ESCUELA POLITÉCNICA DEL EJÉRCITO

FACULTAD DE INGENIERIA EN SISTEMAS E INFORMATICA

Metodología CMMI para la fuerza terrestre

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS EN INFORMÁTICA

POR: Lorena Burgos
Roberto Bautista

SANGOLQUI, Mayo del 2007

INDICE

CAPÍTULO 1……1
INTRODUCCIÓN.....1
1.1 Descripción del Problema.....1
1.2 Antecedentes.....1
1.3 Situación Actual.....3
1.4 Alcance.....4
1.5 Justificación.....5
1.6 Objetivos.....7

CAPÍTULO 2 8
MARCO TEÓRICO 8
2.1 Generalidades..... 8
2.2 Elementos de Calidad del Software 11
2.2.1 Métodos 11
2.2.2 Herramientas 11
2.2.3 Procedimientos 12
2.3 Desarrollo del Ciclo de Vida 12
2.3.1 Procesos del Ciclo de Vida 13
2.4 Modelo CMM..... 13
2.4.1 Concepto 13
2.4.2 Niveles del Modelo CMM 16
2.4.2.1 Inicial……...16
2.4.2.2 Repetible……17
2.4.2.2 Definido……...17
2.4.2.2 Gestionado……...17
2.4.2.2 Optimizado……...17
2.5 Modelo CMMI..... 18
2.5.1 Definición 18
2.5.2 Disciplina 18
2.5.3 Niveles del Modelo CMMI 19
2.5.3.1 Inicial.....19
2.5.3.2 Gestionado.....19
2.5.3.3 Definido……20
2.5.3.4 Cuantitativamente Gestionado.....20
2.5.3.5 Optimizado.....20
2.5.3.6 Requisitos en niveles de madurez "Áreas de Procesos claves".....21
2.5.4 Importancia 21

CAPÍTULO 3 23
ANÁLISIS DE PROCESOS DE LOS SISTEMAS DE LA FUERZA TERRESTRE ……23

3.1 Descripción y Análisis de los Procesos 23
3.1.1 Cadena de valor..... 23
3.1.2 Mapa de Procesos 24
3.1.2.1 Gobernantes…….....24
3.1.2.2 Fundamentales……..25
3.1.2.3 De Apoyo…….....26
3.1.3 Macroprocesos 27
3.1.3.1 Macroproceso de Educación……..27
3.1.3.2 Macroproceso de Gestión del Talento Humano.....28
3.1.3.3 Macroproceso de Gestión Logística……..29
3.1.3.4 Macroproceso Gestión Financiera……29
3.1.4 Análisis Procesos..... 30
3.2 Análisis de Debilidades y Fortalezas de los Sistemas Informáticos 31
3.2.1 Fortalezas 31
3.2.1.1 Análisis de Fortalezas..... 32
3.2.2 Debilidades 34
3.2.2.1 Análisis de Debilidades..... 34
3.3 Hardware de Procesamiento de Datos 35
3.3.1 Equipamiento computacional 35
3.3.2 Conectividad y redes..... 35
3.3.3 Cableado Estructurado 36
3.4 Software de Desarrollo en Funcionamiento 37
3.4.1 Programas Monousuario para Uso en los Repartos……...37
3.4.2 Programas para Explotar en los Repartos……...39
3.4.3 Sistemas en Desarrollo Pertenecientes al “SIFTE”……41
3.4.3.1 Estándares para Desarrollo……42
3.4.4 Productos de la Generación de Software Aplicativo……..45
3.4.5 Estándares para el Equipamiento de Procesamiento de Datos……...47
3.5 Definición de recursos para el desarrollo de software..... 48

CAPÍTULO 4 50
ESTABLECEMIENTO DEL NIVEL DE LA FUERZA TERRESTRE DENTRO DEL CMMI Y ANÁLISIS DE REQUERIMIENTOS 50
4.1 Determinación el nivel de la Empresa dentro de CMMI según Evaluación de Procesos 50
4.2 Análisis de las Áreas del Nivele 2 53
4.3 Componentes de un Área de Proceso 61

CAPÍTULO 5 64
DEFINICIÓN Y DISEÑO DE LA METODOLOGÍA CMMI PARA LA FUERZA TERRESTRE 64
Fases de la Metodología..... 64
5.1 Capacitar al Personal en CMMI 64
5.2 Definición de los principales procesos de la Organización.....65
5.3 Arquitectura de Procesos..... 66
5.4 Desarrollar políticas organizacionales..... 79
5.5 Definir especificaciones de productos de software……... 83
5.5.1 Estándares Clientes y Componentes……83

5.5.2 Estándares Elaboración de base de datos……..88
5.5.3 Metodología de Desarrollo de Software……..114
5.5.4 Plan de Calidad SQA……..115
5.5.5 Plan de Pruebas……..124
5.6 Planificar elaboración e implementación de procedimientos 143
5.6.1 Definición de Procedimientos……...145
5.7 Definir Métricas de Software……..152

