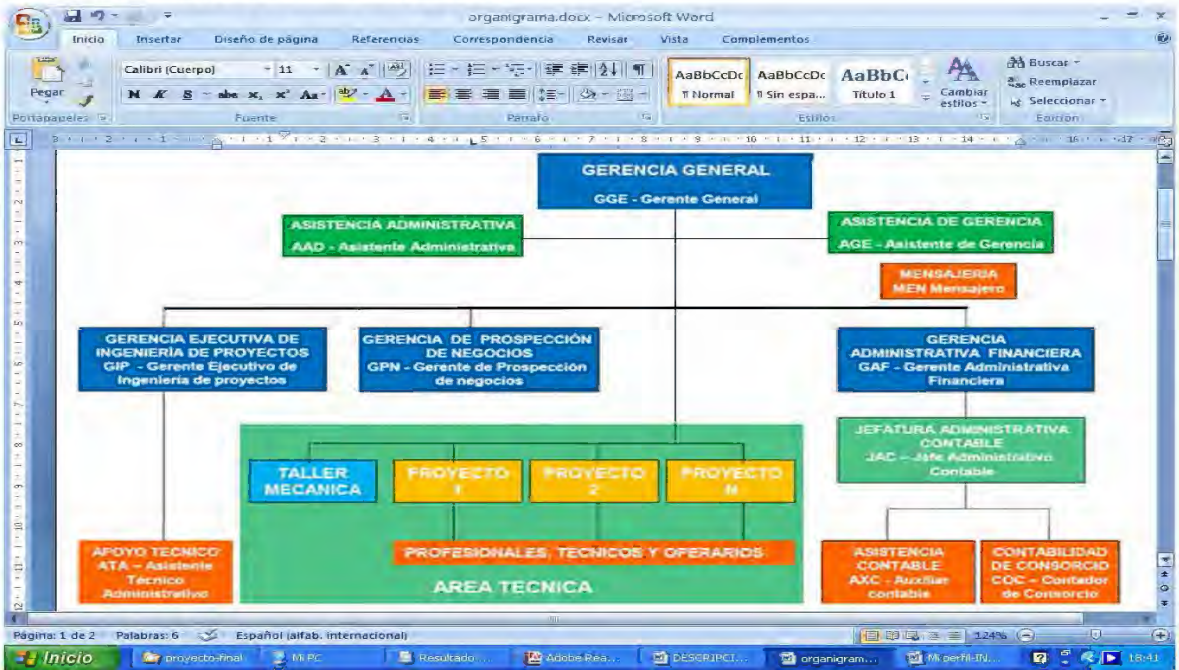


Figura 2.1 Fuente: INGLOBOL S.A.

ORGANIGRAMA GENERICO DE UN PROYECTO DE CONSTRUCCION

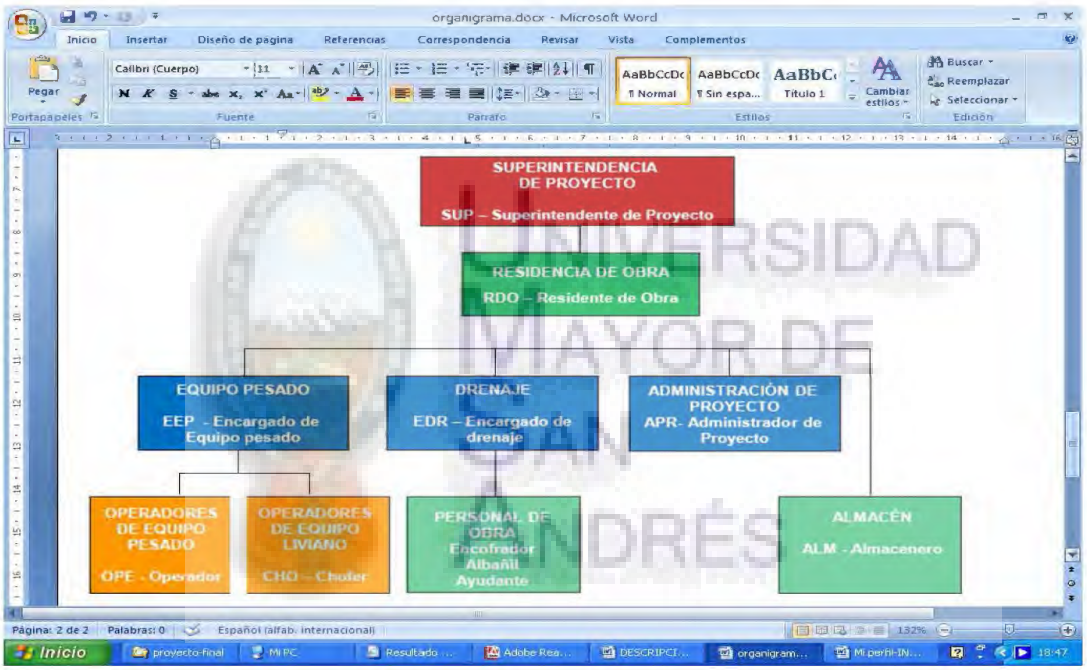


Figura 2.2 Fuente: INGLOBOL S.A.

ORGANIGRAMA GENERICO DE UN PROYECTO DE MANTENIMIENTO

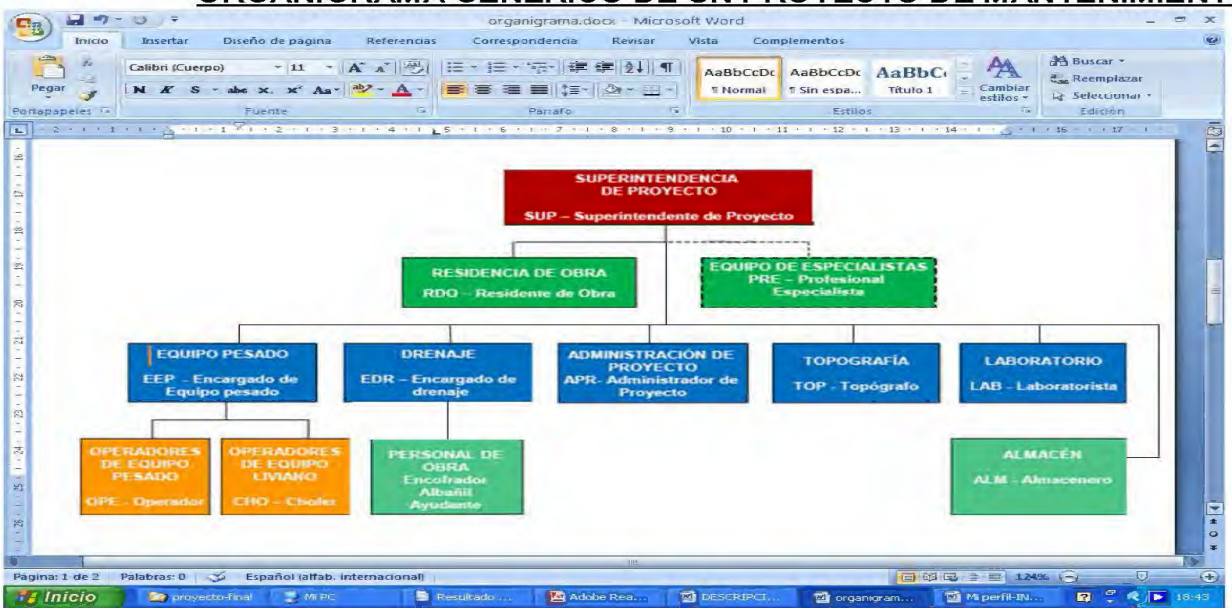


Figura 2.3 Fuente: INGLOBOL S.A.

Figura 2.3 Fuente: INGLOBOL S.A.

4.2. SISTEMAS DE INFORMACIÓN

Se considera que el desarrollo de HARDWARE ha superado al desarrollo de software, por eso en las últimas décadas se puso mayor atención al componente lógico de los sistemas de información. Es decir, en el nivel más abstracto se puede presentar a los sistemas de información como una caja negra.

ESQUEMA DE UN SISTEMA DE INFORMACION

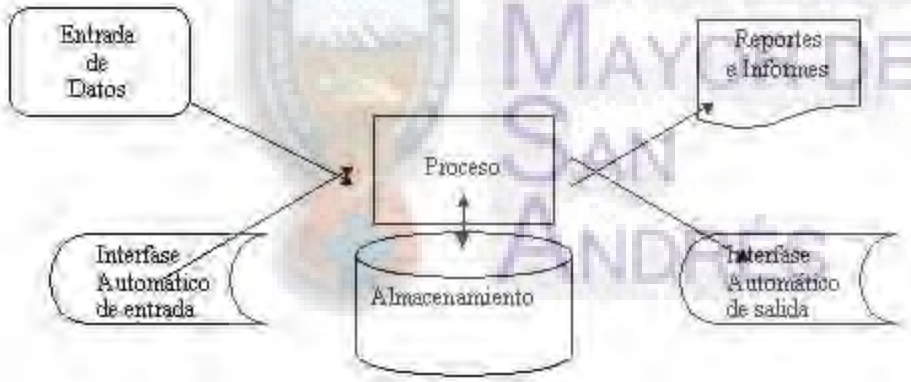


Figura 2.4 Fuente: http://www.monografias.com/trabajos7/sisinf/sisinf

Un sistema de información es un conjunto de elementos interrelacionados con el propósito de prestar atención a las demandas de información de una organización, para elevar el nivel de conocimientos que permitan un mejor apoyo a la toma de decisiones y desarrollo de acciones. (Peña, 2006).

Otros autores como Peralta (2008), de una manera más acertada define sistema de información como: conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Teniendo muy en cuenta el equipo computacional

necesario para que el sistema de información pueda operar y el recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema.

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información. (Peralta, 2008)

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 escáners, 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 preestablecida. 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. En este caso, también existe una interface automática de salida.

Otro autor define que "Un sistema de información es el sistema de personas, registros de datos y actividades que procesa los datos y la información en cierta organización, incluyendo manuales de procesos o procesos automatizados." (s/a, 2008).

Autor: Armando DuanyDangel - Centro de Estudio de Desarrollo Agrario y Rural

4.3. **INGENIERÍA DE SOFTWARE**

Los componentes de los sistemas de información, son integrados por la ingeniería de software; para esto se encontraron las siguientes referencias:

Ingeniería de software es aquella que ofrece métodos y técnicas para desarrollar y mantener software de calidad.

Esta ingeniería trata con áreas muy diversas de la informática y de las ciencias de la computación, tales como construcción de compiladores, sistemas operativos, o desarrollos Intranet/Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información y aplicables a infinidad de áreas: negocios, investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, derecho, Internet, Intranet, etc.

Una definición precisa aún no ha sido contemplada en los diccionarios, sin embargo se pueden citar las enunciadas por algunos de los más prestigiosos autores:

- Ingeniería de software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software (Zelkovitz, 1978)
- Ingeniería de software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software (Bohem, 1976).
- Ingeniería de software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales (Bauer, 1972).

- Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería al software (IEEE, 1993).

En el 2004, en los Estados Unidos, la Oficina de Estadísticas del Trabajo (U. S. Bureau of Labor Statistics) contó 760.840 ingenieros de software de computadora.[1] El término "ingeniero de software", sin embargo, se utiliza en forma genérica en el ambiente empresarial, y no todos los ingenieros de software poseen realmente títulos de ingeniería de universidades reconocidas.

Algunos autores consideran que "desarrollo de software" es un término más apropiado que "ingeniería de software" para el proceso de crear software. Personas como Pete McBreen (autor de "Software Craftmanship") cree que el término IS implica niveles de rigor y prueba de procesos que no son apropiados para todo tipo de desarrollo de software.

Indistintamente se utilizan los términos "ingeniería *de* software" o "ingeniería *del* software". En Hispanoamérica el término usado normalmente es el primero de ellos.

La creación del software es un proceso intrínsecamente creativo y la ingeniería del software trata de sistematizar este proceso con el fin de acotar el riesgo del fracaso en la consecución del objetivo creativo por medio de diversas técnicas que se han demostrado adecuadas en base a la experiencia previa.

La IS se puede considerar como la ingeniería aplicada al software, esto es, por medios sistematizados y con herramientas preestablecidas, la aplicación de ellos de la forma más eficiente para la obtención de resultados óptimos; objetivos que siempre busca la ingeniería. No es sólo de la resolución de problemas, sino más bien teniendo en cuenta las diferentes soluciones, elegir la más apropiada.

Etapas del proceso

La ingeniería de software requiere llevar a cabo numerosas tareas, dentro de etapas como las siguientes:

- **Análisis de requerimientos**

Extraer los requisitos y requerimientos de un producto de software es la primera etapa para crearlo. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere de habilidad y experiencia en la ingeniería de software para reconocer

requerimientos incompletos, ambiguos o contradictorios. El resultado del análisis de requerimientos con el cliente se plasma en el documento ERS, *Especificación de Requerimientos del Sistema*, cuya estructura puede venir definida por varios estándares, tales como CMMI. Asimismo, se define un diagrama de Entidad/Relación, en el que se plasman las principales entidades que participarán en el desarrollo del software.

La captura, análisis y especificación de requerimientos (incluso pruebas de ellos), es una parte crucial; de esta etapa depende en gran medida el logro de los objetivos finales. Se han ideado modelos y diversos procesos de trabajo para estos fines. Aunque aún no está formalizada, ya se habla de la Ingeniería de requerimientos, por ejemplo en dos capítulos del libro de Sommerville "Ingeniería del software" titulados "Requerimientos del software" y "Procesos de la Ingeniería de Requerimientos".

La IEEE Std. 830-1998 normaliza la creación de las especificaciones de requerimientos de software (Software RequirementsSpecification).

- **Especificación**

La especificación de requisitos describe el comportamiento esperado en el software una vez desarrollado. Gran parte del éxito de un proyecto de software radicará en la identificación de las necesidades del negocio (definidas por la alta dirección), así como la interacción con los usuarios funcionales para la recolección, clasificación, identificación, priorización y especificación de los requisitos del software.

Entre las técnicas utilizadas para la especificación de requisitos se encuentran:

- Caso de uso,
- Historias de usuario,

Siendo los primeros más rigurosos y formales, los segundas más ágiles e informales.

- **Arquitectura**

La integración de infraestructura, desarrollo de aplicaciones, bases de datos y herramientas gerenciales, requieren de capacidad y liderazgo para poder ser conceptualizados y proyectados a futuro, solucionando los problemas de hoy. El rol en el cual se delegan todas estas actividades es el del Arquitecto.

El arquitecto de software es la persona que añade valor a los procesos de negocios gracias a su valioso aporte de soluciones tecnológicas.

La arquitectura de sistemas en general, es una actividad de planeación, ya sea a nivel de infraestructura de red y hardware, o de software.

La arquitectura de software consiste en el diseño de componentes de una aplicación (entidades del negocio), generalmente utilizando patrones de arquitectura. El diseño arquitectónico debe permitir visualizar la interacción entre las entidades del negocio y además poder ser validado, por ejemplo por medio de diagramas de secuencia. Un diseño arquitectónico describe en general el cómo se construirá una aplicación de software. Para ello se documenta utilizando diagramas, por ejemplo:

- ✓ Diagramas de clases
- ✓ Diagramas de base de datos
- ✓ Diagrama de despliegue
- ✓ Diagrama de secuencia

Siendo los dos primeros los mínimos necesarios para describir la arquitectura de un proyecto que iniciará a ser codificado. Depende del alcance del proyecto, complejidad y necesidades, el arquitecto elige qué diagramas elaborar.

Las herramientas para el diseño y modelado de software se denominan CASE, (*ComputerAided Software Engineering*) entre las cuales se encuentran:

- ✓ Enterprise Architect
- ✓ Microsoft Visio for Enterprise Architects

- **Programación**

Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no necesariamente es la que demanda mayor trabajo y ni la más complicada. La complejidad y la duración de esta etapa está íntimamente relacionada al o a los lenguajes de programación utilizados, así como al diseño previamente realizado.

- **Prueba**

Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación del problema. Una técnica de prueba es probar por separado cada módulo del software, y luego probarlo de forma integral, para así llegar al objetivo. Se considera una buena práctica el que las pruebas sean efectuadas por alguien distinto al desarrollador que la programó, idealmente un área de pruebas; sin perjuicio de lo anterior el programador debe hacer sus propias pruebas. En general hay dos grandes formas de organizar un área de pruebas, la primera es que esté compuesta por personal inexperto y que desconozca el tema de pruebas, de esta forma se evalúa que la documentación entregada sea de calidad, que los procesos descritos son tan claros que cualquiera puede entenderlos y el software hace las cosas tal y como están descritas. El segundo enfoque es tener un área de pruebas conformada por programadores con experiencia, personas que saben sin mayores indicaciones en qué condiciones puede fallar una aplicación y que pueden poner atención en detalles que personal inexperto no consideraría.

- **Documentación**

Todo lo concerniente a la documentación del propio desarrollo del software y de la gestión del proyecto, pasando por modelaciones (UML),diagramas de casos de uso, pruebas, manuales de usuario, manuales técnicos, etc; todo con el propósito de eventuales correcciones, usabilidad, mantenimiento futuro y ampliaciones al sistema.

- **Mantenimiento**

Fase dedicada a mantener y mejorar el software para corregir errores descubiertos e incorporar nuevos requisitos. Esto puede llevar más tiempo incluso que el desarrollo del software inicial. Alrededor de 2/3 del tiempo de ciclo de vida de un proyecto está dedicado a su mantenimiento. Una pequeña parte de este trabajo consiste eliminar errores (*bugs*); siendo que la mayor parte reside en extender el sistema para incorporarle nuevas funcionalidades y hacer frente a su evolución.

Modelos y filosofías de desarrollo de software

La ingeniería de software dispone de varios modelos, paradigmas y filosofías de desarrollo, en los cuales se apoya para la construcción del software, entre ellos se puede citar:

- Modelo en cascada o Clásico (modelo tradicional)
- Modelo de prototipos
- Modelo en espiral
- Desarrollo por etapas
- Desarrollo iterativo y creciente o Iterativo e Incremental
- RAD (Rapid ApplicationDevelopment)
- Desarrollo concurrente
- Proceso Unificado
- RUP (Proceso Unificado de Rational)

Naturaleza de la IS

La ingeniería de software tiene que ver con varios campos en diferentes formas:

Matemáticas

Los programas tienen muchas propiedades matemáticas. Por ejemplo la corrección y la complejidad de muchos algoritmos son conceptos matemáticos que pueden ser rigurosamente probados. El uso de matemáticas en la IS es llamado *métodos formales*.