CAPÍTULO 6
CONCLUSIONES Y RECOMENDACIONES 167
6.1 Conclusiones.....167
6.2 Recomendaciones 168

LISTA DE TABLAS

CAPÍTULO 3

Tabla 3.1 Listado de Procesos – Educación……....27

Tabla 3.2. Lista de Procesos – Talento Humano…….28

Tabla 3.3 Lista de Procesos – Logística……...29

Tabla 3.4. Lista de Procesos – Finanzas……..29

CAPÍTULO 5

Tabla 5.1 Pruebas Alfa…….. 134

Tabla 5. 2 Pruebas

Beta……...135

Tabla 5.3 Lista de Chequeo para la característica

Fiabilidad……136

Tabla 5.4 Evaluación de pruebas……137

Tabla 5.5 Planilla de Ponderación para la Característica Fiabilidad……138

Tabla 5 6 Escala de Evaluación……..140

Tabla 5.7 Análisis de resultado general……140

Tabla 5.8 Historial de pruebas……141

Tabla 5.9 Categoría de un proyecto en función de sus líneas de código……159

LISTA DE FIGURAS

CAPÍTULO 2

Figura 2.1 Ciclo de vida del software……...12

Figura 2.2 Proceso del Ciclo de Vida……...13

Figura 2.3 Descripción de los Niveles del CMM…….16

CAPÍTULO 3

Figura 3.1 Mapa de Procesos de la Fuerza Terrestre……..23

Figura 3.2 Mapa de Procesos de Segundo Nivel – Procesos Gobernantes…….24

Figura 3.3. Mapa de Procesos de Segundo Nivel – Procesos Fundamentales...25

Figura 3.4. Mapa de Procesos de Segundo Nivel – Procesos Fundamentales...25

Figura 3.5 Mapa de Procesos Segundo Nivel – Procesos de Apoyo……..26

CAPÍTULO 4

Figura 4.1 Componentes Áreas de Proceso……..62

CAPÍTULO 5

Figura 5.1 Sistemas de Información y Comunicación……..65

Figura 5.2 Arquitectura de Procesos……...66

Figura 5.3 Dirección de Sistemas de Información y Comunicaciones……...72

Figura 5.4 Organización Interna……73

Figura 5.5 Esctrutura RUP……...114

Figura 5.6 Equipo SQA……..119

AGRADECIMIENTO

Agradezco a mi familia en especial a mi madre, a mi padre, a Nancy quienes con su apoyo moral y económico hicieron posible el estudio de inicio a fin en esta prestigiosa institución.

También quiero agradecer a cada una de las personas que estuvieron cerca, con quienes sufrimos, reímos, pasamos días sin dormir, incluso sin comer; si a ellos que sin duda fueron el gran respaldo en los peores momentos pues si a todos los amigos gracias muchas gracias por su amistad, su confianza.

Sin duda agradezco de todo corazón a Dios quien me dio voluntad y fuerza para lograr las metas y los objetivos.

Roberto Bautista

AGRADECIMIENTOS

Agradezco a mi familia por todo el apoyo recibido y por todo el sacrifico que han hecho mis padres durante el tiempo de mi carrera.

A mis hermanos por todo el apoyo que siempre me brindaron.

A mi novio Christian Cabezas por la ayuda brindada y el cariño surante todo el desarrollo de este proyecto.

A mis amigos por todos los momentos compartidos de nuestra vida universitaria y por el gran apoyo ya que sin ellos esto no hubiese sido posible.

Lorena Burgos

DEDICATORIA

Dedico esta tesis de grado a mi hijo quien es mi inspiración, a mi madre quien es mi fuerza, a mi esposa mi compañera, también a toda mi familia por su gran predisposición y sincero apoyo.

Quiero también dedicar este trabajo a un gran amigo, Eduardo, quien fue un gran soporte durante gran parte de la carrera, lamento que hoy no esté presente físicamente aunque se que en mi corazón siempre tendrá un espacio porque sin duda se lo merece, descansa en paz amigo.

Roberto Bautista

DEDICATORIA

Con infinito amor dedico esta tesis a mis padres por todo el camino que hemos recorrido juntos y todos los sacrificios que han hecho por darme esta carrera.

A mis maestros por su infinita paciencia, y por su gran enseñanza hacia nosotros.

Lorena Burgos

CAPÍTULO 1

INTRODUCCIÓN

Descripción del Problema

La Dirección de Comunicaciones y Sistemas de la Fuerza Terrestre (DICOMSI), se halla realizando el Sistema Integrado de la Fuerza Terrestre (SIFTE) utilizando estándares y documentación para el desarrollo de sistemas; los mismos que se hallan desactualizados. Al no contar con una metodología para el desarrollo de software de calidad; es imprescindible el planteamiento de una Metodología bajo el modelo CMMI, para generar y darle un seguimiento a los sistemas actuales y futuros que se desarrollen.