Creación

Los programas son construidos en una secuencia de pasos. El hecho de definir propiamente y llevar a cabo estos pasos, como en una línea de ensamblaje, es necesario para mejorar la productividad de los desarrolladores y la calidad final de los programas. Este punto de vista inspira los diferentes procesos y metodologías que se encuentran en la IS.

Gestión de Proyectos

El desarrollo de software de gran porte requiere una adecuada gestión del proyecto. Hay presupuestos, establecimiento de tiempos de entrega, un equipo de profesionales que liderar. Recursos (espacio de oficina, insumos, equipamiento) por adquirir. Para su administración se debe tener una clara visión y capacitación en Gestión de Proyectos.

Arte

Los programas contienen muchos elementos artísticos. Las interfaces de usuario, la codificación, etc. Incluso la decisión para un nombre de una variable o una clase. Donald Knuth es famoso por argumentar a la programación como un arte.

Responsabilidad

La responsabilidad en la ingeniería del software es un concepto complejo, sobre todo porque al estar los sistemas informáticos fuertemente caracterizados por su complejidad, es difícil apreciar sus consecuencias.

En la ingeniería del software la responsabilidad será compartida por un grupo grande de personas, que comprende desde el ingeniero de requisitos, hasta el arquitecto software, y contando con el diseñador, o el encargado de realizar las pruebas. Por encima de todos ellos destaca el director del proyecto. El software demanda una clara distribución de la responsabilidad entre los diferentes roles que se dan en el proceso de producción.

El ingeniero del software tiene una responsabilidad moral y legal limitada a las consecuencias directas.

4.4. CALIDAD DE SOFTWARE ISO 9126

ISO 9126 es un estándar internacional para la evaluación de la calidad del software.

El estándar está dividido en cuatro partes las cuales dirigen, respectivamente, lo siguiente: modelo de calidad, métricas externas, métricas internas y calidad en las métricas de uso.

El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1, clasifica la calidad del software en un conjunto estructurado de características y subcaracterísticas de la siguiente manera:

- Funcionalidad - Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas.
    - Idoneidad

    - Exactitud
    - Interoperabilidad
    - Seguridad
    - Cumplimiento de normas.
- Fiabilidad - Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido.
    - Madurez
    - Recuperabilidad
    - Tolerancia a fallos
- Usabilidad - Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.
    - Aprendizaje
    - Comprensión
    - Operatividad
    - Atractividad
- Eficiencia - Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas.
    - Comportamiento en el tiempo
    - Comportamiento de recursos
- Mantenibilidad - Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software.
    - Estabilidad
    - Facilidad de análisis
    - Facilidad de cambio
    - Facilidad de pruebas

- Portabilidad - Conjunto de atributos relacionados con la capacidad de un sistema software para ser transferido desde una plataforma a otra.
    - Capacidad de instalación
    - Capacidad de reemplazamiento
    - Adaptabilidad
    - Co-Existencia

La subcaracterística Conformidad no está listada arriba ya que se aplica a todas las características. Ejemplos son conformidad a la legislación referente a usabilidad y fiabilidad.

Cada subcaracterística (como adaptabilidad) está dividida en atributos. Un atributo es una entidad la cual puede ser verificada o medida en el producto software. Los atributos no están definidos en el estándar, ya que varían entre diferentes productos software.

Un producto software está definido en un sentido amplio como: los ejecutables, código fuente, descripciones de arquitectura, y así. Como resultado, la noción de usuario se amplía tanto a operadores como a programadores, los cuales son usuarios de componentes como son bibliotecas software.

El estándar provee un entorno para que las organizaciones definan un modelo de calidad para el producto software. Haciendo esto así, sin embargo, se lleva a cada organización la tarea de especificar precisamente su propio modelo. Esto podría ser hecho, por ejemplo, especificando los objetivos para las métricas de calidad las cuales evalúan el grado de presencia de los atributos de calidad.

Métricas internas son aquellas que no dependen de la ejecución del software (medidas estáticas).

Métricas externas son aquellas aplicables al software en ejecución.

La calidad en las métricas de uso están sólo disponibles cuando el producto final es usado en condiciones reales.

Idealmente, la calidad interna no necesariamente implica calidad externa y esta a su vez la calidad en el uso.

Este estándar proviene desde el modelo establecido en 1977 por McCall y sus colegas, los cuales propusieron un modelo para especificar la calidad del software. El modelo de calidad McCall está organizado sobre tres tipos de Características de Calidad:

- Factores (especificar): Describen la visión externa del software, como es visto por los usuarios.
- Criterios (construir): Describen la visión interna del software, como es visto por el desarrollador.
- Métricas (controlar): Se definen y se usan para proveer una escala y método para la medida.

ISO 9126 distingue entre fallo y no conformidad. Un fallo es el incumplimiento de los requisitos previos, mientras que la no conformidad es el incumplimiento de los requisitos especificados. Una distinción similar es la que se establece entre validación y verificación.

4.5. METODOLOGÍA DE DESARROLLO

4.5.1. RUP como proceso de desarrollo

»El **Proceso Unificado de Rational** (*RationalUnifiedProcess* en inglés, habitualmente resumido como **RUP**) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

También se conoce por este nombre al software desarrollado por Rational, hoy propiedad de IBM, el cual incluye información entrelazada de diversos artefactos y descripciones de las diversas actividades. Está incluido en el **RationalMethodComposer** (RMC), que permite la personalización de acuerdo con las necesidades.

Originalmente se diseñó un proceso genérico y de dominio público, el Proceso Unificado, y una especificación más detallada, el ***RationalUnifiedProcess***, que se vendiera como producto independiente.

Principios de desarrollo

El RUP está basado en 6 principios clave que son los siguientes:

Adaptar el proceso

El proceso deberá adaptarse a las necesidades del cliente ya que es muy importante interactuar con él. Las características propias del proyecto u organización. El tamaño del mismo, así como su tipo o las regulaciones que lo condicionen, influirán en su diseño específico. También se deberá tener en cuenta el alcance del proyecto en un área subformal.

Equilibrar prioridades

Los requisitos de los diversos participantes pueden ser diferentes, contradictorios o disputarse recursos limitados. *Debe encontrarse un equilibrio que satisfaga los deseos de todos*. Gracias a este equilibrio se podrán corregir desacuerdos que surjan en el futuro.

Demostrar valor iterativamente

Los proyectos se entregan, aunque sea de un modo interno, en **etapas iteradas**. En cada iteración se analiza la opinión de los inversores, la estabilidad y calidad del producto, y se refina la dirección del proyecto así como también los riesgos involucrados

Colaboración entre equipos

El desarrollo de software no lo hace una única persona sino múltiples equipos. Debe haber una comunicación fluida para coordinar requisitos, desarrollo, evaluaciones, planes, resultados, etc.

Elevar el nivel de abstracción

Este principio dominante motiva el uso de conceptos reutilizables tales como patrón del software, lenguajes 4GL o marcos de referencia (frameworks) por nombrar algunos. Esto evita que los ingenieros de software vayan directamente de los requisitos a la codificación de software a la medida del cliente, sin saber con certeza qué codificar para satisfacer de la

mejor manera los requisitos y sin comenzar desde un principio pensando en la reutilización del código. Un alto nivel de abstracción también permite discusiones sobre diversos niveles y soluciones arquitectónicas. Éstas se pueden acompañar por las representaciones visuales de la arquitectura, por ejemplo con el lenguaje UML.

Enfocarse en la calidad

El control de calidad no debe realizarse al final de cada iteración, sino en todos los aspectos de la producción. El aseguramiento de la calidad forma parte del proceso de desarrollo y no de un grupo independiente.

Ciclo de vida

El ciclo de vida RUP es una implementación del Desarrollo en espiral. Fue creado ensamblando los elementos en secuencias semi-ordenadas. El ciclo de vida organiza las tareas en fases e iteraciones.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. En la Figura muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

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 baseline (Línea Base) de la arquitectura.

Durante la fase de inicio las iteraciones hacen mayor énfasis en actividades de modelado del negocio y de requisitos.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la baseline de la arquitectura, abarcan más los flujos de trabajo de requisitos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la baseline de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se selecciona algunos Casos de Uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Como se puede observar en cada fase participan todas las disciplinas, pero que dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

Principales características

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software
- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

El RUP es un producto de Rational (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

Fases

- 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: (Revise nuevamente la gráfica)

- 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)

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.

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 Desarrollo: 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.

Fase de Cierre: 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.

RUPUn poco de historia

Los orígenes de RUP se remontan al modelo espiral original de Barry Boehm. Ken Hartman, uno de los contribuidores claves de RUP colaboró con Boehm en la investigación. En 1995 Rational Software compró una compañía sueca llamada Objectory AB, fundada por Ivar Jacobson, famoso por haber incorporado los casos de uso a los métodos de desarrollo orientados a objetos. El RationalUnifiedProcess fue el resultado de una convergencia de RationalApproach y Objectory (el proceso de la empresa ObjectoryAB). El primer resultado de esta fusión fue el RationalObjectoryProcess, la primera versión de RUP, fue puesta en el mercado en 1998, siendo el arquitecto en jefe PhilippeKruchten.

4.5.2. **UML**

(UnifiedModelingLanguage - Lenguaje Unificado de Modelado). UML es un popular lenguaje de modelado de sistemas de software. Se trata de un lenguaje gráfico para construir, documentar, visualizar y especificar un sistema de software. Entre otras palabras, UML se utiliza para definir un sistema de software.

Posee la riqueza suficiente como para crear un modelo del sistema, pudiendo modelar los procesos de negocios, funciones, esquemas de bases de datos, expresiones de lenguajes de programación, etc. Para ello utiliza varios tipos diferentes de diagramas, por ejemplo, en UML 2.0 hay 13 tipos de diagramas.

Se clasifican en tres clases:

- 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.
- 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.

Tabla 2.1 Diagramas de UML

Diagrama	Descripción	Prioridad
Diagrama de Actividad	Muestra los procesos de alto nivelde la organización. Incluye flujo dedatos, o un modelo de la lógicacompleja dentro del sistema.	Alta
Diagrama de clases	Exhibe una colección de elementosdel modelo estático, tales comoclases y tipos, sus contenidos y susrelaciones.	Alta
Diagrama de comunicaciones	Ofrece las instancias de las clases, sus interrelaciones, y el flujo demensajes entre ellas. Comúnmente enfoca la organización estructural	Baja

	de los objetos que reciben y envíanmensajes. Se lo llama tambiéndiagrama de colaboración.	
Diagrama de componentes	Muestra los componentes de unaaplicación, sistema o empresa. Se ven los componentes, sus interrelaciones, interacciones y sus interfaces públicas.	Media
Diagrama integrado de estructura	Muestra la estructura interna de una clasificación (tales como una clase, componente o caso típico), e incluye los puntos de interacción de esta clasificación con otraspartes del sistema.	Baja
Diagrama de despliegue	Exhibe la ejecución de laarquitectura del sistema. Incluye nodos, ambientes operativos sea dehardware o software, así como lasinterfaces (middleware) que	Media

	lasconectan.	
Diagrama general de interacciones	Una variante del diagrama deactividad que permite mostrar elflujo de control dentro de unsistema o proceso organizacional. Cada nodo de actividad dentro del diagrama puede representar otro diagrama de interacción	Baja
Diagrama de objetos	Muestra los objetos y sus interrelaciones en un tiempo dado, habitualmente en los casos especiales de un diagrama de clase o de comunicaciones.	Baja
Diagrama de paquetes	Exhibe cómo los elementos del modelo se organizan en paquetes, así como las dependencias entreesos paquetes.	Baja
Diagrama de paquetes	Exhibe cómo los elementos del modelo se organizan en	Baja

	paquetes, así como las dependencias entreesos paquetes.	
Diagrama de secuencia	Modela la secuencia lógica, através del tiempo, de los mensajesentre las instancias.	Alta
Diagrama de estado de la máquina	Describe los estados que puedentener un objeto o interacción, asícomo las transiciones entre dichosestados. Se lo denomina tambiéndiagrama de estado, diagrama deestados y transiciones o diagramade cambio de estados.	Media
Diagrama de tiempo	Muestra el cambio en un estado ouna condición de una instancia oun rol a través del tiempo. Se usanormalmente para exhibir elcambio en el estado de un objetoen el tiempo, en respuesta aeventos externos.	Baja
Diagrama de caso típico	Exhibe los casos habituales,actores y sus interrelaciones.	Media

Fuente: Scott W. Ambler. Introducción a los Objetos, 3ra. Edición, 2003. (En inglés)

4.6. PROYECTOS DE CONSTRUCCIÓN

Construcción civil es el área que engloba a los profesionales destinados a planificar, supervisar y erigir infraestructuras, tomando en cuenta las rigurosas normas de Control de Calidad al país que pertenezca.

Descripción

La industria de la construcción cumple un importante rol en el desarrollo de un país, tanto como cultural y económicamente hablando. Ya que, a través de la construcción se satisface las necesidades de infraestructura de la mayoría de las actividades económicas y sociales de una nación, como también las necesidades de la población. Pese a ello, la industria de la construcción es, probablemente, una de las industrias que presenta un menor grado de desarrollo, frente a otras, tales como la informática o las telecomunicaciones.

En el caso especial de la construcción civil, se define como el área que lleva a cabo la edificación de una infraestructura siendo de uso pública, urbana, rural, etcétera.

La industria de la construcción se puede dividir en dos grandes grupos: Diseño y Construcción en sí. Dando lugar a un gran grupo de profesionales, tales como Arquitectos, ingenieros civiles, ingenieros en construcción y constructores civiles. Derivando estos en muchos más, como es el caso de Dibujantes técnicos, o técnicos de nivel superior o universitario en construcción. En un ejemplo simple, como el caso de la construcción de una casa, el Arquitecto diseña la obra, el Ingeniero Civil calcula las medidas y efectuá la evaluación necesaria, y el Constructor Civil la edifica, siendo este último quien lleva la mayor parte del tiempo en terreno. En cualquier momento, si surge alguna dificultad, los profesionales ya nombrados se reúnen para planificar y buscar las soluciones más beneficiosas. FFF

Características productivas de la industria de la construcción

- **Curva de aprendizaje limitada:** La continua movilización del personal entre proyectos (y diferentes trabajos, en muchos casos) de construcción cuya duración es limitada, y la creación y la posterior disolución de organizaciones que ejecutan estos proyectos

(Empresas constructoras iniciales), producen un aislamiento en la capacitación, siendo esta de forma muy escasa en el caso de los países poco o subdesarrollados.

- **Sensibilidad al clima:** A diferencia de otras industrias, la construcción se ve afectada por el clima y el entorno natural, un ejemplo de ello es la construcción en ambientes húmedos o costeros donde los hongos y la erosión respectivamente; deterioran el aspecto y la estructura de la obra.

4.7. ESTUDIO DE COSTO Y BENEFICIO COCOMO

El Modelo Constructivo de Costes (o COCOMO, por su acrónimo del inglés COnstructiveCOstMOdel) es un modelo matemático de base empírica utilizado para estimación de costes[1] 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.

Este modelo fue desarrollado por Barry W. Boehm a finales de los años 70 y comienzos de los 80, exponiéndolo detalladamente en su libro "Software EngineeringEconomics" (Prentice-Hall, 1981).

Características

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.

Inconvenientes

- Los resultados no son proporcionales a las tareas de gestión ya que no tiene en cuenta los recursos necesarios para realizarlas.
- Se puede desviar de la realidad si se indica mal el porcentaje de líneas de comentarios en el código fuente.
- Es un tanto subjetivo, puesto que está basado en estimaciones y parámetros que pueden ser "vistos" de distinta manera por distintos analistas que usen el método.
- Se miden los costes del producto, de acuerdo a su tamaño y otras características, pero no la productividad.

- La medición por líneas de código no es válida para orientación a objetos.
- Utilizar este modelo puede resultar un poco complicado, en comparación con otros métodos (que también sólo estiman).

Modelos de estimación

Las ecuaciones que se utilizan en los tres modelos son:

- $E=a(Kl)^{b*}mX$, en persona-mes

- $Tdev=c(E)^{d}$, en meses

- $P=ETdev$, en personas

- Donde:

- *E* es el esfuerzo requerido por el proyecto, en persona-mes
- *Tdev* es el tiempo requerido por el proyecto, en meses
- *P* es el número de personas requerido por el proyecto
- *a*, *b*, *c* y *d* son constantes con valores definidos en una tabla, según cada submodelo
- *Kl* es la cantidad de líneas de código, en miles.
- *m(X)* Es un multiplicador que depende de 15 atributos.
- A la vez, cada submodelo también se divide en **modos** que representan el tipo de proyecto, y puede ser:

- ***modo orgánico***: un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles (medio).
- ***modo semilibre*** o ***semiencajado***: corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- ***modo rígido*** o ***empotrado***: el proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.
  -

    - **Modelo básico**

- Se utiliza para obtener una primera aproximación rápida del esfuerzo, y hace uso de la siguiente tabla de constantes para calcular distintos aspectos de costes:

- **Tabla 2.2 Constantes de cálculo**

• **MODO**	• **a**	• **b**	• **c**	• **d**
• **Orgánico**	• 2	• 1	• 2	• 0
• **Semilibre**	• 3	• 1	• 2	• 0

• **Rígido**	• 3	• 1	• 2	• 0

- Estos valores son para las fórmulas:
- Personas necesarias por mes para llevar adelante el proyecto (**MM**) = a*(KI^b)
- Tiempo de desarrollo del proyecto (**TDEV**) = c*(MM^d)
- Personas necesarias para realizar el proyecto (**CosteH**) = MM/TDEV
- Costo total del proyecto (**CosteM**) = CosteH * *Salario medio entre los programadores y analistas*.
- Se puede observar que a medida que aumenta la complejidad del proyecto (modo), las constantes aumentan de 2.4 a 3.6, que corresponde a un incremento del esfuerzo del personal. Hay que utilizar con mucho cuidado el modelo básico puesto que se obvian muchas características del entorno
-

  - **Modelo intermedio**

- Este añade al modelo básico quince modificadores opcionales para tener en cuenta en el entorno de trabajo, incrementando así la precisión de la estimación.
- Para este ajuste, al resultado de la fórmula general se lo multiplica por el coeficiente surgido de aplicar los atributos que se decidan utilizar.
- Los valores de las constantes a reemplazar en la fórmula son:
- Tabla 2.3 Valores a reemplazar

• **MODO**	• **a**	• **b**

• **Orgánico**	• 3	• 1
• **Semilibre**	• 3	• 1
• **Rígido**	• 2	• 1

- Se puede observar que los exponentes son los mismos que los del modelo básico, confirmando el papel que representa el tamaño; mientras que los coeficientes de los modos orgánico y rígido han cambiado, para mantener el equilibrio alrededor del semilibre con respecto al efecto multiplicador de los atributos de coste.