Antecedentes

La Dirección de Comunicaciones y Sistemas de la Fuerza Terrestre, es responsable de la automatización de las diferentes áreas estratégicas de la Fuerza, pero en la actualidad no dispone de la cantidad suficiente de recurso humano especializado en tecnología informática de punta, ni cuenta con una metodología basada en un modelo para el desarrollo de software de calidad, para construir los sistemas del “SIFTE”.

Por este motivo la Fuerza Terrestre ha realizado el convenio “Socio Tecnológico” para la construcción e implantación de los sistemas: planificación estratégica institucional, finanzas, auditoria interna, logística, operaciones, comunicación social, educación, doctrina, comunicaciones y sistemas, y sanidad, componentes del sistema integrado de la fuerza terrestre antes mencionado “SIFTE”; este convenio se lo ha hecho con la ESPE la misma que realizó el Análisis y Diseño del “SIFTE” y la construcción e implantación del Sistema Integrado de Personal (SIPER) mediante contrato.

La Fuerza Terrestre tiene la necesidad de desarrollar una Metodología CMMI con la finalidad de ubicar a la organización en un nivel y aplicar dicha metodología.

En diciembre de 2000, el SEI publicó un nuevo modelo, el CMMI o "Modelo de Capacidad y Madurez – Integración ", con el objetivo de realizar algunas mejoras respecto al modelo CMM que paso a ser considerado como "obsoleto".

El CMMI incluye cuatro disciplinas, en función de la amplitud de los procesos que cubre:

- CMMI-SW : Software
- CMMI-SE/SW : + Ingeniería de sistemas
- CMMI-SE/SW/IPPD : + Desarrollo integrado de procesos y productos
- CMMI-SE/SW/IPPD/SS : + Gestión de proveedores

A su vez se presenta en dos posibles representaciones, "Por niveles" y "Continua". En el primer caso permite evaluar el nivel de madurez de una organización en todas las áreas de proceso incluidas en un nivel y los inferiores simultáneamente, mientras que el segundo permite evaluar el nivel en cada área independientemente.

Las características principales son:

- Pone un mayor énfasis en el uso continuo de métricas
- Insiste en la necesidad de la trazabilidad desde los requerimientos al producto final
- Desglosa y detalla las áreas de proceso relativas a la ingeniería

Cambia el nombre a los niveles 2 y 4 que pasan a llamarse "gestionado" y "gestionado cuantitativamente".

Situación Actual

La Fuerza Terrestre, consciente de la importancia de su misión de seguridad y desarrollo, ha emprendido el proceso de modernización para lograr la eficiencia administrativa y operativa , está consciente que el manejo correcto de la información facilitará el cumplimiento de la misión y objetivos; razón por lo cual se halla empeñado en desarrollar el proyecto SISTEMA INTEGRADO DE INFORMACIÓN, el mismo que a mediano plazo permitirá satisfacer los requerimientos de información y una toma de decisiones ágil, oportuna y con un alto grado de certeza. Además, prevee proporcionar un significativo apoyo al Comando Conjunto de las FFAA.

Alcance

Este proyecto permitirá realizar una Metodología CMMI para la Fuerza Terrestre (F.T), partiendo de los procesos de la organización aplicando el CMMI, de lo cual se podrá obtener mayor beneficio para la organización.

Los Niveles del CMMI que se tomarán en cuenta para el análisis son:

- Nivel Inicial: Estado inicial donde el desarrollo se basa en la responsabilidad de los individuos.

    - Los procedimientos son inexistentes o localizados a áreas concretas.
    - No existen plantillas definidas a nivel corporativo.

- Nivel Gestionado: Se normalizan las buenas prácticas en el desarrollo de proyectos (en base a la experiencia y al método).

  Contiene las siguientes áreas:

    - Administración de Requerimientos
    - Planificación de Proyecto
    - Monitoreo y Control de Proyectos
    - Administración de Acuerdos con Proveedoeres
    - Medición y Análisis
    - Aseguramiento de la Calidad de Procesos y Productos
    - Administración de la Configuración

La metodología contemplara la evaluación de la empresa para hallar el nivel de ubicación dentro del modelo CMMI y sus procesos, si la empresa se encontrase en el nivel inicial se definirán los procesos para alcanzar el nivel 2.

Justificación

Las importantes mejoras tanto en la productividad como en la calidad del software terminado, hacen que la metodología CMMI comience a interesar a las grandes organizaciones que buscan una inversión de bajo riesgo con resultados medibles.