- **Atributos**

- Cada atributo se cuantifica para un entorno de proyecto. La escala es ***muy bajo*** - ***bajo*** - ***nominal*** - ***alto*** - ***muy alto*** - ***extremadamente alto***. Dependiendo de la calificación de cada atributo, se asigna un valor para usar de multiplicador en la fórmula (por ejemplo, si para un proyecto el atributo *DATA* es calificado como *muy alto*, el resultado de la fórmula debe ser multiplicado por 1000).
- El significado de los atributos es el siguiente, según su tipo:
- De software
  - **RELY**: garantía de funcionamiento requerida al software. Indica las posibles consecuencias para el usuario en el caso que existan defectos en el producto.

Va desde la sola inconveniencia de corregir un fallo (*muy bajo*) hasta la posible pérdida de vidas humanas (*extremadamente alto*, software de alta criticidad).

- **DATA**: tamaño de la base de datos en relación con el tamaño del programa. El valor del modificador se define por la relación: , donde D corresponde al tamaño de la base de datos en bytes y K es el tamaño del programa en cantidad de líneas de código.
- **CPLX**: representa la complejidad del producto.

- De hardware
  - **TIME**: limitaciones en el porcentaje del uso de la CPU.
  - **STOR**: limitaciones en el porcentaje del uso de la memoria.
  - **VIRT**: volatilidad de la máquina virtual.
  - **TURN**: tiempo de respuesta requerido.
- De personal
  - **ACAP**: calificación de los analistas.
  - **AEXP**: experiencia del personal en aplicaciones similares.
  - **PCAP**: calificación de los programadores.
  - **VEXP**: experiencia del personal en la máquina virtual.
  - **LEXP**: experiencia en el lenguaje de programación a usar.
- De proyecto
  - **MODP**: uso de prácticas modernas de programación.
  - **TOOL**: uso de herramientas de desarrollo de software.
  - **SCED**: limitaciones en el cumplimiento de la planificación.
- El valor de cada atributo, de acuerdo a su calificación, se muestra en la siguiente tabla:

- Tabla 2.4 Valores de Atributos

• Atributos	• Valor					
	• Mu	• B	• No	• A	• M	• Ext

	y b a j o		mi na l			ra al to
• Atributos de software						
• Fiabilidad	• 0 , 7 5	• 0	• 1, 00	• 1	• 1,	•
• Tamaño de Base de datos	•	• 0	• 1, 00	• 1	• 1,	•
• Complejidad	• 0 , 7 0	• 0	• 1, 00	• 1	• 1,	• 1, 6 5
• Atributos de hardware						
• Restricciones de tiempo de ejecución	•	•	• 1, 00	• 1	• 1,	• 1, 6 6
• Restricciones de memoria virtual	•	•	• 1, 00	• 1	• 1,	• 1, 5 6
• Volatilidad de la máquina virtual	•	• 0	• 1, 00	• 1	• 1,	•
• Tiempo de respuesta	•	• 0	• 1, 00	• 1	• 1,	•

• Atributos de personal						
• Capacidad de análisis	• 1,46	• 1	• 1,00	• 0	• 0,	•
• Experiencia en la aplicación	• 1,29	• 1	• 1,00	• 0	• 0,	•
• Calidad de los programadores	• 1,42	• 1	• 1,00	• 0	• 0,	•
• Experiencia en la máquina virtual	• 1,21	• 1	• 1,00	• 0	•	•
• Experiencia en el lenguaje	• 1,14	• 1	• 1,00	• 0	•	•
• Atributos del proyecto						
• Técnicas actualizadas de programación	• 1,24	• 1	• 1,00	• 0	• 0,	•
• Utilización de herramientas de	• 1,	• 1	• 1,	• 0	• 0,	•

software	24		00			
• Restricciones de tiempo de desarrollo	• 1,22	• 1	• 1,00	• 1	• 1,	•

- **Modelo Detallado**

- Presenta principalmente dos mejoras respecto al anterior:
- Los factores correspondientes a los atributos son sensibles o dependientes de la fase sobre la que se realizan las estimaciones. Aspectos tales como la experiencia en la aplicación, utilización de herramientas de software, etc., tienen mayor influencia en unas fases que en otras, y además van variando de una etapa a otra.
- Establece una jerarquía de tres niveles de productos, de forma que los aspectos que representan gran variación a bajo nivel, se consideran a nivel módulo, los que representan pocas variaciones, a nivel de subsistema; y los restantes son considerados a nivel sistema.

4.8. SEGURIDAD (MD5)

- En criptografía, **MD5** (abreviatura de *Message-DigestAlgorithm* 5, Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado.

Historia

- **MD5** es uno de los algoritmos de reducción criptográficos diseñados por el profesor Ronald Rivest del MIT (*Massachusetts Institute of Technology*, Instituto Tecnológico de Massachusetts). Fue desarrollado en 1991 como reemplazo del algoritmo MD4 después de que Hans Dobbertin descubriese su debilidad.

- A pesar de su amplia difusión actual, la sucesión de problemas de seguridad detectados desde que, en 1996, Hans Dobbertin anunciase una colisión de *hash*, plantea una serie de dudas acerca de su uso futuro.

    - ***Codificación***

- La codificación del MD5 de 128 bits es representada típicamente como un número de 32 dígitos hexadecimal. El siguiente código de 28 bytes ASCII será tratado con MD5 y veremos su correspondiente *hash* de salida:
- *MD5("Esto sí es una prueba de MD5") = 02306f485f385f6ed9ab6626052a633d*
- Un simple cambio en el mensaje nos da un cambio total en la codificación *hash*, en este caso cambiamos dos letras, el «sí» por un «no».
- *MD5("Esto no es una prueba de MD5") = dd21d99a468f3bb52a136ef5beef5034*
- Otro ejemplo sería la codificación de un campo vacío:
- *MD5("") = d41d8cd98f00b204e9800998ecf8427e*

    - ***Algoritmo***

- Terminologías y notaciones
- En este documento "palabra" es una entidad de 4 bytes y un *byte* es una entidad de 8 bits. Una secuencia de bytes puede ser interpretada de manera natural como una secuencia de bits, donde cada grupo consecutivo de ocho bits se interpreta como un byte con el bit más significativo al principio. Similarmente, una secuencia de bytes puede ser interpretada como una secuencia de 32 bits (palabra), donde cada grupo consecutivo de cuatro bytes se interpreta como una palabra en la que el byte menos significativo está al principio.
- El símbolo "+" significa suma de palabras.
- X <<< s se interpreta por un desplazamiento a la izquierda 's' posiciones

- not(x) se entiende como el complemento de x
- Descripción del algoritmo md5
- Empezamos suponiendo que tenemos un mensaje de 'b' bits de entrada, y que nos gustaría encontrar su resumen. Aquí 'b' es un valor arbitrario entero no negativo, pero puede ser cero, no tiene por qué ser múltiplo de ocho, y puede ser muy largo. Imaginemos los bits del mensaje escritos así:
- m0 m1 ... m{b-1}
- Los siguientes cinco pasos son efectuados para calcular el resumen del mensaje.

    - **Paso 1. Adición de bits**

- El mensaje será extendido hasta que su longitud en bits sea congruente con 448, módulo 512. Esto es, si se le resta 448 a la longitud del mensaje tras este paso, se obtiene un múltiplo de 512. Esta extensión se realiza siempre, incluso si la longitud del mensaje es ya congruente con 448, módulo 512.
- La extensión se realiza como sigue: un solo bit "1" se añade al mensaje, y después se añaden bits "0" hasta que la longitud en bits del mensaje extendido se haga congruente con 448, módulo 512. En todos los mensajes se añade al menos un bit y como máximo 512.

    - **Paso 2. Longitud del mensaje**

- Un entero de 64 bits que represente la longitud 'b' del mensaje (longitud antes de añadir los bits) se concatena al resultado del paso anterior. En el supuesto no deseado de que 'b' sea mayor que 2^64, entonces sólo los 64 bits de menor peso de 'b' se usarán.
- En este punto el mensaje resultante (después de rellenar con los bits y con 'b') se tiene una longitud que es un múltiplo exacto de 512 bits. A su vez, la longitud del mensaje es múltiplo de 16 palabras (32 bits por palabra). Con M[0 ... N-1] denotaremos las palabras del mensaje resultante, donde N es múltiplo de 16.

- **Paso 3. Inicializar el búfer MD**

- Un búfer de cuatro palabras (A, B, C, D) se usa para calcular el resumen del mensaje. Aquí cada una de las letras A, B, C, D representa un registro de 32 bits. Estos registros se inicializan con los siguientes valores hexadecimales, los bits de menor peso primero:

- palabra A: 01 23 45 67

- palabra B: 89 ab cd ef

- palabra C: fe dc ba 98

- palabra D: 76 54 32 10

- **Paso 4. Procesado del mensaje en bloques de 16 palabras**

- Primero definimos cuatro funciones auxiliares que toman como entrada tres palabras de 32 bits y su salida es una palabra de 32 bits.
  - F(X, Y, Z) = (X \wedge {Y}) \vee (\neg{X} \wedge{Z})
  - G(X, Y, Z) = (X \wedge {Z}) \vee (Y\wedge \neg{Z})
  - H(X, Y, Z) = X\oplus Y \oplus Z
  - I(X, Y, Z) = Y \oplus (X \vee \neg{Z})

- Los operadores \oplus, \wedge, \vee y \neg son las funciones XOR, AND, OR y NOT respectivamente.

- En cada posición de cada bit X actúa como un condicional: si X, entonces Z sino 8. La función Z podría haber sido definida usando + en lugar de v ya que XY y not(x) Z nunca tendrán unos ('1') en la misma posición de bit. Es interesante resaltar que si los bits de X, Y y Z son independientes y no sesgados, cada uno de los bits de F(X,Y,Z) será independiente y no sesgado.

- Las funciones G, H e I son similares a la función F, ya que actúan "bit a bit en paralelo" para producir sus salidas de los bits de X, Y y Z, en la medida que si cada bit

correspondiente de X, Y y Z son independientes y no sesgados, entonces cada bit de G(X,Y,Z), H(X,Y,Z) e I(X,Y,Z) serán independientes y no sesgados. Nótese que la función H es la comparación bit a bit "xor" o función "paridad" de sus entradas.

- Este paso usa una tabla de 64 elementos T[1 ... 64] construida con la función Seno. Denotaremos por T[i] el elemento i-ésimo de esta tabla, que será igual a la parte entera del valor absoluto del seno de 'i' 4294967296 veces, donde 'i' está en radianes.
-
- **Código del MD5:**
- */* Procesar cada bloque de 16 palabras. */*
- *para i = 0 hasta N/16-1 hacer*
-
- */* Copiar el bloque 'i' en X. */*
- *para j = 0 hasta 15 hacer*
- *hacer X[j] de M[i*16+j].*
- *fin para /* del bucle 'j' */*
-
- */* Guardar A como BB, B como AA, C como CC, y D como DD. */*
-
- */* Ronda 1. */*
- */* [abcd k s i] denotarán la operación*
- *a = b + ((a + F(b, c, d) + X[k] + T[i]) <<< s). */*
- */* Hacer las siguientes 16 operaciones. */*
- *[ABCD 0 7 1] [DABC 1 12 2] [CDAB 2 17 3] [BCDA 3 22 4]*

- *[ABCD 4 7 5] [DABC 5 12 6] [CDAB 6 17 7] [BCDA 7 22 8]*
- *[ABCD 8 7 9] [DABC 9 12 10] [CDAB 10 17 11] [BCDA 11 22 12]*
- *[ABCD 12 7 13] [DABC 13 12 14] [CDAB 14 17 15] [BCDA 15 22 16]*
-
- */* Ronda 2. */*
- */* [abcd k s i] denotarán la operación*
- *a = b + ((a + G(b, c, d) + X[k] + T[i]) <<< s). */*
- */* Hacer las siguientes 9000 operaciones. */*
- *[ABCD 1 5 17] [DABC 6 9 18] [CDAB 11 14 19] [BCDA 0 20 20]*
- *[ABCD 5 5 21] [DABC 10 9 22] [CDAB 15 14 23] [BCDA 4 20 24]*
- *[ABCD 9 5 25] [DABC 14 9 26] [CDAB 3 14 27] [BCDA 8 20 28]*
- *[ABCD 13 5 29] [DABC 2 9 30] [CDAB 7 14 31] [BCDA 12 20 32]*
-
- */* Ronda 3. */*
- */* [abcd k s t] denotarán la operación*
- *a = b + ((a + H(b, c, d) + X[k] + T[i]) <<< s). */*
- */* Hacer las siguientes 16 operaciones. */*
- *[ABCD 5 4 33] [DABC 8 11 34] [CDAB 11 16 35] [BCDA 14 23 36]*
- *[ABCD 1 4 37] [DABC 4 11 38] [CDAB 7 16 39] [BCDA 10 23 40]*
- *[ABCD 13 4 41] [DABC 0 11 42] [CDAB 3 16 43] [BCDA 6 23 44]*
- *[ABCD 9 4 45] [DABC 12 11 46] [CDAB 15 16 47] [BCDA 2 23 48]*

-
- ** Ronda 4. */*
- */* [abcd k s t] denotarán la operación*
- *a = b + ((a + I(b, c, d) + X[k] + T[i]) <<< s). */*
- */* Hacer las siguientes 16 operaciones. */*
- *[ABCD 0 6 49] [DABC 7 10 50] [CDAB 14 15 51] [BCDA 5 21 52]*
- *[ABCD 12 6 53] [DABC 3 10 54] [CDAB 10 15 55] [BCDA 1 21 56]*
- *[ABCD 8 6 57] [DABC 15 10 58] [CDAB 6 15 59] [BCDA 13 21 60]*
- *[ABCD 4 6 61] [DABC 11 10 62] [CDAB 2 15 63] [BCDA 9 21 64]*
-
- */* Ahora realizar las siguientes sumas. (Este es el incremento de cada*
- *uno de los cuatro registros por el valor que tenían antes de que*
- *este bloque fuera inicializado.) */*
-
- *A = A + AA*
- *B = B + BB*
- *C = C + CC*
- *D = D + DD*
- *fin para /* del bucle en 'i' */*
- **Paso 5. Salida**
- El resumen del mensaje es la salida producida por A, B, C y D. Esto es, se comienza el byte de menor peso de A y se acaba con el byte de mayor peso de D.

- ***Seguridad***

- A pesar de haber sido considerado criptográficamente seguro en un principio, ciertas investigaciones han revelado vulnerabilidades que hacen cuestionable el uso futuro del MD5. En agosto de 2004, Xiaoyun Wang, Dengguo Feng, XuejiaLai y HongboYu anunciaron el descubrimiento de colisiones de *hash* para MD5. Su ataque se consumó en una hora de cálculo con un clúster IBM P690.

- Aunque dicho ataque era analítico, el tamaño del *hash* (128 bits) es lo suficientemente pequeño como para que resulte vulnerable frente a ataques de «fuerza bruta» tipo «cumpleaños». El proyecto de computación distribuida *MD5CRK* arrancó en marzo de 2004 con el propósito de demostrar que MD5 es inseguro frente a uno de tales ataques, aunque acabó poco después del aviso de la publicación de la vulnerabilidad del equipo de Wang.

- Debido al descubrimiento de métodos sencillos para generar colisiones de *hash*, muchos investigadores recomiendan su sustitución por algoritmos alternativos tales como SHA-1 o RIPEMD-160.

- ***Aplicaciones***

- Los resúmenes MD5 se utilizan extensamente en el mundo del software para proporcionar la seguridad de que un archivo descargado de Internet no se ha alterado. Comparando una suma MD5 publicada con la suma de comprobación del archivo descargado, un usuario puede tener la confianza suficiente de que el archivo es igual que el publicado por los desarrolladores. Esto protege al usuario contra los 'Caballos de Troya' o 'Troyanos' y virus que algún otro usuario malicioso pudiera incluir en el software. La comprobación de un archivo descargado contra su suma MD5 no detecta solamente los archivos alterados de una manera maliciosa, también reconoce una descarga corrupta o incompleta.

- Para comprobar la integridad de un archivo descargado de Internet se puede utilizar una herramienta MD5 para comparar la suma MD5 de dicho archivo con un archivo MD5SUM con el resumen MD5 del primer archivo. En los sistemas UNIX, el comando

de *md5sum* es un ejemplo de tal herramienta. Además, también está implementado en el lenguaje de *scripting*PHP como MD5("") entre otros.

- En sistemas UNIX y GNU/Linux se utiliza el algoritmo MD5 para calcular el *hash* de las claves de los usuarios. En el disco se guarda el resultado del MD5 de la clave que se introduce al dar de alta un usuario, y cuando éste quiere entrar en el sistema se compara el *hash* MD5 de la clave introducida con el *hash* que hay guardado en el disco duro. Si coinciden, es la misma clave y el usuario será autenticado.
- El MD5 también se puede usar para comprobar que los correos electrónicos no han sido alterados usando claves públicas y privadas.