El CMMI se ha convertido en el estándar mundial, que permite a las organizaciones medir e incorporar mayores niveles de eficiencia o madurez en sus procesos de desarrollo y mantenimiento de software a través de una serie de procedimientos lógicos suficientemente probados.

Debido al avance tecnológico a nivel mundial, las organizaciones de nuestro país se ven en la obligación de ponerse a nivel de la globalización, y un gran paso para este cambio es la implantación de un modelo que ayude y guíe el desarrollo de software de calidad que garantice la inversión que se realiza para mantener los sistemas de información, los cuales ayudan a toda organización a preservar su información.

La Fuerza Terrestre en una iniciativa de mejora de la calidad de sus procesos requiere una Metodología de calidad la cual será elaborada bajo las tendencias del modelo CMMI, ya que este modelo actúa con la tendencia actual en el mercado que es la de adquirir un nivel de madurez en los procesos que asegure la calidad de los productos obtenidos.

La implantación de este modelo, ofrecerá a la Fuerza Terrestre importantes beneficios como el ahorro en costos de producción, y la reducción de errores en el producto terminado.

El CMMI no solo representa un modelo para mejorar la calidad del proceso, si no que representa la mejor forma de alinear la tecnología con el negocio de la compañía, la Fuerza Terrestre puede integrar sus procesos de desarrollo de software actual con los de este modelo.

Es importante mencionar que la implantación de este modelo dentro de la Fuerza Terrestre, puede requerir de algún tiempo por lo que los resultado se verán a largo plazo.

1.6 Objetivos

Objetivo General

Definir y Diseñar la Metodología CMMI para la Fuerza Terrestre para alcanzar el siguiente nivel de Calidad.

Objetivos Específicos

- Recolectar la información necesaria para llevar a cabo la determinación del nivel en el que se encuentra la organización.
- Realizar una evaluación para determinar el nivel actual de la organización en base a la información recolectada para poder iniciar con la metodología utilizando el modelo CMMI.
- Diseñar la metodología CMMI para la Fuerza Terrestre de acuerdo a la evaluación realizada.

CAPÍTULO 2

MARCO TEÓRICO

2.1 Generalidades

El Modelo CMM (Modelo de Capacidad y Madurez), se considera como un método de definir y gestionar los procesos a realizar por una organización.

En sus inicios fue desarrollado para los procesos relativos al software por la Universidad Carnegie-Mellon para el SEI[1].

El SEI es un centro de investigación y desarrollo patrocinado por el Departamento de Defensa de los Estados Unidos de América y gestionado por la Universidad Carnegie-Mellon. "CMM" es una marca registrada del SEI. Su homólogo en Europa es la Fundación ESI (Instituto Europeo de Software), con sede en Bilbao (España).

A partir de noviembre de 1986 el SEI, a requerimiento del Gobierno Federal de los Estados Unidos de América, realizó una primera definición de un modelo de madurez de procesos en el desarrollo de software, que se publicó en septiembre de 1987. Este trabajo evolucionó al modelo CMM o SW-CMM (CMM para Software), cuya última versión (v1.1) se publicó en febrero de 1993.

Las organizaciones que utilizan este modelo para mejorar sus procesos disponen de una guía útil para orientar sus esfuerzos. Además, el SEI proporciona formación a evaluadores certificados, capacitados para evaluar y certificar el nivel CMM en el que se encuentra una organización. Esta certificación es requerida por el Departamento de Defensa de los Estados Unidos, pero también es utilizada por multitud de organizaciones de todo el mundo para valorar a sus subcontratistas de software.

[1] SEI(Software Engineering Institute) Crea los modelos de Calidad CMM y CMMI

A partir de 2001, en que se presentó el modelo CMMI, el SEI ha dejado de desarrollar el SW-CMM, debido a esto todas las organizaciones que deseen certificarse deberán adaptarse al nuevo modelo CMMI a partir del 2005.

La integración del Modelo de Capacidad de Madurez (CMMI) es un proyecto patrocinado por el Departamento de Defensa de los Estados Unidos. El SEI, el gobierno y la industria de este país posteriormente se unieron para desarrollar el framework del CMMI y solucionar el problema de utilizar múltiples modelos que limitaban la habilidad de las organizaciones para enfocar sus mejoras exitosamente. El CMMI es un modelo de procesos (no de mejora) que evalúa la madurez de una organización basándose en la capacidad de sus procesos y surge como la integración del CMM v.2.0 y de la ISO 15504 Draft Standar v.1.00. Este modelo de procesos tiene dos representaciones: continua y por etapas, siendo la diferencia entre éstas la evaluación por niveles de la capacidad de procesos o de la madurez de la organización, respectivamente. Las áreas de procesos en este modelo se agrupan en cuatro categorías:

- Gestión de Proyectos
- Soporte
- La Gestión de Procesos
- Ingeniería

Las áreas de proceso de la Ingeniería abarcan las actividades de desarrollo y mantenimiento compartidas a través de las distintas disciplinas (Ingeniería de Sistemas e Ingeniería de Software). Las seis áreas de procesos de la Ingeniería tienen interrelaciones inherentes que son el resultado del uso de un proceso de desarrollo de productos.