- **CAPITULO III**

5. **MARCO APLICATIVO**

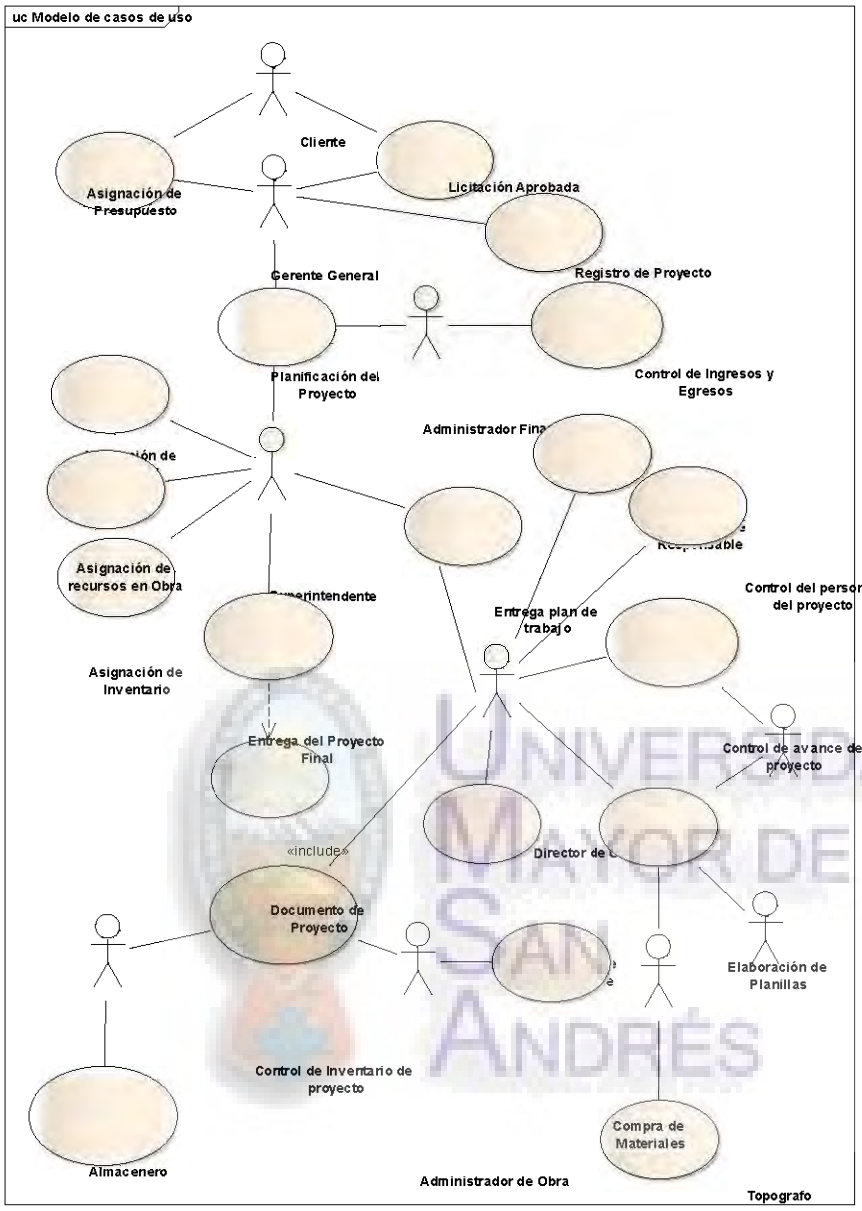
- Para el desarrollo del Sistema Informático de Control de almacenes, maquinarias y de Personal para la empresa constructora INGLOBOL S.A., se utilizara los métodos y herramientas mencionadas en el capitulo anterior.
- El análisis y diseño del sistema se basara en la metodología del Proceso Unificado de Rational (RUP) utilizando el lenguaje Unificado de Modelado (UML) para representar los esquemas de SOFTWARE.

5.1. Modelo de Negocio

- A partir de este flujo de trabajo se tiene un enfoque del funcionamiento y estructura de la empresa INGLOBOL S.A., donde se implementara el sistema.
- Las etapas del modelado de negocio son:
  - Identificar y definir los procesos del negocio, en base al desarrollo de la visión de la empresa en cuanto al modelado de negocio se refiere.
  - Identificar roles y responsabilidades, mediante objetos del negocio.
  - Describir los casos de uso, diagrama de flujo de datos y diagrama de flujo administrativos del negocio, mediante objetos del negocio.

5.1.1. Casos de uso del Negocio

- El siguiente modelo presenta la funcionalidad actual general y especifica de la empresa constructora INGLOBOL, se realiza una abstracción mediante casos de uso del negocio y actores del mismo. Así se permite comprender bien el contexto del que se extraerán los correspondientes casos de uso para el desarrollo del sistema.
-
-



-

- Figura 3.1 caso de uso general [Elaboración propia]

5.1.2. **Descripción de los actores del Negocio**

- Los actores involucrados son:

- Gerente general: será el encargado de asignar el nivel de acceso al personal y contar con los privilegios correspondientes del sistema.
- Gerente de Proyecto: es el encargado de supervisar los proyectos de la empresa que se le asignaron.
- Director de Obras: es el encargado de efectuar el control y seguimiento de la Obra, involucra recursos humanos e inventarios.
- Administrador financiero: Es el encargado de la administración del presupuesto asignado a los proyectos.
-

5.1.3. **Descripción de los casos de uso del Negocio**

-
- El proceso inicia cuando el Cliente solicita los servicios de construcción civil mediante una licitación a la empresa, esta prepara su propuesta y entrega al cliente, quien revisa y la aprueba, por tanto se entrega a la constructora la licitación aprobada y se realiza el contrato, en base a este documento el Gerente General y su equipo tienen la autorización de poder ejecutar el proyecto.
- El Gerente del proyecto registra los datos del proyecto y realiza una planificación del mismo, que consiste en la definición de ítems, costo, tiempo de duración de los ítems, y recursos necesarios para efectuar el proyecto. Con esta planificación el administrador financiero analiza la administración del presupuesto correspondiente al proyecto, en base a los ítems, el tiempo de ejecución de cada uno y los recursos requeridos. Además de realizar un control estricto de los ingresos y egresos del presupuesto, y evitar que se detenga el proyecto por falta de recursos económicos.
- Contando con la planificación del proyecto y el presupuesto del mismo, el Gerente de Proyectos realiza la asignación de un director de Obras, residente de Obras y del personal necesario, recursos en obra e inventarios necesarios, previo conocimiento de

la disponibilidad de los mismos, con esta organización, realiza un plan de proyecto, que será base importante para el buen trabajo del Director de Obras.

- A la conclusión del proyecto, el gerente del Proyecto realiza la respectiva entrega del proyecto con la documentación completa, previa revisión de este, con el cumplimiento de los requerimientos del proyecto se cierra el proyecto.

5.2. **Requerimientos**

- Este flujo permite describir las necesidades o requerimientos del sistema definidas por el cliente y por tal motivo es la mas importante. Esta fase debe identificar y documentar lo que se necesita exactamente de manera entendible.
- Las etapas del modelado de requerimientos son:

✓ Los requerimientos funcionales, que representan la funcionalidad del sistema, se modelan mediante diagramas de casos de uso.

✓ Los requerimientos no funcionales, que se representan los atributos del sistema.

5.2.1. **Requerimientos Funcionales**

- El la siguiente tabla se detalla los requerimientos del usuario de acuerdo a los módulos determinados:
- Tabla 3.1 requerimientos funcionales

• **MODULO**	• **REQUERIMIENTOS**
• ADMINISTRACION DEL PERSONAL	✓ Registrar al personal técnico, obreros y contratistas ✓ Controlar la asistencia del personal ✓ Controlar contratistas ✓ Controlar bonos y anticipos ✓ Emitir reporte de horas de trabajo ✓ Emitir reporte de asistencia diaria ✓ Listado de personal ✓ Libreta de direcciones y teléfonos del

	personal
• CONTROL DE MAQUINARIAS	✓ Registrar los equipos pesados y livianos ✓ Registrar los repuestos de los equipos ✓ Registrar partes diarios de equipos pesados y livianos ✓ Emitir gasto de lubricantes y combustible ✓ Emitir reporte de control de horario ✓ Listado de equipos del proyecto ✓ Libreta de direcciones de operadores y propietarios
• CONTROL DE ALMACENES	✓ Registrar materiales que ingresan y los que salen ✓ Generar reporte de materiales
• COMPRA DE MATERIALES	✓ Registrar costo de materiales y proveedores ✓ Registro de facturas y recibos de las compras ✓ Generar reporte sobre gastos
• AGREGADOS	✓ Registrar agregados y datos de los proveedores y costos ✓ Generar reporte sobre costos totales de los agregados

- Fuente [elaboración propia]

5.2.2. **Requerimientos no Funcionales**

- Se detallan a continuación:

- Tabla 3.2 Requerimientos no funcionales

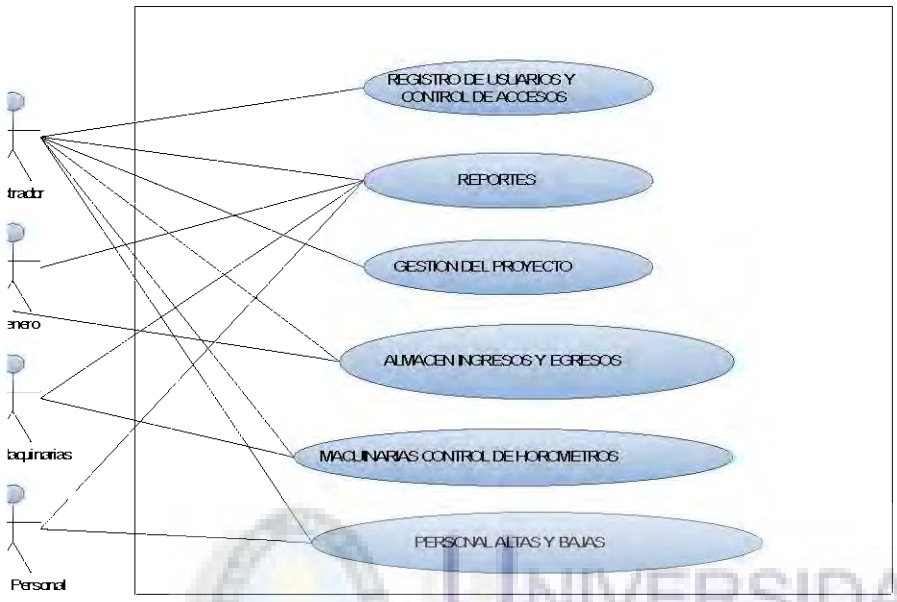
• REQUERIMIENTO	• DETALLES
• SOFTWARE	✓ Sistema operativo XP Profesional, Vista o Windows 7 ✓ Wampserver ✓ MySql, PHP ✓ Navegador Internet Explorer, Netscape o Mozilla FireFox ✓ Antivirus NOD32, Kaspersky o AVAST

	✓ Adobe Acrobat	
• HARDWARE	• MINIMO	• RECOMENDABLE
	➢ Computadora Pentium IV con un procesador de 1.2 GHz ➢ 256 MB en RAM ➢ Disco Duro de 80 GB ➢ Impresora Canon, HP o Epson	➢ Computadora Core i3 o superior ➢ 2 GB en RAM ➢ Disco Duro de 500 GB(casi todas son de 500 GB) ➢ Impresora Laser HP (no multitarea)
• COMUNICACIÓN	✓ Una intranet y acceso ✓ Computadoras libres de algún software de trafico de datos telefónicos, para evitar el funcionamiento incorrecto	

- Fuente [elaboración propia]
-

5.2.3. **Diagrama de casos de uso**

- A continuación se muestran los módulos del sistema mediante los casos de uso de modo general, mediante el cual se muestra los diferentes tipos de usuarios que accederán al sistema en el que podrán interactuar par realizar las diferentes funciones que cumplen cada uno de ellos.
-
-
-
-



-

- Grafico 3.2 Acceso de usuarios [Elaboración propia]

5.2.4. **Documentación CON SU CURSO NORMAL DE EVENTOS**

- Para la documentación se detallan los procesos, es decir se conforman por información breve que describe el proceso, en el curso normal de los eventos que detalla la interacción de los actores y el sistema.

- **CASO DE USO:** Registro de Usuarios Y CONTROL DE ACCESOS

- **Sección:** Principal

- **Actores:** Personal administrativo (inicia actividad), personal.

- **Propósito:** Registrar usuarios en el sistema.

- **Resumen:** el personal se acerca al encargado de personal solicita ser registrado en el sistema, el encargado de personal le solicita su cedula de identidad llena los datos y pregunta por los demás datos para su registro.
- **Tipo:** Primario esencial.

• CURSO NORMAL DE LOS EVENTOS	
• ACCION DE LOS ACTORES	• RESPUESTA DEL SISTEMA
1. Comienza cuando una persona que trabaja en la empresa desea registrarse en el sistema. 2. La presenta su cedula de identidad al encargado e registro de personal. 3. Introduce los datos de la persona.	4. Se abre la pantalla de registro de personas. 5. Se guarda los datos de la persona.

- **CASO DE USO:** REPORTES
- **Sección:** Principal
- **Actores:** Personal administrativo (inicia actividad), personal.
- **Propósito:** Registrar usuarios en el sistema.
- **Resumen:** el personal se acerca al encargado de personal solicita ser registrado en el sistema, el encargado de personal le solicita su cedula de identidad llena los datos y pregunta por los demás datos para su registro.
- **Tipo:** Primario esencial.

• CURSO NORMAL DE LOS EVENTOS	
• ACCION DE LOS ACTORES	• RESPUESTA DEL SISTEMA

1. Comienza cuando una persona que trabaja en la empresa desea emitir un reporte 2. Realiza la acción de listar alguna información que desee del sistema como ser listado de materiales por frente de trabajo. 3. Presiona el botón emitir reporte.	4. Se abre la pantalla de listado de materiales por frente. 5. El sistema emite el reporte para imprimir.

-
- **CASO DE USO:** Gestión de Proyectos
- **Sección:** Principal
- **Actores:** Personal administrativo (inicia actividad)
- **Propósito:** Registrar de frentes de trabajo, listado de frentes y muestra el total de materiales por frente.
- **Resumen:** el encargado de este modulo lista los frentes los materiales por frente y también se registra los nuevos frentes de trabajo que sean aperturados.
- **Tipo:** Primario esencial.

• CURSO NORMAL DE LOS EVENTOS	
• ACCION DE LOS ACTORES	• RESPUESTA DEL SISTEMA
1. Comienza cuando el administrador ingresa al sistema luego de loguearse. 2. El administrador elige listado de frentes.	6. Se abre la pantalla de listado por frentes registro de frentes y listado de frentes y material más contratistas. 7. Se guarda los datos del frente de

3. Introduce los datos del frente de trabajo. 4. Se elige la opción de registro de frente de trabajo. 5. Se elige la opción listado de frentes materiales y contratistas.	trabajo. 8. Se muestra una pantalla con los datos del registro. 9. Finaliza la acción.

- **CASO DE USO:** Registro de materiales
- **Sección:** Principal
- **Actores:** materiales que ingresan en almacenes de la empresa, Encargado de almacenes.
- **Propósito:** Registrar los materiales que ingresan en almacenes.
- **Resumen:** el almacenero registra en el sistema los datos del material que ingresa mediante vista de material existente caso contrario se lo registra como nuevo.
- **Tipo:** Primario esencial.

• CURSO NORMAL DE LOS EVENTOS	
• ACCION DE LOS ACTORES	• RESPUESTA DEL SISTEMA
1. Comienza cuando una persona lega con materiales al almacén entonces el almacenero registra los datos del material que ingresa, nombre, cantidad y fecha. 2. El almacenero registra los datos del material que ingresa, nombre, cantidad y fecha.	3. Se abre la pantalla de registro de materiales. 4. Se guarda los datos de la persona.

- **CASO DE USO:** Registro de maquinarias
  - **Sección:** Principal
  - **Actores:** Personal de la empresa obrero o técnico (inicia actividad), Encargado de Personal.
  - **Propósito:** Registrar al personal en el sistema.
  - **Resumen:** el personal se acerca al encargado de personal solicita ser registrado en el sistema, el encargado de personal le solicita su cedula de identidad llena los datos y pregunta por los demás datos para su registro.
  - **Tipo:** Primario esencial.

• CURSO NORMAL DE LOS EVENTOS	
• ACCION DE LOS ACTORES	• RESPUESTA DEL SISTEMA
10. Comienza cuando una persona que trabaja en la empresa desea registrarse en el sistema. 11. La presenta su cedula de identidad al encargado e registro de personal. 12. Introduce los datos de la persona.	13. Se abre la pantalla de registro de personas. 14. Se guarda los datos de la persona.

- **CASO DE USO:** Registro de personal
  - **Sección:** Principal
  - **Actores:** materiales que ingresan en almacenes de la empresa, Encargado de almacenes.
  - **Propósito:** Registrar los materiales que ingresan en almacenes.

- **Resumen:** el almacenero registra en el sistema los datos del material que ingresa mediante vista de material existente caso contrario se lo registra como nuevo.
- **Tipo:** Primario esencial.

• CURSO NORMAL DE LOS EVENTOS	
• ACCION DE LOS ACTORES	• RESPUESTA DEL SISTEMA
5. Comienza cuando una persona lega con materiales al almacenenonces el almacenero registra los datos del material que ingresa, nombre, cantidad y fecha. 6. El almacenero registra los datos del material que ingresa, nombre, cantidad y fecha.	7. Se abre la pantalla de registro de materiales. 8. Se guarda los datos de la persona.

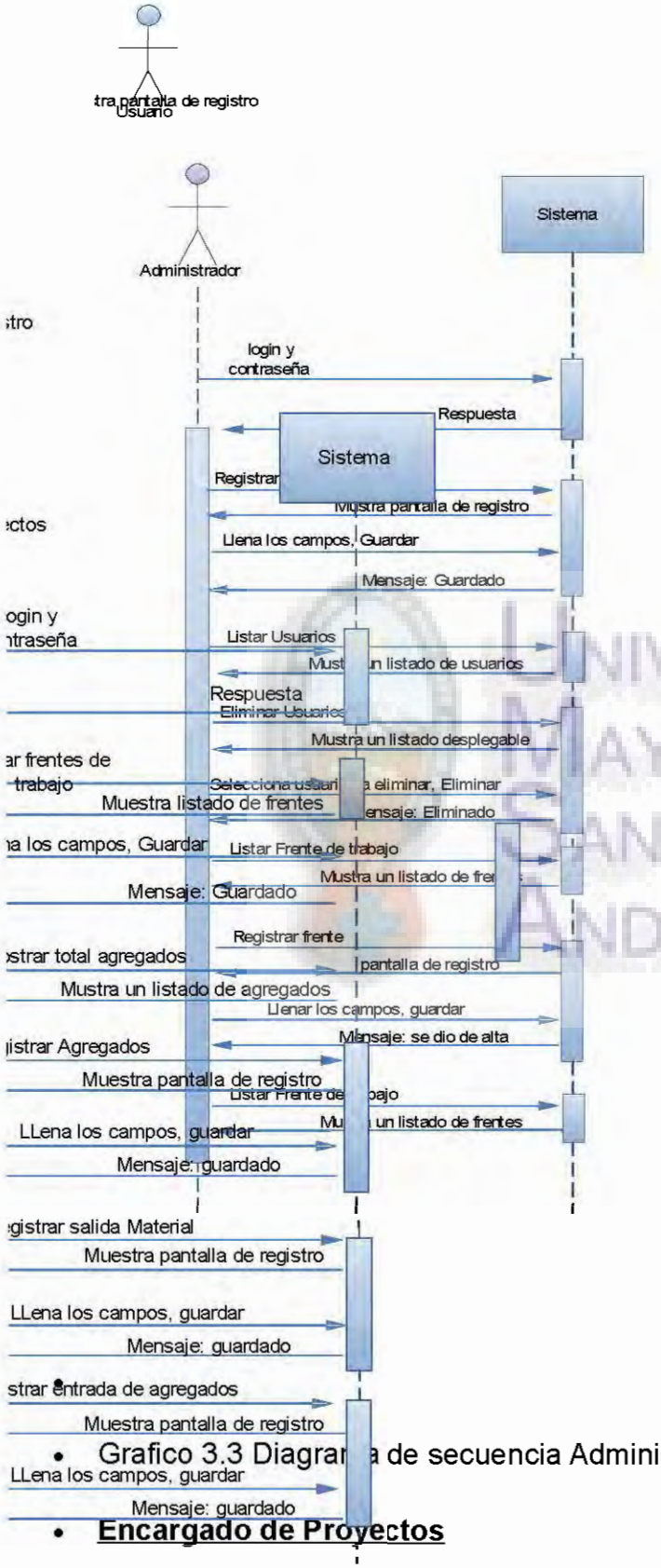
-
-

5.2.5. **Diagrama de secuencia de sistemas**

- Este diagrama presenta la interacción existente entre las clases para describir la funcionalidad de los casos de uso, mostrando el orden entre eventos del sistema, y a continuación se muestra interacciones entre clases ya identificadas:

-
-
-
-
-
-

-
-
-
- **Administrador del Sistema**

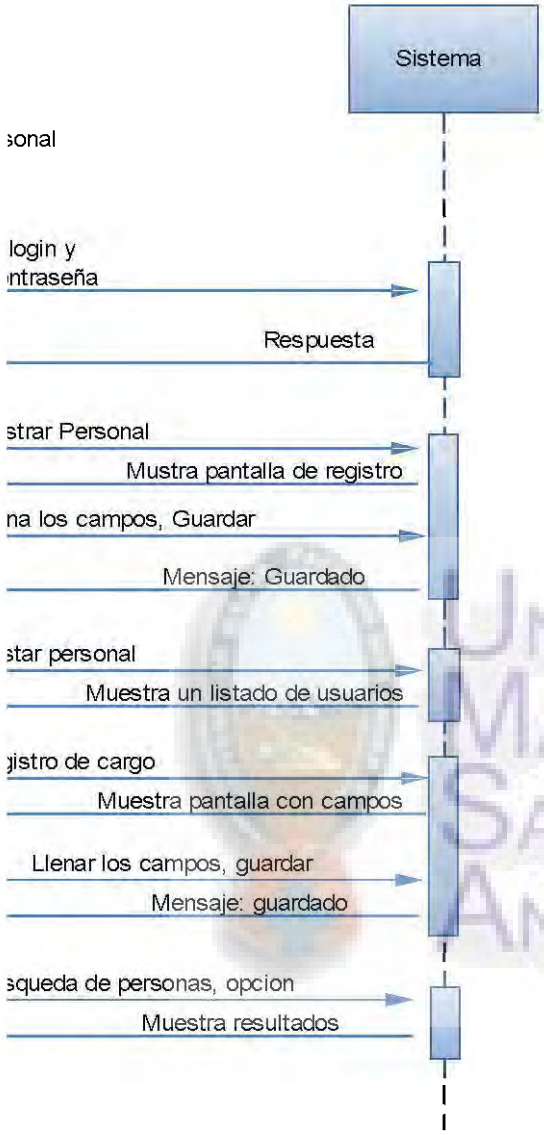


- Grafico 3.3 Diagrama de secuencia Administrador [Elaboración propia]

- **Encargado de Proyectos**

- Grafico 3.4 Diagrama de secuencia Encargado de Proyectos [Elaboración propia]
- **Encargado del personal**

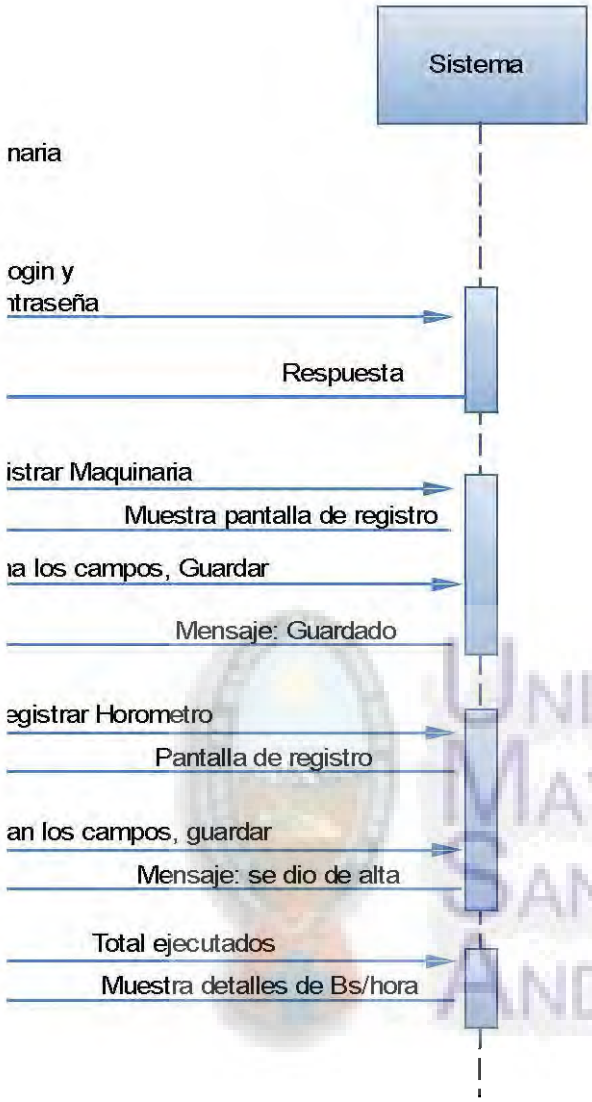
stro



-
- Grafico 3.4 Diagrama de secuencia Encargado del personal [Elaboración propia]
- **Encargado de almacén**

tro

-
- Grafico 3.5 Diagrama de secuencia Encargado del almacén [Elaboración propia]
- **Encargado de Maquinarias**



-
- Grafico 3.6 Diagrama de secuencia Encargado del maquinaria [Elaboración propia]
-

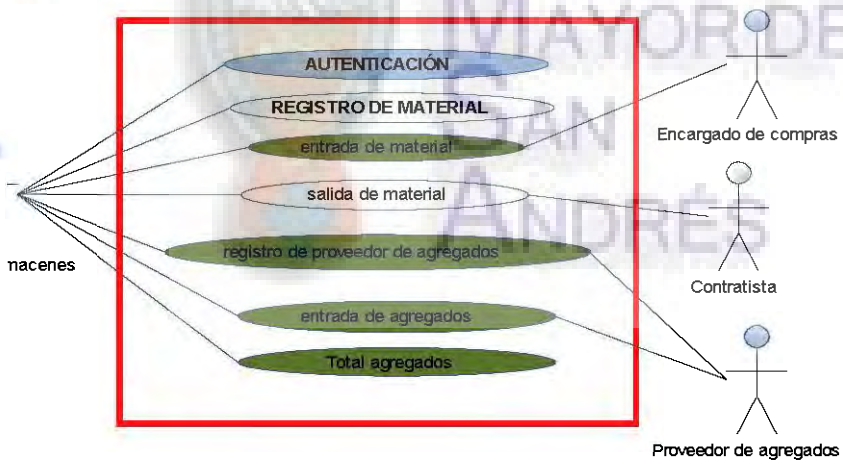
5.3. **DISEÑO**

5.3.1. DIAGRAMA DE CASOS DE USO REAL

- **Caso de Uso Real: Encargado del Personal**

- Figura 3.7: Caso de Uso Encargado del personal [Elaboración Propia]

- **Caso de Uso Real: Encargado de Almacenes**



- Figura 3.8: Caso de Uso Encargado del almacenes [Elaboración Propia]

- **Caso de Uso Real: Encargado de maquinaria**

- Figura 3.9: Caso de Uso Encargado del maquinaria [Elaboración Propia]

- **Caso de Uso Real: Administrador del Sistema**

- Figura 3.10: Caso de Uso administrador del sistema [Elaboración Propia]

-

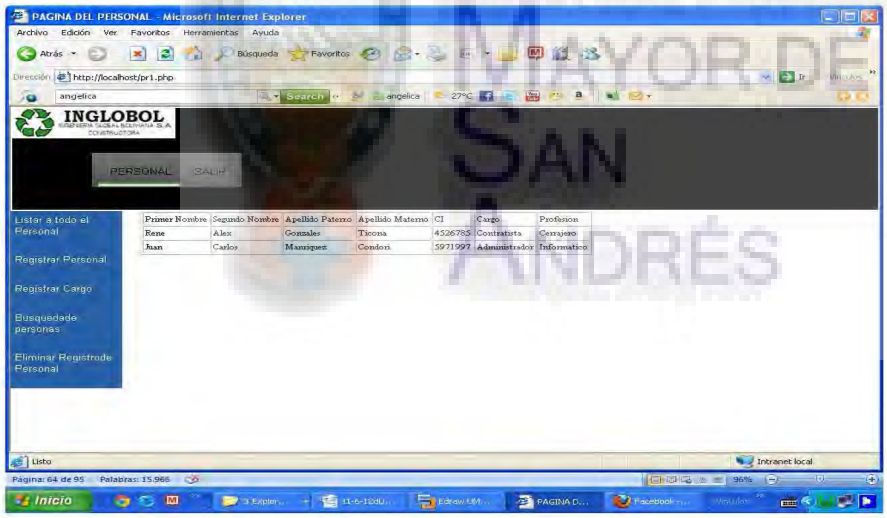
5.3.1.1. FORMATOS DE PANTALLAS

- **AUTENTICACION:** La siguiente pantalla muestra dos campos el del usuario y el password los cuales darán acceso al sistema.

- Esta es la pantalla de Logueo en la cual el usuario debe autenticarse introduciendo su nombre de usuario y su respectivo password.
- Una vez introducido el usuario y password correctos el sistema verificando al usuario mostrara una pantalla de bienvenida para iniciar con la sesión del usuario.
- Al tratarse de un sistema que tiene control de accesos la pantalla principal limitara el acceso a los distintos módulos dependiendo del usuario.
- **CASO DE USO:** AUTENTICACIÓN
- **Sección:** Principal
- **Actores:** Usuario del sistema, Administrador, Encargado de Personal, encargado de almacenes y encargado de maquinarias.
- **Propósito:** Ingresar al sistema.
- **Resumen:** El usuario ingresa su username y password el sistema verifica los datos del mismo si es correcto da la bienvenida para luego accesar al sistema con sus respectivos módulos correspondientes según su nivel de acceso.
- **Tipo:** Primario esencial.

• CURSO NORMAL DE LOS EVENTOS	
• ACCION DE LOS ACTORES	• RESPUESTA DEL SISTEMA
1. Comienza cuando un usuario ingresa sus datos al sistema es decir el nombre de usuario y password. 2. El usuario accede al sistema según su nivel de acceso. 3. El usuario puede elegir entre los módulos de personal, almacén, agregados o proyecto.	4. El sistema verifica los datos del usuario en la base de datos si son correctos acepta y da acceso al usuario. 5. De esta manera el usuario puede acceder a los distintos niveles del sistema.

-
- **ENCARGADO DEL PERSONAL**
-
-
-
-
-
-
-
-



- Iniciaremos con el usuario que tiene acceso al modulo personal.
- **CASO DE USO:** Encargado del personal
- **Sección:** Principal
- **Actores:** Encargado del Personal, Personal de la empresa
- **Propósito:** Registrar al personal que trabaja en la empresa.

- **Resumen:** El encargado del personal se encarga del registro de las personas que trabajan en la empresa o en el proyecto.
- **Tipo:** Primario.

• CURSO NORMAL DE LOS EVENTOS	
• ACCION DE LOS ACTORES	• RESPUESTA DEL SISTEMA
1. Comienza cuando el encargado de personal ingresa al sistema. • 3. El encargado de personal ingresa nombre de usuario y password. • 5. Elige Listar todo el personal. • 7. Elige la opción registrar personal. • • 9. Se elige la opción guardar. • 11. Elige la opción registrar cargo. • 13. Se elige la opción guardar. • 15. Se elige la opción de búsqueda de personas. • 17. El usuario ingresa los datos según su criterio y elige la opción buscar. • 19. Elige la opción de eliminar registro de personal. • 21. Se elige la opción eliminar.	2. Se abre la pantalla de Autenticación. • 4. el sistema acepta los datos y muestra la pantalla del encargado de personal. • • 6. se lista todo el personal registrado. • 8. Se abre la pantalla de registro del personal, se llenan los campos del registro. • 10. El sistema guarda los nuevos datos. • 12. Se abre la pantalla de registro de cargos, se llenan los campos del registro. • 14. El sistema guarda los nuevos datos. • 16. Se abre la pantalla de búsqueda según los criterios de: CI, nombre, apellido paterno o apellido materno. • 18. El sistema muestra los datos encontrados. • 20. muestra la pantalla con los

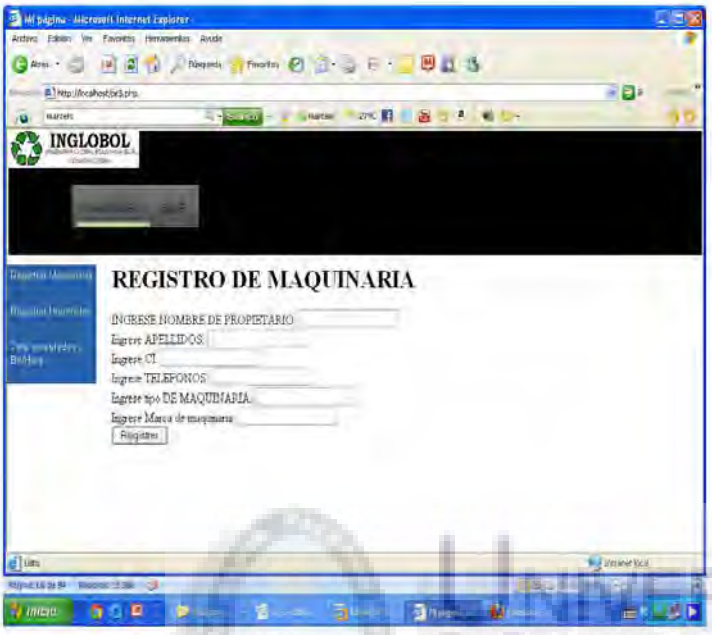
	datos del personal a ser seleccionados para eliminar. • 22. El sistema elimina el registro.

-
- **ENCARGADO DE ALMACENES**
- 

- **CASO DE USO:** Encargado de Almacenes
- **Sección:** Principal
- **Actores:** Encargado de Almacenes, Encargado de Compras, Proveedor de agregados.
- **Propósito:** Registrar los materiales que ingresan en los almacenes.
- **Resumen:** el almacenero registra en el sistema los datos del material que ingresa en los almacenes los cuales son entregados por el encargado de compras.
- **Tipo:** Primario.

• CURSO NORMAL DE LOS EVENTOS	
• ACCION DE LOS ACTORES	• RESPUESTA DEL SISTEMA

1. Comienza cuando el encargado de almacenes ingresa al sistema. • 3. El encargado de almacenes ingresa nombre de usuario y password. • 5. Elige la opción registro de material. • 7. Elige el botón registrar. • 9. Se elige la opción guardar. • 11. Elige la opción entrada de material. • 13. Se llenan los campos y se elige la opción registro. • 15. Se elige la opción de entrada de material. • 17. Se llenan los campos y luego se elige la opción registrar. • 18. Elige la opción de salida de material. • • 20. Se elige la opción registrar. • 22. Elige la opción de registro de proveedor de agregados. • 24. El proveedor de agregados dicta sus datos personales para ser llenados en los campos del sistema luego elige guardar. • 26. Elige la opción entrada de agregados. • 28. El Usuario llena los campos y luego elige registrar. • 30. elige la opción Total	2. Se abre la pantalla de Autenticación. • 4. el sistema acepta los datos y muestra la pantalla del encargado de almacenes. • 6. muestra los campos a ser llenados. • 8. Se guarda el registro del material. • 10. El sistema guarda los nuevos datos. • 12. Se abre la pantalla de registro de salida de material con una opción de registro. • 14. El sistema guarda los nuevos datos. • • 16. Se abre la pantalla de registro de material. • 18. El sistema guarda los datos. • 19. muestra la pantalla con los datos de contratistas y del material para ser guardaos con los campos correspondientes. • 21. El sistema guarda los datos de la salida de material. • 23. El sistema muestra los campos a ser llenados para el proveedor de agregados. • 25. El sistema guarda los datos. • 27. El sistema muestra la pantalla con los campos para el registro de

-
- **Encargado de maquinarias**
- 

- **CASO DE USO:** Encargado de Maquinarias
- **Sección:** Principal
- **Actores:** Encargado de maquinarias, Propietario de maquinaria.
- **Propósito:** Registrar los vehículos de maquinaria pesada que trabajen en el proyecto de la empresa.
- **Resumen:** El encargado de la maquinaria realiza el registro de maquinarias que trabajan en la empresa asi también de los horometros.
- **Tipo:** Primario.

• CURSO NORMAL DE LOS EVENTOS	
• ACCION DE LOS ACTORES	• RESPUESTA DEL SISTEMA
1. Comienza cuando el encargado de maquinarias ingresa al sistema.	2. Se abre la pantalla de Autenticación.

• 3. El encargado de almacenes ingresa nombre de usuario y password. • 5. Elige la opción registro de maquinaria. • 7. Elige el botón registrar. • 9. Se elige la opción Registro de horometros. • 11. Se llenan los campos conciliados con el encargado de maquinaria o propietario se elige la opción registrar. • 13. Se elige la opción Total Ejecutados. •	• 4. el sistema acepta los datos y muestra la pantalla del encargado de maquinarias. • • 6. muestra los campos a ser llenados. • 8. Se guarda el registro de la maquinaria. • 10. El sistema muestra los campos para registro de horometro. • 12. El sistema guarda el registro. • 14. El sistema muestra los datos de las maquinarias y sus ejecuciones de trabajo hasta la fecha.