Las áreas de procesos de la Ingeniería según el CMMI son:

- Desarrollo de Requisitos
- Gestión de Requisitos
- Solución Técnica
- Integración del Producto
- Verificación
- Validación

El proceso de validación en el CMMI, efectúa validaciones incrementales de los productos contra las necesidades de los clientes. Este proceso puede realizarse en el ambiente operacional o en una simulación del mismo. La coordinación con el cliente en los requisitos de la validación es uno de los elementos esenciales en esta área de proceso. La validación confirma que el producto desempeñará el uso previsto, es decir que se desarrolle lo correcto. Si bien a primera vista la validación y la verificación aparentan ser muy similares en el CMMI, son dos

procesos diferentes.

2.2 Elementos de Calidad del Software

Los elementos de Calidad del software son:

- Métodos
- Herramientas
- Procedimientos

2.2.1 Métodos

Indican “como” construir el software

Abarcan un amplio espectro:

- Planificación y estimación de proyectos
- Análisis de los requisitos del sistema y del software
- Diseño de estructuras de datos
- Arquitectura de programas y procedimientos algorítmicos
- Codificación, prueba y mantenimiento

2.2.2 Herramientas

Suministran un soporte automático para los métodos

Existen herramientas para soportar cada uno de los métodos mencionados anteriormente:

- Un entorno con integración de diferentes herramientas es lo que se denomina un sistema CASE (Ingeniería del Software Asistida por Ordenador)

2.2.3 Procedimientos

Son el “pegamento” que junta los métodos y las herramientas.

Definen:

- La secuencia en la que se aplican los métodos.
- Las entregas (documentos, informes, etc.) que se requieren los controles que ayudan a asegurar la calidad y coordinar los cambios.
- Las directrices que ayudan a los gestores del software a evaluar el progreso.

2.3 Desarrollo del Ciclo de Vida

Un gran problema de los sistemas de información es definir un marco de referencia común que pueda ser empleado por todos, en el que se definan procesos, actividades y tareas.

Por lo que se crea el ciclo de vida de un Software como lo representa la figura.

Ciclo de Vida de un Software

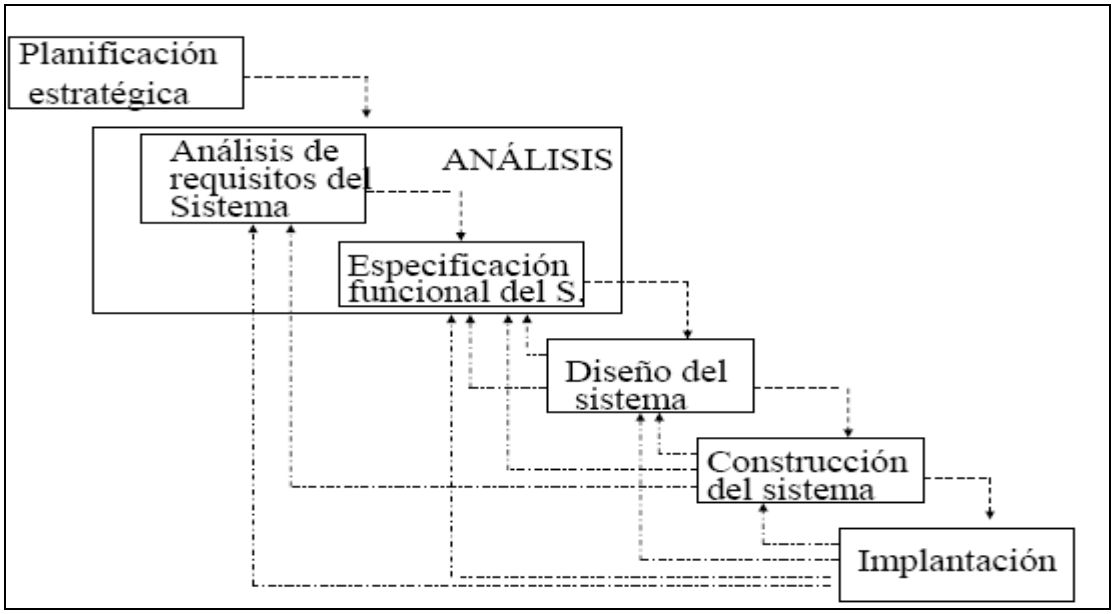


Figura 2.1 Ciclo de vida

2.3.1 Procesos del Ciclo de Vida

Las actividades del ciclo de vida se agrupan en 5 procesos principales, 8 procesos de soporte y 4 procesos generales.