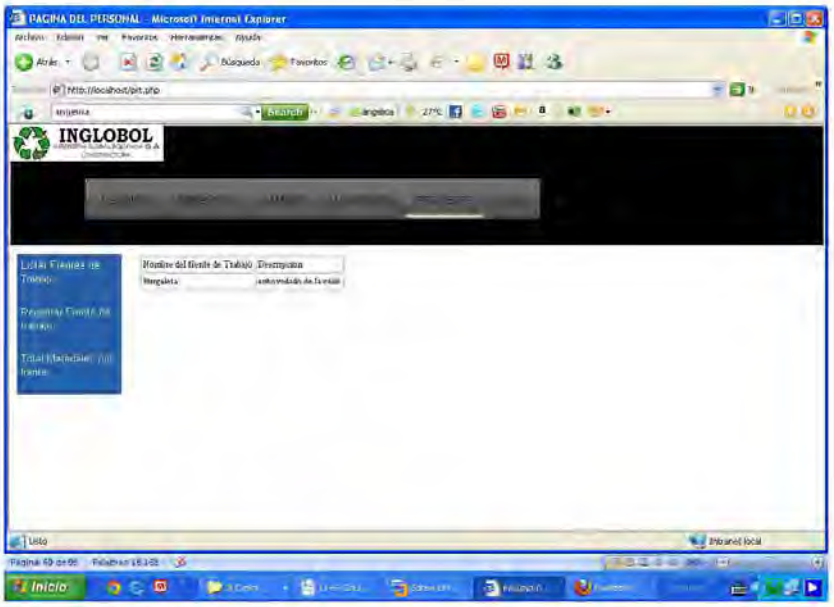
-
- Usuario ADMINISTRADOR DE USUARIOS
- 

- **CASO DE USO:** ADMINISTRADOR DE USUARIOS
- **Sección:** Principal
- **Actores:** ADMINISTRADOR, Usuario
- **Propósito:** Registrar los usuarios que tendrán acceso al sistema.

- **Resumen:** El Administrador será el único que tenga acceso a todos los módulos de control, el se encarga de registrar nuevos usuarios para el manejo de los respectivos módulos del sistema.
- **Tipo:** Primario.

• CURSO NORMAL DE LOS EVENTOS	
• ACCION DE LOS ACTORES	• RESPUESTA DEL SISTEMA
1. Comienza cuando el administrador ingresa al sistema. • 3. El administrador ingresa nombre de usuario y password. • 5. Elige la opción listar usuarios. • 7. Elige la opción registrar nuevo usuario. • 9. Se llenan los campos con datos del nuevo usuario y se elige la opción Registrar. • 11. Se elige la opción eliminar usuario. • 13. Se elige un usuario a ser eliminado luego la opción eliminar. •	2. Se abre la pantalla de Autenticación. • 4. el sistema acepta los datos y muestra la pantalla del encargado de maquinarias. • • 6. muestra los usuarios con acceso. • 8. Se muestran los campos a ser llenados para registrar un nuevo usuario. • 10. El sistema guarda los datos y el nuevo usuario ya puede loguearse. • 12. El sistema muestra datos de los usuarios activos. • 14. El sistema elimina el registro del usuario.

•

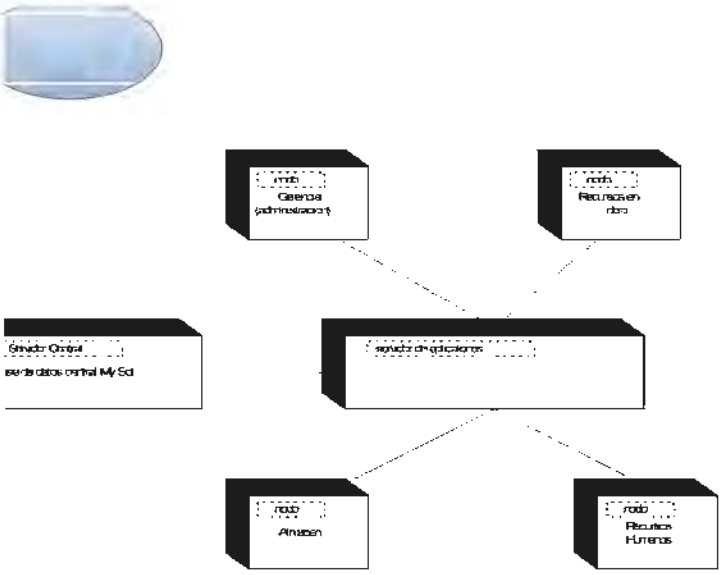
- 

- **CASO DE USO:** ADMINISTRADOR DE PROYECTOS
- **Sección:** Principal
- **Actores:** ADMINISTRADOR, residente de obra.
- **Propósito:** Registrar los frentes de trabajo y ver los materiales que se han usado.
- **Resumen:** El Administrador será el único que tenga acceso a todos los módulos de control, el se encarga de registrar nuevos frentes de trabajo con ayuda del residente de obra.
- **Tipo:** Primario.

• CURSO NORMAL DE LOS EVENTOS	
• ACCION DE LOS ACTORES	• RESPUESTA DEL SISTEMA
3. Comienza cuando el administrador ingresa al sistema. • 3. El administrador ingresa nombre de usuario y password. • 5. Elige la opción listar frentes de trabajo.	4. Se abre la pantalla de Autenticación. • 4. el sistema acepta los datos y muestra la pantalla del encargado de maquinarias.

• 7. Elige la opción registro de nuevo frente. • 9. Se llenan los campos con datos del nuevo frente de trabajo y se elige la opción Registrar. • 11. Se elige la opción eliminar usuario. • 13. Se elige la opción total materiales por frente. •	• • 6. muestra los frentes de trabajo. • 8. Se muestran los campos a ser llenados para registrar un nuevo frente. • 10. El sistema guarda los datos y el nuevo frente de trabajo. • 12. El sistema muestra datos de los usuarios activos. • 14. El sistema muestra los datos de materiales por frente de trabajo.

5.3.1.2. **DIAGRAMAS DE DESPLIEGUE**

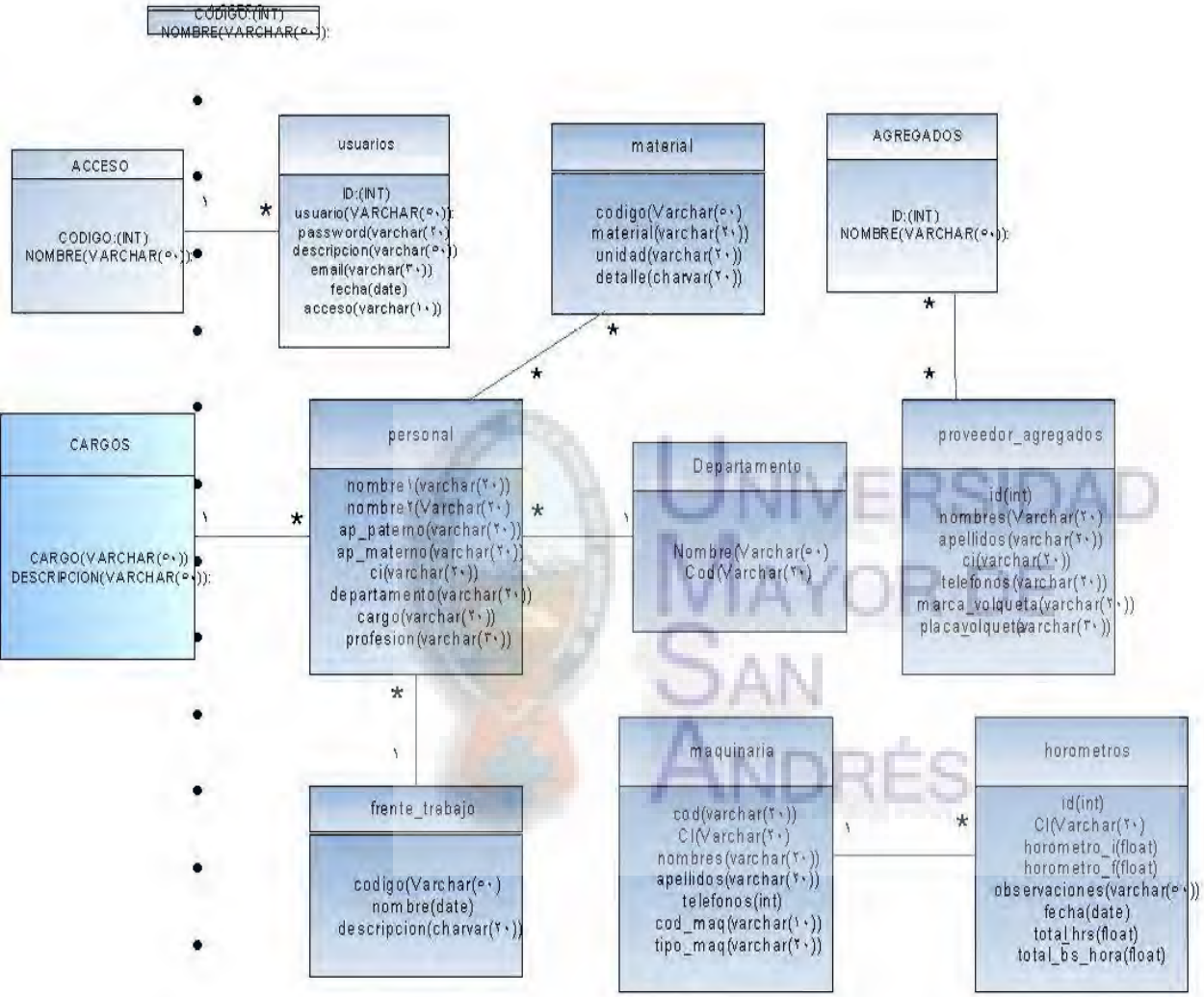


- Figura 3.11 Diagrama de despliegue [elaboración propia]

5.3.1.3. DIAGRAMAS DE PAQUETE

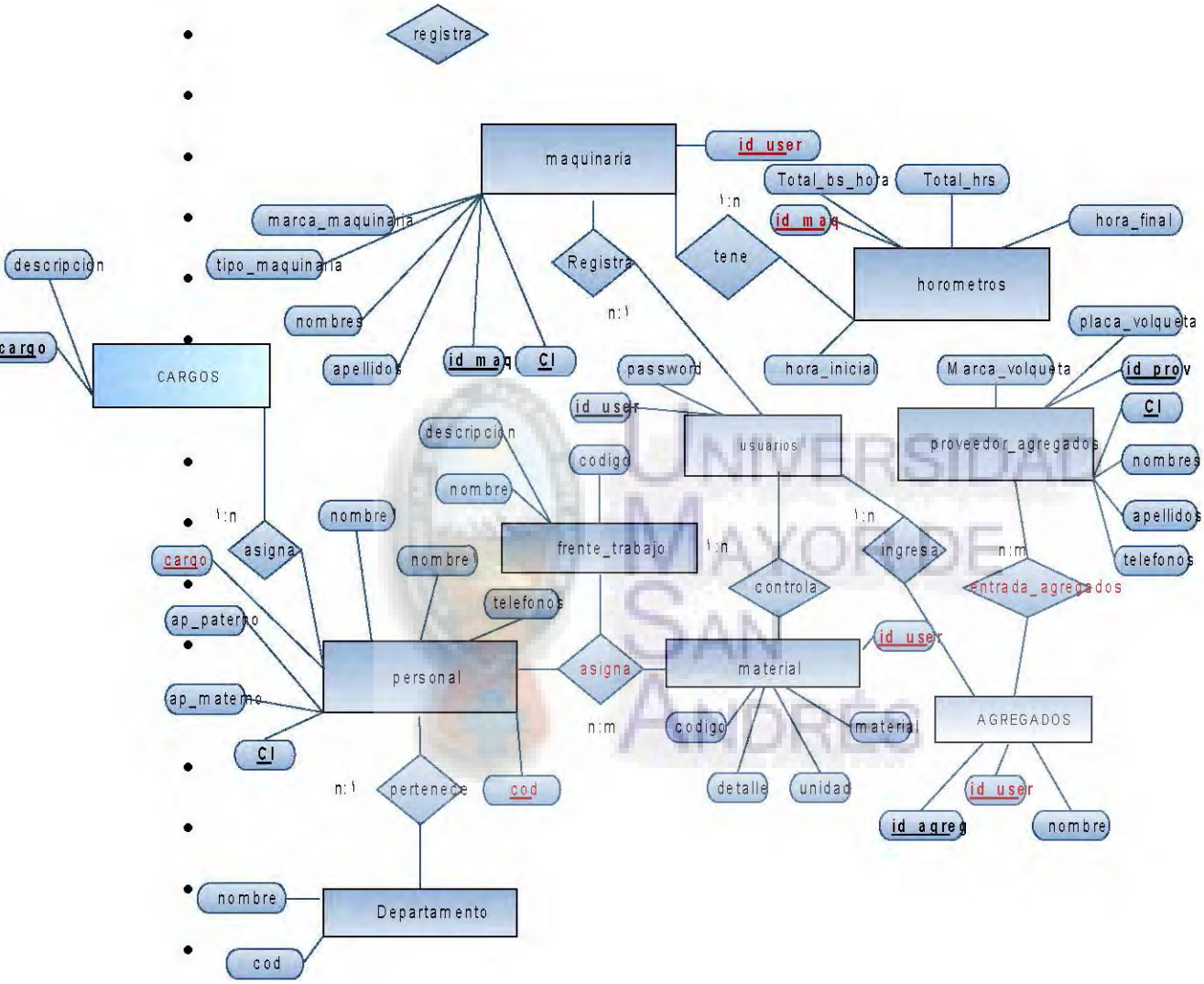
- Figura 3.12 Diagrama de despliegue [elaboración propia]

5.3.2. **DIAGRAMA DE CLASES**



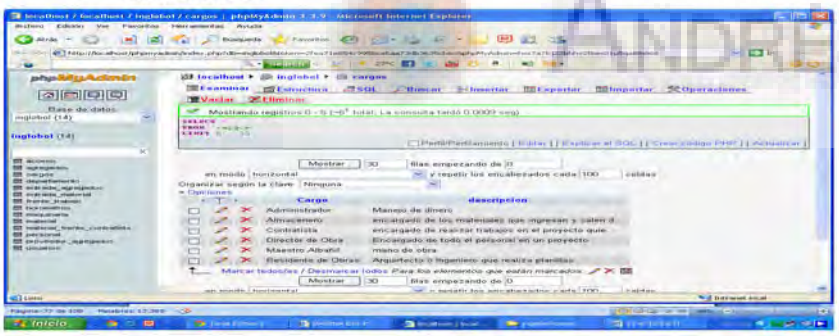
- Figura 3.13 Diagrama de clases [elaboración propia]

5.3.3. **MODELO ENTIDAD-RELACION**


registra
maquinaria
id_user
Total_bs_hora
Total_hrs
1:n
id_maq
hora_final
marca_maquinaria
tipo_maquinaria
descripcion
Registra
tene
horometros
nombres
n:1
placa_volqueta
cargo
CARGOS
apellidos
id_maq
CI
password
hora_inicial
Marca_volqueta
id_prov
id_user
usuarios
proveedor_agregados
CI
descripcion
codigo
nombres
nombre
1:n
nombre
asigna
1:n
ingresa
n:m
apellidos
cargo
nombre
frente_trabajo
1:n
telefonos
controla
entrada_agregados
ap_paterno
telefonos
id_user
personal
asigna
material
ap_materno
n:m
codigo
material
AGREGADOS
CI
n:1
pertenece
cod
detalle
unidad
id_user
id_agreg
nombre
nombre
Departamento
cod

-
-
-
-
-
- Figura 3.14 Modelo E/R [elaboración propia]
- Mediante el modelo físico se muestran las tablas y su composición.
- **TABLAS**
- AGREGADOS