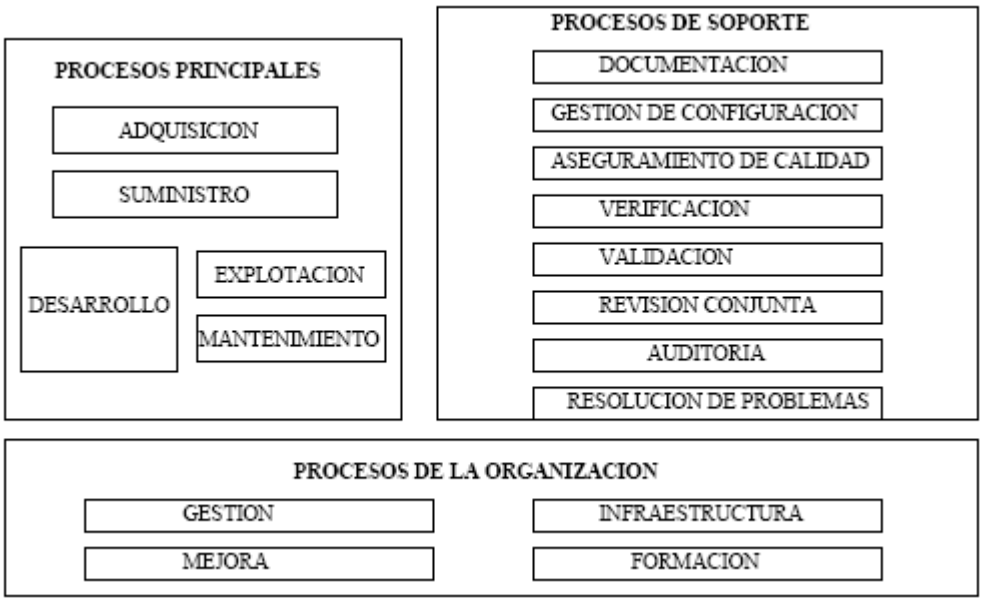


Figura 2.2 Procesos del Ciclo de Vida

2.4 Modelo CMM

2.4.1 Concepto

En 1984, el Departamento de Defensa de los Estados Unidos, al observar los grandes problemas que se presentaban en el desarrollo de software, decidió formar el Instituto de Ingeniería de Software (SEI) con el fin de establecer estándares de excelencia en la ingeniería de software. En 1987, el SEI creó el esquema de madurez para el proceso de software, desarrollado con la ayuda de otras agencias del gobierno de Estados Unidos. Este esquema recibió una buena acogida en la industria del software de los Estados Unidos y en el gobierno. Luego de algunos cambios, se acordó establecer el Modelo de Madurez de Capacidades, en su versión 1.0 en 1991 y luego en su versión 1.1 en 1992.

CMM (Modelo de Madurez de la Capacidad del Software) del SEI se define como un conjunto de áreas claves del proceso, que describen las funciones de ingeniería del software que deben llevarse a cabo para el desarrollo de una buena práctica, agrupadas en cinco niveles inclusivos.

Konrad[2] define al CMM como "un modelo de prácticas fundamentales que pueden ser implementadas por toda organización interesada en desarrollar y mejorar la calidad de sus productos y su productividad". Este modelo está basado en conceptos de calidad total y de mejoramiento continuo y ha sido ampliamente aceptado por la comunidad de ingeniería de software.

Rápidamente se ha convertido en el estándar de factor en los Estados Unidos, y también a nivel internacional, para evaluar la madurez que tiene las organizaciones para producir software. Este modelo se puede utilizar no solamente como un manual de prácticas recomendables, sino que además como referencia para llevar a cabo auditorias y evaluaciones internas en organizaciones que desarrollan y mantienen software.

El CMM es uno de los mejores modelos para el proceso de software. Algunas ventajas que presenta son las siguientes:

- Provee en detalle el contenido de cada nivel de madurez. Esto permite su interpretación de manera correcta y lo convierte en un modelo fácil de usar.
- Enfatiza el mejoramiento continuo de procesos en sus cinco niveles.
- Este modelo estimula dicho mejoramiento en la industria del software.
- Realiza mediciones a la madurez de capacidades del software, por lo que puede ser cuantificado. Esta característica provee un método para evaluar las capacidades del software.
- Toma el proceso de software bajo control. Esto incrementa la productividad de una organización de software y mejora la calidad del software.

Este modelo también presenta algunas desventajas:

- Enfatiza la gestión de proyectos en lugar del desarrollo del producto.
- No contempla las metodologías y lenguajes de programación.
- No contempla al personal.