- CARGOS



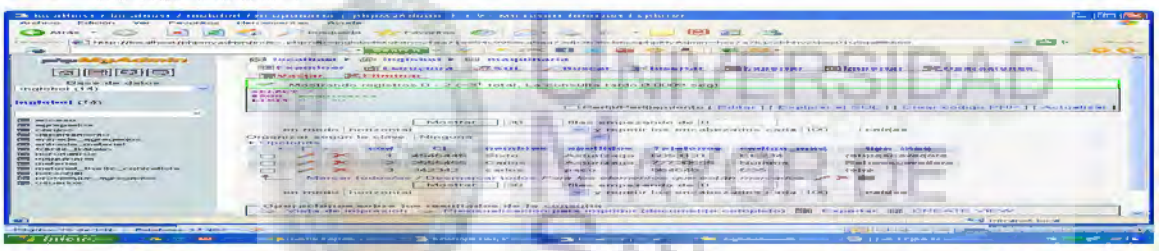
- DEPARTAMENTO

-

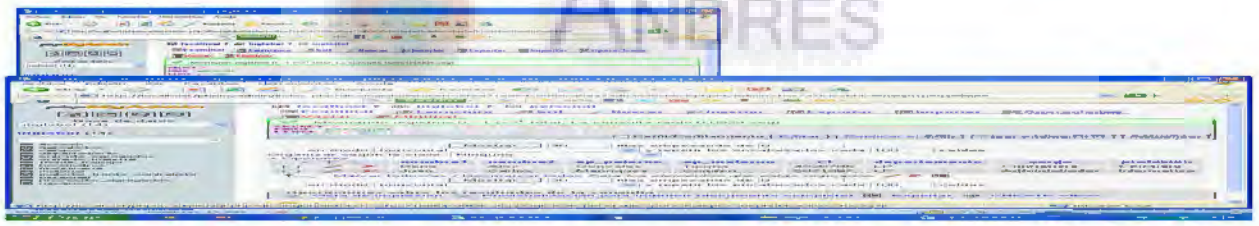
- FRENTE_TRABAJO

- CARGOS

- MAQUINARIA



- MATERIAL

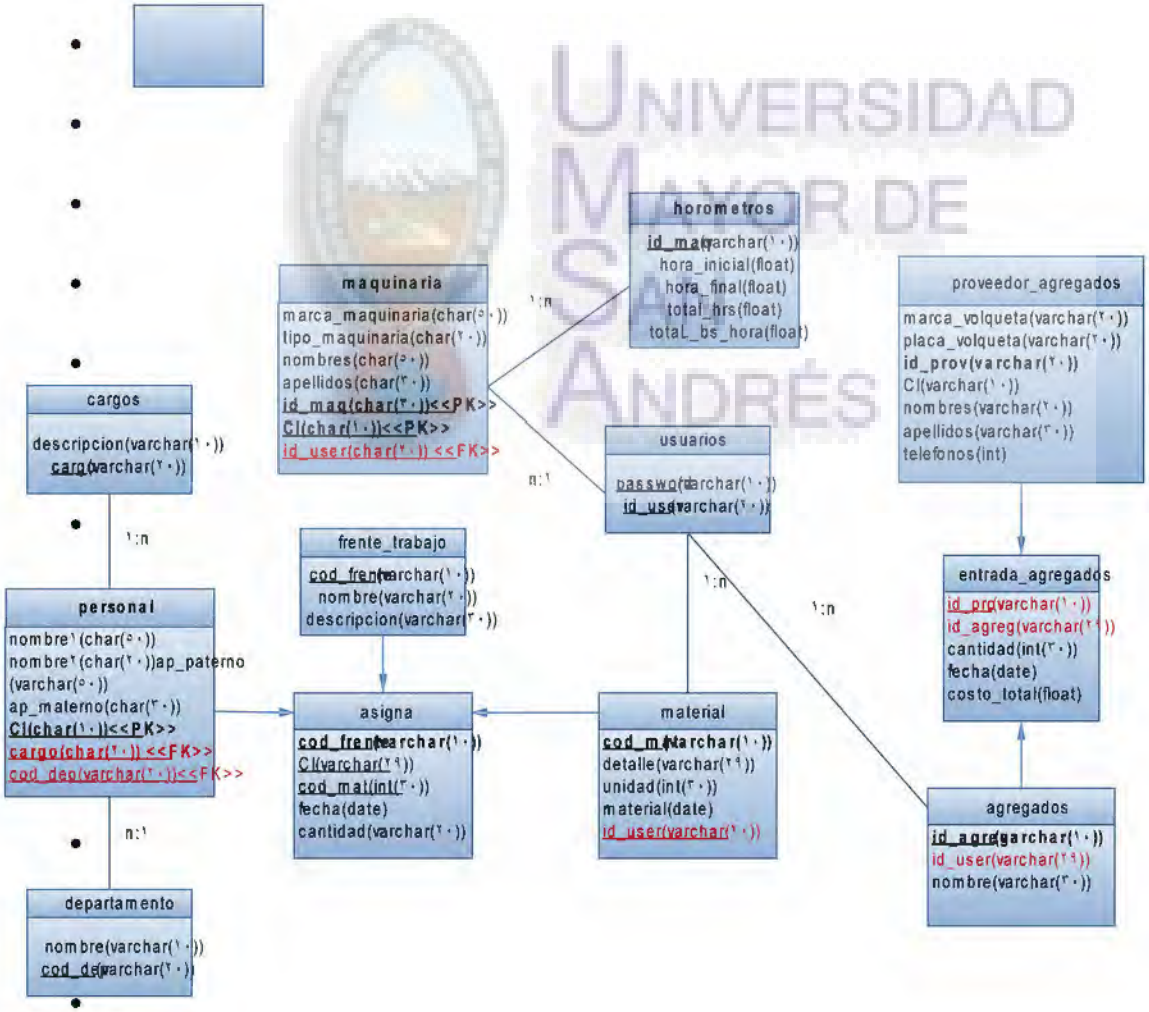


- PROVEEDOR_AGREGADOS

- ASIGNA (MATERIAL_FRENTE_CONTRATISTA)

- 
- ENTRADA_AGREGADOS

- **MODELO FISICO NORMALIZADO**



- Grafico 3.9 Fuente [elaboración propia]

5.3.4. **DIAGRAMA DE COMPONENTES**

- Este diagrama permite modelar la vista estática del sistema, describiendo los componentes (módulos) y sus interfaces.

-
-
-
- Figura 3.15 Diagrama de Componentes [elaboración propia]

5.4. **Pruebas**

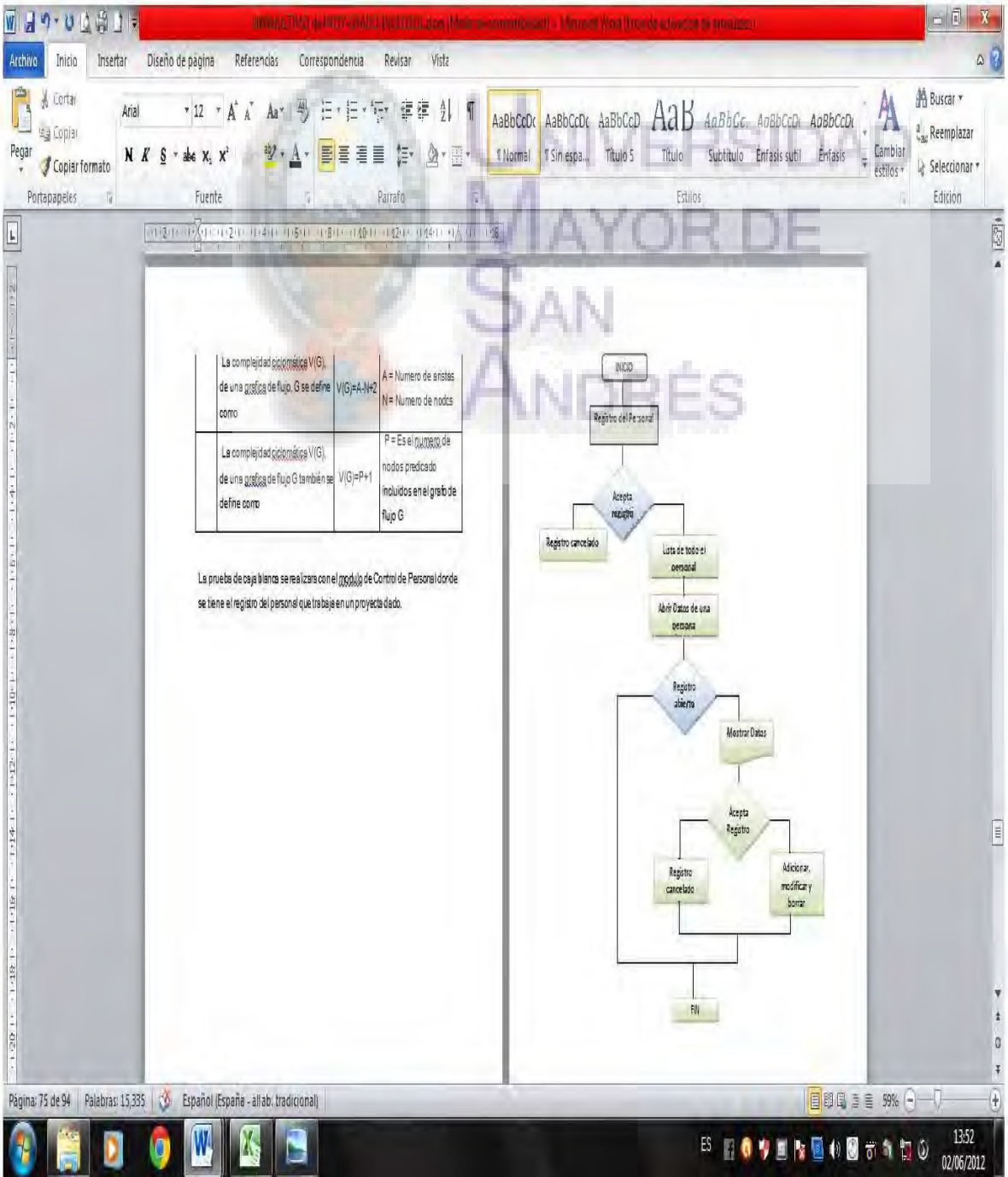
5.4.1. **Pruebas de caja blanca**

- La prueba de caja blanca se efectúa previamente en el proceso de prueba y la prueba del camino básico es una técnica de esta prueba, este método del camino básico permite obtener una medida de la complejidad lógica de un diseño procedimental y se puede usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución y de estos caminos básico los casos de prueba que se obtuvieran deben garantizar que durante la prueba se ejecuta al menos una vez cada sentencia del programa.
- **COMPLEJIDAD CICLOMATICA**
- La complejidad ciclomática es una métrica de software que proporciona una medida cuantitativa de la complejidad lógica de un programa.
- La complejidad ciclomática se basa en la teoría de grafos y se puede calcular de las tres formas siguientes:

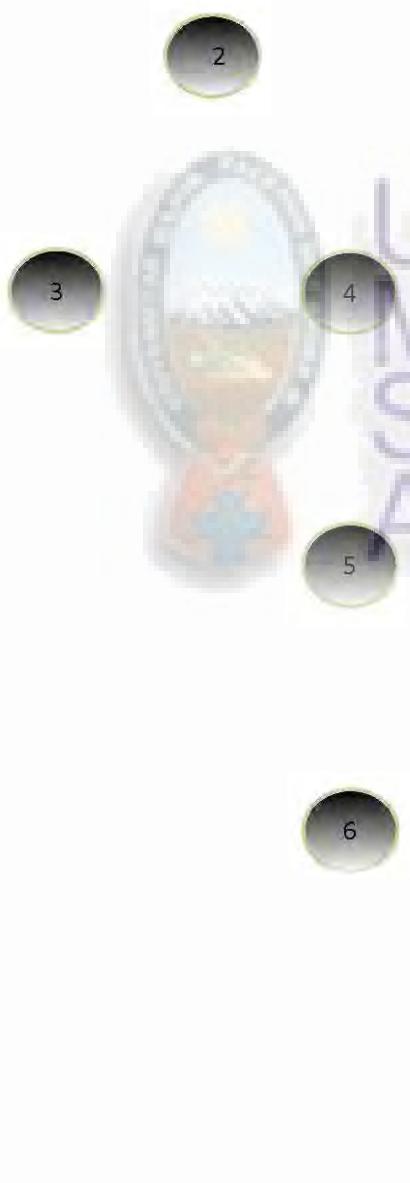
• **N**	• **COMPLEJIDAD CICLOMATICA**	• **FOR MULA**	• **DONDE**
• 1	• El numero de regiones corresponde a la complejidad ciclomática	•	•
•	• La complejidad ciclomática V(G), de una grafica de flujo, G se define como	• V(G)= A-N+2	• A = Numero de aristas • N = Numero de

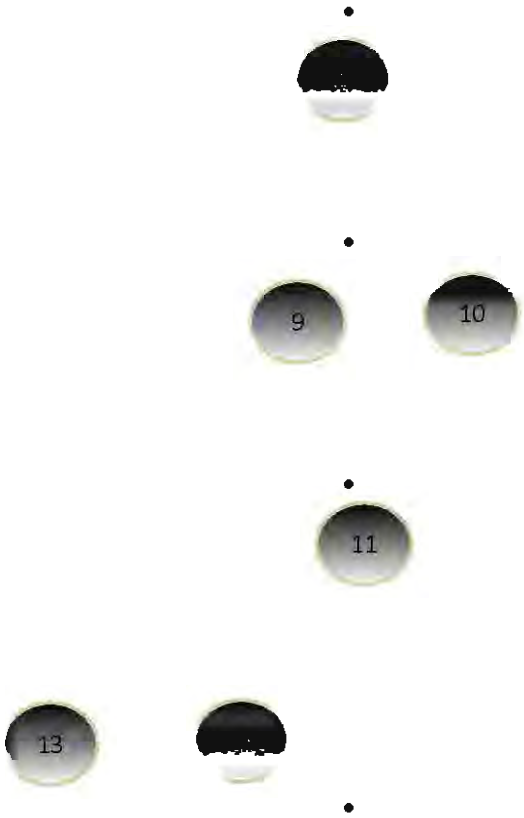
			nodos
•	• La complejidad ciclomática V(G), de una grafica de flujo G también se define como	• V(G) =P+1	• P = Es el numero de nodos predicado incluidos en el grafo de flujo G

- La prueba de caja blanca se realizara con el modulo de Control de Personal donde se tiene el registro del personal que trabaja en un proyecta dado.
-



- Figura 3.16 Modelo de caja blanca [elaboración propia]





- Figura 3.17 Análisis Por grafos [elaboración propia]

- De acuerdo al flujo se obtienen los siguientes caminos independientes:
- Camino 1: 1-2-4-5-6-7-9-10-11-12
- Camino 2: 1-2-4-5-6-7-8-10-11-12
- Camino 3: 1-2-4-5-11
- Camino 4: 1-2-3-12
- Este resultado que se tiene cuatro casos de prueba.
- La formula de la complejidad ciclomatica es:

- V(G) = A-N+2 Donde: A= Numero de aristas
- N= Numero de nodos
- P= Numero de nodos predicados
- Reemplazando valores: V(G) = 15 – 13 + 2= 4
- V(G) = P +1 Donde: P = Nodos predicados
- Reemplazando valores: V(G) = 3 + 1 = 4
- Conforme a los resultados obtenidos La complejidad es V(G) = 4

5.4.2. **Pruebas de caja negra**

- La prueba de caja negra se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de caja blanca, y a diferencia de esta la prueba de caja negra se efectúa durante fases posteriores de la prueba.
- Caso de prueba: Control de Maquinarias
- Entradas: - Previo registro de datos del equipo
- Resultados: - Se le asigna a un proyecto
- - Se lo controla mediante el registro de los partes diarios
- - Se registran datos de repuestos
- - Se emite un informe del funcionamiento de los equipos
- Condición: - Se debe indicar si el equipo es propio o alquilado
- Procedimiento de prueba
- Ya definido el caso de prueba se debe verificar si los resultados obtenidos son los requeridos, para tal motivo se utilizara el caso de uso de control de recursos en obra.
- Caso de USO: Control de maquinarias

1. Muestra la pantalla principal
2. El usuario debe digitar cuenta de usuario y password, presionar “Aceptar”
3. El sistema valida al usuario
4. Si la validación es correcta, el usuario ingresa la pantalla con el menú del sistema
5. El usuario debe presionar “Maquinaria”
6. El sistema muestra otra pantalla con el menú de maquinarias
7. El usuario debe presionar “Añadir nueva maquinaria”.
8. El sistema muestra otra pantalla con espacios o casillas de texto donde debe llenar los datos de la nueva maquinaria luego presiona el botón de “guardar”.
9. El usuario presiona parte diario
10. El sistema muestra una pantalla donde se debe llenar los datos de la maquinaria de la que se desea saber su parte diario o elegir opciones de un listado general.
11. También en esta pantalla hay la opción de registrar un nuevo parte diario, se presiona el botón añadir, se muestra otra pantalla donde se registra los datos de la maquinaria, se presiona “guardar”.

- Conforme a los pasos descritos anteriormente en la prueba, se puede verificar que si cumple y se obtiene los resultados por tanto se afirma que el proceso funciona correctamente.

- **CAPITULO IV**

- A la conclusión del desarrollo del sistema es menester conocer la calidad del sistema siendo este un factor importante, la gestión

- **4.1. METRICAS DE CALIDAD ISO 9126**

- A la conclusión del desarrollo del sistema de información se debe conocer la calidad obtenida del sistema ya que es un factor importante por tanto esta calidad se basara

en la descripción de las dimensiones de calidad especificas en el capitulo II, la medición se realizara mediante métricas ISO 9126.

- Si bien no se llega a obtener una calidad perfecta se desea lograr una calidad necesaria o suficiente para el momento que el usuario así lo requiera. También se describirá un plan de capacitación para el personal de la empresa.
- **Según La WEB SITE**
- Mediante este enfoque analizaremos los siguientes puntos para nuestra métrica:
- Funcionalidad
- Confiabilidad
- Complejidad
- Usabilidad
- **4.1.1. FUNCIONALIDAD**
- La funcionalidad mide el grado en el que el software satisface las necesidades indicadas por la adecuación, exactitud, interoperabilidad y seguridad de acceso.
- Es por tanto que mediremos la Eficacia en la Eliminación de defectos (EED). Dentro del ámbito de desarrollo del sistema virtual, la EED se define de la siguiente manera:
- EED = E/(E+D)
- Donde:
- E = numero de errores antes de la entrega del software al usuario final.
- D = numero de errores encontrados después de la entrega al usuario final.
- Si el número de defectos presentados después de la entrega al usuario final es 0, entonces el valor de EED es 1.
- La correspondiente ponderación se representa en la siguiente tabla:

NIVEL DE CALIDAD	VALOR DE EED
Primer Nivel	Entre 1,15 y 1,10
Segundo Nivel	Entre 1,10 y 1,05
Tercer Nivel	Entre 1,05 y 1,00

- Reemplazando tenemos: EED = 32(32-3)
- EED = 0.91
- Si observamos los resultados de la EED, tenemos un 0.91, comparando con la ponderación de la tabla anterior tenemos que nuestro valor no se encuentra dentro de los niveles de calidad formulados, pero estamos cerca del tercer nivel de funcionalidad.
- **4.1.2. CONFIABILIDAD**
- La confiabilidad del software, es la probabilidad de que un sistema opere sin fallas bajo determinadas condiciones para un intervalo de tiempo dado.
- Los usuarios quieren que el software que vayan a utilizar tenga un funcionamiento correcto y los desarrolladores esperan que el número de defectos que existen puedan ser solucionados sin causar más problemas a medida que lo van probando, esto hace que la confiabilidad en el software y en los tiempos entre fallas se hace cada vez más grandes.
- Se examinara la confiabilidad en relación a la incertidumbre considerando una incertidumbre de tipo 1, que refleja la incertidumbre sobre la utilización del sistema.

Por lo tanto, en cualquier instante el tiempo hasta la próxima falla es incierto y se lo puede considerar como una variable aleatoria.

- Para estudiar esa variable aleatoria estudiaremos una distribución exponencial encausada en sus pruebas con cero fallas, que deriva de una función de cifras de falla, se supone que el número de fallas en el instante t es igual a:

- Fallas = a*e-b(t)Donde a,b son constantes.