[2] [Konrad 2003] Konrad, M; 2003, "Designing CMM - Integrated : The Practice of Simplicity", New Riders Publishing

- No especifica el dominio de la aplicación.
- No contempla a la reproducción, distribución e instalación del producto.
- No contempla al entorno de mantenimiento.
- Excluye el análisis y determinación de riesgos.
- No es apropiado para el software de pequeñas dimensiones o de corta vida.

2.4.2 Niveles del Modelo CMM

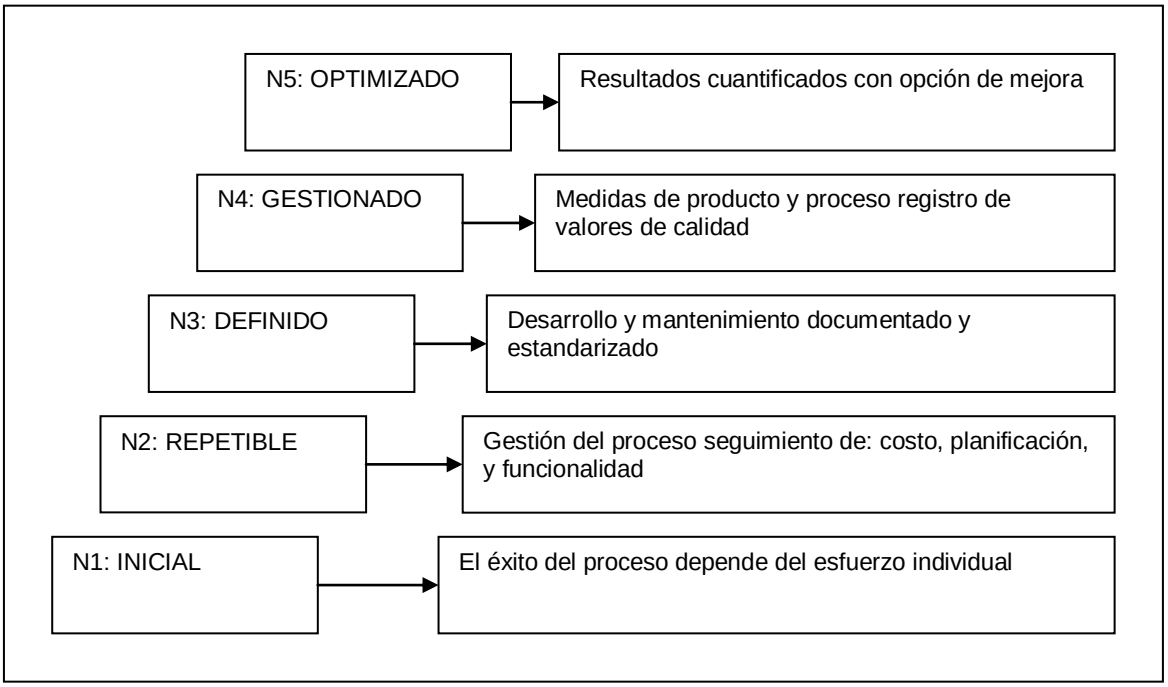


Figura 2.3 Descripción de los Niveles del CMM[3]

2.4.2.1 Inicial

Las organizaciones en este nivel no disponen de un ambiente estable para el desarrollo y mantenimiento de software. Aunque se utilicen técnicas correctas de ingeniería, los esfuerzos se ven minados por falta de planificación. El éxito de los proyectos se basa la mayoría de las veces en el esfuerzo personal, aunque a menudo se producen fracasos y casi siempre retrasos y sobre costos. El resultado de los proyectos es impredecible.

[3] Tomado http://es.wikipedia.org/wiki/Modelo_de_Capacidad_y_Madurez

2.4.2.2 Repetible

En este nivel las organizaciones disponen de unas prácticas institucionalizadas de gestión de proyectos, existen métricas básicas y un razonable seguimiento de la calidad. La relación con subcontratistas y clientes está gestionada sistemáticamente.

2.4.2.3 Definido

Además de una buena gestión de proyectos, a este nivel las organizaciones disponen de correctos procedimientos de coordinación entre grupos, formación del personal, técnicas de ingeniería más detallada y un nivel más avanzado de métricas en los procesos.

2.4.2.4 Gestionado

Se caracteriza por que las organizaciones disponen de un conjunto de métricas significativas de calidad y productividad, que se usan de modo sistemático para la toma de decisiones y la gestión de riesgos. El software resultante es de alta calidad.

2.4.2.5 Optimizado

La organización completa está volcada en la mejora continua de los procesos. Se hace uso intensivo de las métricas y se gestiona el proceso de innovación.

Las organizaciones que utilizan este modelo para mejorar sus procesos disponen de una guía útil para orientar sus esfuerzos. Además, el SEI proporciona formación a evaluadores certificados (Lead Assesors) capacitados para evaluar y certificar el nivel CMM en el que se encuentra una organización.