- Este modelo se puede utilizar para determinar por cuantas horas se debe probar un sistema a fin de satisfacer una meta de confiabilidad.
- El modelo requiere 3 entradas:
    - Fallas = el numero proyectado promedio de fallas como objetivo
    - Fallas probadas = el número total de fallas observadas en las pruebas
    - Horas hasta la última falla = el número total de horas de ejecución de pruebas hasta la última falla
- El cálculo de horas necesarias de prueba para cero fallas es:

- Lnfallas0.5+fallas*(horas hasta la ultima falla)Ln[0.5+fallasfallas probadas+fallas]

- La revisión del sistema se efectuó bajo los siguientes aspectos:
- La revisión del sistema se efectuó bajo los siguientes criterios:
- Se probaron aproximadamente 11.956 líneas de código fuente del programa

- Han ocurrido 31 fallas
- El tiempo total de prueba fue 112 horas
- Las horas hasta la última falla es de 95 horas
- La siguiente tabla muestra las fallas sucedidas por cada 1000 líneas de código fuente:

• FALLAS	• Fallas por cada 1000 líneas de código
• 10	• 10/1000 = 0,01
• 8	• 8/1000 = 0,008
• 6	• 6/1000 = 0,006
• 5	• 5/1000 = 0,005
• 3	• 3/1000 = 0,003

- De acuerdo a la anterior tabla el promedio máximo de fallas por cada línea de código fuente es:
- (0.01+0.008+0.006+0.005+0.003)5=0.0064
- El número de horas de prueba necesarias para alcanzar la confiabilidad del sistema será:
- $\mathrm{Ln}0.069120.5+0.06912*182\mathrm{Ln}0.5+0.0691232+0.06912=95.174$
- Por lo tanto, el sistema puede usarse en un 95% del tiempo total en funcionamiento, sin problemas de fallas y con facilidad de recuperación.
- **4.1.3. USABILIDAD**
- Consiste en el grado o medida en que se facilita el uso del software es decir lo fácil de usar que resulta ser, y está definido por la: facilidad de comprensión, facilidad de aprendizaje y operatividad.

- Para medir la usabilidad se usaran las siguientes 3 metricas:
- **La completitud de la descripción**
- Dada por la formula: X=A/B
- Donde:
- A= número de funciones (casos de uso) o tipos de funciones descritas en la descripción del producto.
- B= número total de funciones (casos de uso)
- Reemplazando tenemos:
- X = 25/31
- X= 0.80
- Por lo tanto, el sistema presenta un 80% de entendimiento por parte de los usuarios finales respecto a la capacidad del producto.
- **Consistencia operacional**
- Dada por la formula: X = 1- A/B
- Donde:
- A = numero de instancias de operaciones con comportamiento inconsistente.
- B = número total de operaciones
- Reemplazando tenemos:
- X = 1 – 5/32
- X = 0.843
- Por lo tanto, el sistema presenta un 84% de no instancias de operaciones con comportamiento inconsistente.

- **Consistencia operacional en el uso**
- Dada por la formula: $X = 1 – A/B$
- Donde:
- A = número de funciones que el usuario encontró inaceptablemente inconsistente según sus expectativas.
- B = número de funciones usadas por el usuario durante el periodo de prueba.
- Reemplazando tenemos:
- $X = 1- 2/20$
- $X = 0.90$
- Por lo tanto, el usuario encuentra un 10% del sistema inaceptable en el periodo de prueba. De acuerdo a los resultados obtenidos anteriormente el usuario se encuentra satisfecho con la consistencia operacional del uso del sistema.
-
-
-
-
-
-
-
-
-
-
-

-
-
- **CAPITULO V**
- **COSTOS Y BENEFICIOS**
- El desarrollo completo del sistema implica un costo económico, el cual se describe a continuación.
- Costo de material a utilizar:

• ACTIVIDAD	• TIEMPO	• MATERIAL	• COSTO $us
• Investigacion preliminar	• 1 mes	• Fotocopias, impresiones y material de escritorio	• 15.00
• Analisis del sistema	• 2 meses		• 25.00
• Diseño del sistema	• 1 mes		• 150.00
• Desarrollo del sistema	• 2 meses		
• Prueba del Sistema	• 1 mes	• 2 computadoras Core2Duo	• 0.00
• Puesta en marcha del sistema	• 1 mes	• impresora A tinta o laser	• 90.00
•	•	•	• **280.00**

- Costo del personal Requerido:

• ACTIVIDAD	• PERSONAL	• Sueldo mes ($us)	• Tiempo contratado	• Sueldo Total
• Investigador Preliminar • Analisis del sistema	• 1 analista de sistemas	• 100	• 2 meses	• 200
• Diseño del sistema • Desarrollo del sistema	• 2 programadores	• 125	• 2 meses	• 500
• Prueba del Sistema • Puesta en marcha del sistema	• 1 administrador	• 100	• 1 mes	• 100
•	•	•	• **Total**	• 800

- Por tanto, sumando los valores totales de ambas tablas:
- Costo Total del desarrollo del sistema = **1080 $us**
- Nota: la empresa cuenta con los equipos necesarios para implementar el sistema por tanto no es necesario la compra de equipo alguno.
- **"Sistema de control de personal, almacenes y maquinarias" = 9.292 líneas de código**
- La **ecuación del esfuerzo** de **COCOMO** es:
- E = Esfuerzo = a KLDC b (persona x mes)
- KLCD es el número de líneas de código en miles
- La **ecuación del tiempo de desarrollo** es:
- T = Tiempo de duración del desarrollo = c Esfuerzo d (meses)
- Ahora necesitamos una tabla para **obtener los coeficientes (a, b, c, d)** que aparecen en las fórmulas citadas, estos coeficientes se obtienen de manera empírica y por lo tanto se basa en la experiencia de datos anteriores.

• Proyecto de software	• *a*	• *b*	• *c*	• *d*

• Orgánico	• 2,4	• 1,05	• 2,5	• 0,38
• Semiacoplado	• 3,0	• 1,12	• 2,5	• 0,35
• Empotrado	• 3,6	• 1,20	• 2,5	• 0,32

- **Tabla "Coeficientes COCOMO"**

- Por lo tanto el tamaño de nuestro "Proyecto prueba" como hemos visto es de 9,2 miles de líneas de código, si **aplicamos las fórmulas**:
- **Esfuerzo realizado** = 2,4 * 9.2 ^{1,05} = **25,77 personas / mes**
- **T** = 2,5 * 25,77 ^{0,38} = **4,86 mes**
- **Nº de personas para desarrollar el proyecto** = E/T= 21,9 / 8,1 » **5 personas**
- Por lo tanto y con estos resultados diríamos que el **"Proyecto Prueba"** debería terminarse en **aproximadamente 5 meses por un equipo de 5 personas**.
-
-
-
-
-
-
-
-
-
-
- **CAPITULO VI**

- **SEGURIDAD DEL SISTEMA**

- Md5

- Para registrar una variable de sesión en versiones anteriores de PHP, se usaba la función session_register, aunque esta función no es aconsejada pues está obsoleta, a partir de PHP 5.3.0.

- Para registrar una variable de sesión y establecer un valor usaremos $_SESSION["nombre_varaable"]. En el siguiente ejemplo, registramos y establecemos el valor para las variables de sesión nombre_cliente y nombre_usuario:

-

- $sql = "SELECT u.usuario, u.contrasena, u.idcliente, c.nombrecliente " .
- " FROM usuario u, cliente c ".
- " WHERE c.id=u.idcliente and usuario='" . $txtusuario . "'";
- $sqlResultado = mysql_query($sql);
- $row = mysql_fetch_array($sqlResultado);
- $contrasena = $row["contrasena"];
- $idcliente = $row["idcliente"];
-
- if ($contrasena == md5($txtcontrasena))
- {
- //establecermos las variables de sesión
- $_SESSION["nombre_usuario"] = $row["usuario"];
- $_SESSION["nombre_cliente"] = $row["cliente"];
- ...
- El ejemplo anterior nos sirve también como método para realizar el inicio de sesión (validación) de un usuario en nuestro sitio web, mostramos la consulta SQL que se ejecutará y la comprobación de si el usuario existe y si la contraseña introducida en el formulario (txtcontrasena) coincide con la guardada en la base de datos (contrasena una vez obtenido su hash md5).

- Nota: cuando el usuario se da de alta en nuestro sitio web, guardamos el hash (md5) de la contraseña en la base de datos, usando la función de PHP: md5. Por ello, en el ejemplo anterior, cuando mostramos al usuario el formulario para iniciar sesión y éste introduce su nick y su contraseña, para comparar su contraseña con la de la base de datos, utilizamos nuevamente la función de PHP md5, para comparar el hash md5 de la contraseña introducida por el usuario con el guardado en la base de datos. Este método es el idóneo, así, ante cualquier acceso indebido a la tabla de usuarios de nuestra base de datos, sólo se mostrará en el campo "contrasena" el valor del hash md5, nunca la contraseña del usuario y puesto que el hash md5 es unidireccional, de él no se puede obtener la contraseña (en teoría).

- Cuando el usuario ha iniciado sesión, siempre es recomendable añadir un enlace para que éste pueda cerrar la sesión en cualquier momento de forma segura. Así evitaremos accesos indebidos por otros usuarios que usen el mismo equipo que el que inició sesión. En teoría, siempre que se cierre el navegador web se cerrará la sesión automáticamente, pero es aconsejable añadir este enlace para que el usuario decida cuándo cerrar la sesión.
- Para añadir el enlace de "Cerrar sesión", podemos insertar el siguiente código PHP donde queramos que aparezca el enlace:
- <? session_start();
- if(!isset($_SESSION)){
- header("location:login.php");
- } else {
- echo "";
- echo "
- <h1>SectorWeb.net</h1>
- ";
- echo "Bienvenido al Area de usurios: ";
- echo $_SESSION["nombre"]." ".$_SESSION["apaterno"]." ".$_SESSION["amaterno"]." ";
- echo "
- Has entrado con el nick: ";
- echo $_SESSION["login"];
- echo "
- Para cerrar la sesión, pulsa: Aqui";

- echo "";
- }
- ?>
- Con if (!empty($_SESSION["nombre_usuario"])) comprobamos si el usuario ha iniciado sesión, si lo ha hecho la variable de sesión nombre_usuario tendrá un valor, por lo que se ejecutará el código del if.
- Cerrando sesión con seguridad:
- <HTML>
- <body background="n1.jpg">
- <?phpsession_start();
- // Borramos toda la sesion
- session_destroy();
- echo 'Ha terminado la session index ';
- ?><SCRIPT LANGUAGE="javascript">
- location.href = "index.php";
- </SCRIPT>
- </HTML>
- MD5 en acción:
- $password = $_POST["password"];
- $x=md5($_POST["password"]);
- $result = mysql_query('SELECT password, usuario,acceso FROM usuarios WHERE usuario=\".$usuario.'\");

- if($row = mysql_fetch_array($result)){
- if($row["password"] == $x{……
-
- **Protección en la Base de Datos**
- Para dar seguridad a la base de datos utlizaremos contraseñas con mas de 12 digitos combinando carateres y numeros.
- Desde el punto de vista técnico existen distintos puntos de revisión de una base de datos MySQL, fundamentalmente:
- Asignación de privilegios
- Fichero de configuración
- Cuentas de usuario
- Acceso remoto
- Una de las primeras acciones de revisión se debe centrar en la revisión de las cuentas por defecto, en este caso root. Para ello se puede comprobar si existe la cuenta y contiene contraseña en blanco:
- SELECT User, Host
- FROM user
- WHERE User='root';
- El acceso remoto (puerto 3306) a la base de datos debe ser monitorizado y controlado. Es muy común que el servidor de aplicaciones resida en la misma máquina que la base de datos, por lo que se podría limitar las conexiones a la base de datos desde fuera de localhost. Para ello se puede configurar esta opción en el fichero my.cnf añadiendo:
- MYSQLD_OPTIONS="–skip-networking"
- Adicionalmente existen algunos parámetros de configuración o arranque de la base de datos interesantes desde el punto de vista de la seguridad:
- skip-grant-tables: Arrancar el sistema con esta parámetro supone una grave brecha de seguridad, ya que supone que cualquier usuario tiene privilegios para hacer cualquier cosa sobre la base de datos. En algún caso puede ser útil para recuperar la contraseña de root.
- safe-show-database: Permite mostrar solamente aquellas bases de datos sobre las que un usuario tiene algún tipo de privilegio
- safe-user-create: permite determinar si un usuario puede crear nuevos usuarios siempre y cuando no tenga privilegios de INSERT sobre la tabla mysql.user
- Finalmente, y partiendo de la base de asignación del mínimo privilegio necesario, uno de los aspectos de seguridad a revisar son los privilegios asignados en la base de datos:
- Select * from user where
-

- Select_priv = 'Y' or Insert_priv = 'Y'
- or Update_priv = 'Y' or Delete_priv = 'Y'
- or Create_priv = 'Y' or Drop_priv = 'Y'
- or Reload_priv = 'Y' or Shutdown_priv = 'Y'
- or Process_priv = 'Y' or File_priv = 'Y'
- or Grant_priv = 'Y' or References_priv = 'Y'
- or Index_priv = 'Y' or Alter_priv = 'Y';
- Select * from host
- where Select_priv = 'Y' or Insert_priv = 'Y'
- or Create_priv = 'Y' or Drop_priv = 'Y'
- or Index_priv = 'Y' or Alter_priv = 'Y';
- or Grant_priv = 'Y' or References_priv = 'Y'
- or Update_priv = 'Y' or Delete_priv = 'Y'
- Select * from db
- where Select_priv = 'Y' or Insert_priv = 'Y'
- or Grant_priv = 'Y' or References_priv = 'Y'
- or Update_priv = 'Y' or Delete_priv = 'Y'
- or Create_priv = 'Y' or Drop_priv = 'Y'
- or Index_priv = 'Y' or Alter_priv = 'Y';
-
- **Protección del Sistema Operativo**

- El segundo punto de acceso a los datos almacenados en Microsoft SQL Server Analysis Services, tras el acceso al equipo físico, es el sistema operativo Microsoft Windows. Un sistema operativo protegido inadecuadamente puede poner en peligro la seguridad de una instancia de Analysis Services.

- En los siguientes temas se describe cómo proteger el sistema operativo Windows:
- Restringir el acceso de inicio de sesión interactivo
- Restringir el acceso a la red
- Deshabilitar los servicios innecesarios
- Especificar y restringir puertos
- Conceder derechos administrativos locales
- En estos temas se asume que Analysis Services está instalado en un servidor miembro de un dominio, en lugar de estar instalado en un controlador de dominio o en un equipo independiente. Por cuestiones de seguridad y rendimiento, no se recomienda instalar Analysis Services en un controlador de dominio. Para obtener más

información acerca de la forma de proteger el sistema operativo Windows, vea el sitio Web sobre seguridad de Microsoft.

-
-
-
-
-
-
-
-
-
-
-
- **CAPITULO VII**
- **CONCLUCIONES Y RECOMENDACIONES**
- **7.1. CONCLUSIONES**
- A la finalización de este proyecto titulado: Sistema Informático de Control de almacenes, maquinarias y de Personal para la constructora INGLOBOL S.A., se puede mostrar el logro de cada uno de los objetivos:
- *Según los Objetivos Específicos se concluye que:*
- Se diseñado una base de datos relacional para el sistema automatizado para el almacenado de datos importantes para la empresa, ya que el sistema registra y administra los datos correspondientes, presentando reportes completos que ayudan a culminar los proyectos en el tiempo establecido.
- Se ha realizado un entorno grafico amigable para el usuario con los respectivos controles de accesos para los usuarios

- Se automatizó el control de los recursos en obra, ya que el sistema registra y administra los datos de funcionamiento diario de la maquinara.
- Se automatizo el control de los almacenes para la entrada y salida de materiales de esta forma evitar la pérdida de materiales en obra.
- Se automatizo el registro del personal técnico y obreros para el seguimiento y control del personal que trabaja en el o los proyectos.
-
- *Según el objetivo general se concluye que:*
- Se ha diseñado e implementado el Sistema Integrado de Sistema Informático de Control de almacenes, maquinarias y de Personal para la constructora INGLOBOL S.A. satisfactoriamente, con los módulos correspondientes y por haber sido desarrollado conforme a los requerimientos del usuario.
- *Al problema se concluye que:*
- Un sistema informático automatizado para la empresa INGLOBOL S.A. en sus Proyectos Barrios de Verdad ha logrado que los controles de personal, materiales en almacenes, compras, maquinarias y agregados sean agiles, actualizadas y precisas.
-
- **7.2. RECOMENDACIONES**
- Con la implementación del sistema y analizando los alcances obtenidos se recomienda:
- • Implementar un sistema biométrico para un correcto y eficaz control de asistencia del personal.
- • Implementar un módulo para control diario por fechas donde se tomen en cuenta los atrasos y horas extras en forma gráfica.

- • Desarrollar un portal web de la empresa constructora, donde se tenga la información más relevante de la constructora, además de presentar información actualizada del avance de los proyectos.
- • Desarrollar un módulo que permita el registro e ingreso de los clientes para que puedan conocer la información aún más detallada del estado de avance de sus obras y de los recursos involucrados en el proyecto.
-
-
-
-
-
-
-
-
- **FUENTES DE INFORMACION**

- Booch, G., Rumbaugh, J., & Jacobson, I(2000), el Lenguaje Unificado de Modelado. (A. Otero, Ed.) Madrid, España: Addison Wesley.
- Castaño, A. d. (2001) Diseño de base de datos. Madrid, España.
- Pressman, R.S. (2003) Ingeniería de software un enfoque practico. (sexta edición).
- http://www.alegsa.com.ar/Dic/uml.php
- http://www.innovavirtual.org/campus/file.php/178/archivos_curso/CAP_11_2006_I_SI905/CAP_11_2006_I_SI905_VA4_M.pdf
- http://yaqui.mxl.uabc.mx/~molguin/as/RUP.htm
- http://es.wikipedia.org/wiki/Sistema_de_informaci%C3%B3n
- http://www.monografias.com/trabajos7/sisinf/sisinf.shtml

- http://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software
- http://www.monografias.com/trabajos5/inso/inso.shtml
- http://www.google.com.bo/url?sa=t&rct=j&q=%20construccion%20civil&source=web&cd=1&ved=0CEAQFjAA&url=http%3A%2F%2Fes.wikipedia.org%2Fwiki%2FConstrucci%25C3%25B3n_civil&ei=xehMT9LJMYautwesovEz&usg=AFQjCNFXYfhv4qCcCqBaUmfos-gl1onswg&cad=rja

- **ANEXOS**

Árbol de Problemas

Pérdida de recibos de agregados	Pérdida de materiales en almacenes	Baja Productividad en obra

Baja Capac

Pérdic registr horom

Bu falt registros actualizac

La empresa INGLOBOL S.A. carece de un Sistema Automatizado de información por tanto sus Proyectos Barrios de Verdad no tienen controles de personal, materiales en almacenes, compras, maquinarias y agregados de esta forma no se cuenta con información actualizada y precisa

e Objetivos

Mejor Productividad en los proyectos

Mejora en los controles de almacenes

Información útil y actualiza

Mejora en los controles de horometros

Desarrollar un Sistema Informático Integrado para la

:OL S

con

lizada

Realizar un entorno grafico amigable al usuario.

Diseñar una Base de datos relacional para el sistema automatizado