Se considera típico que una organización dedique unos 18 meses para progresar un nivel, aunque algunas consiguen mejorarlo. En cualquier caso requiere un amplio esfuerzo y un compromiso intenso de la dirección.

2.5 Modelo CMMI

2.5.1 Definición

Modelo de Madurez de Capacidades Integrado es un conjunto de prácticas recomendadas para el desarrollo de software, recopiladas a partir de experiencias en proyectos.

Está Organizado por niveles de madurez y áreas de proceso y provee guías para mejorar los procesos y la habilidad para administrar el desarrollo, adquisición y mantenimiento de productos y servicios.

El CMMI es el modelo más reconocido internacionalmente para la industria de software. Creado por el SEI – Sofware Engineering Institute de la Universidad Carnegie Mellon y a pedido del DoD – Departamento de Defensa de los EEUU.

2.5.2 Disciplina

CMMI se aplica a 4 disciplinas distintas y se puede elegir una de ellas para centrarse es aspectos específicos.

- Ingeniería de Sistemas - Cubre la construcción de un sistema con o sin software.
- Ingeniería de Software - Cubre la construcción de soluciones software
- Integración de productos y procesos de desarrollo - Cubre la relación a largo plazo con el cliente.
- Relación con proveedores - Cubre los procesos relacionados con la subcontratación de partes del sistema

2.5.3 Niveles del Modelo CMMI

CMMI propone 5 distintos modelos de madurez de las organizaciones:

2.5.3.1 Inicial

Estado inicial donde el desarrollo se basa en la responsabilidad de los individuos.

- Los procedimientos son inexistentes o localizados a áreas concretas.
- No existen plantillas definidas a nivel corporativo.

2.5.3.2 Gestionado

Se normalizan las buenas prácticas en el desarrollo de proyectos (en base a la experiencia y al método).

- En este nivel consolidado, las buenas prácticas se mantienen en los momentos de estrés.
- Están definidos los productos a realizar.
- Se definen hitos para la revisión de los productos.

2.5.3.3 Definido

La organización entera participa en el proceso eficiente de proyecto software.

- Se conoce de antemano los procesos de construcción de software.
- Existen métodos y plantillas bien definidas y documentados.
- Los procesos no solo afectan a los equipos de desarrollo sino a toda la organización relacionada.
- Los proyectos se pueden definir cualitativamente.

2.5.3.4 Cuantitativamente Gestionado

- Se puede seguir con indicadores numéricos (estadísticos) la evolución de los proyectos.
- Las estadísticas son almacenadas para aprovechar su aportación en siguientes proyectos.
- Los proyectos se pueden medir cuantitativamente.

2.5.3.5 Optimizado

- En base a criterios cuantitativos se pueden determinar las desviaciones más comunes y optimizar procesos.
- En los siguientes proyectos se produce una reducción de costos gracias a la anticipación de problemas y la continua revisión de procesos conflictivos.

2.5.3.6 Requisitos en niveles de madurez “Áreas de Procesos Claves”

A excepción del nivel 1, cada nivel de madurez está compuesto por varias áreas de procesos claves que indican las áreas en que la organización debe enfocarse para mejorar el proceso de software. Estas áreas identifican las pautas que deben cumplirse para llegar a cada nivel de madurez. Cada área de procesos identifica un conjunto de actividades relacionadas que, realizadas en conjunto, logran una serie de objetivos considerados importantes en la mejora de procesos.

Existen prácticas a realizar de modo:

- Particular (SP = práctica específica) en cada requerimiento.
- Globales (GP = Practica global) que van apareciendo repetitivamente en distintos puntos.

Cada práctica tiene a su vez sub-prácticas y practicas opcionales.

2.5.4 Importancia

El CMMI proporciona una guía para la mejora de procesos de la organización y su habilidad para gestionar el desarrollo, adquisición y el mantenimiento de productos.

- Ayuda a la organización a examinar la efectividad de sus procesos
- Establece prioridades para la mejora
- Ayuda a implementar esas mejoras
- Mayor manejo de conexiones explícitas y actividades de ingeniería para sus objetivos de negocio.
- Expandir el “alcance de” y la visibilidad dentro del ciclo de vida del producto y dentro de las actividades de ingeniería para asegurar que el producto o el servicio satisfacen las expectativas de los clientes.
- Incorporar lecciones aprendidas desde las mejores prácticas de áreas adicionales.
- Dirigir funciones críticas de la organización para sus productos y servicios.

CAPÍTULO 3

ANÁLISIS DE PROCESOS DE LOS SISTEMAS DE LA FUERZA TERRESTRE

3.1 Descripción y Análisis de los Procesos

La Fuerza Terrestre se encuentra realizando un proyecto de Implementación de Procesos. Dentro de este proyecto se ha definido la siguiente cadena de valor:

3.1.1 Cadena de valor

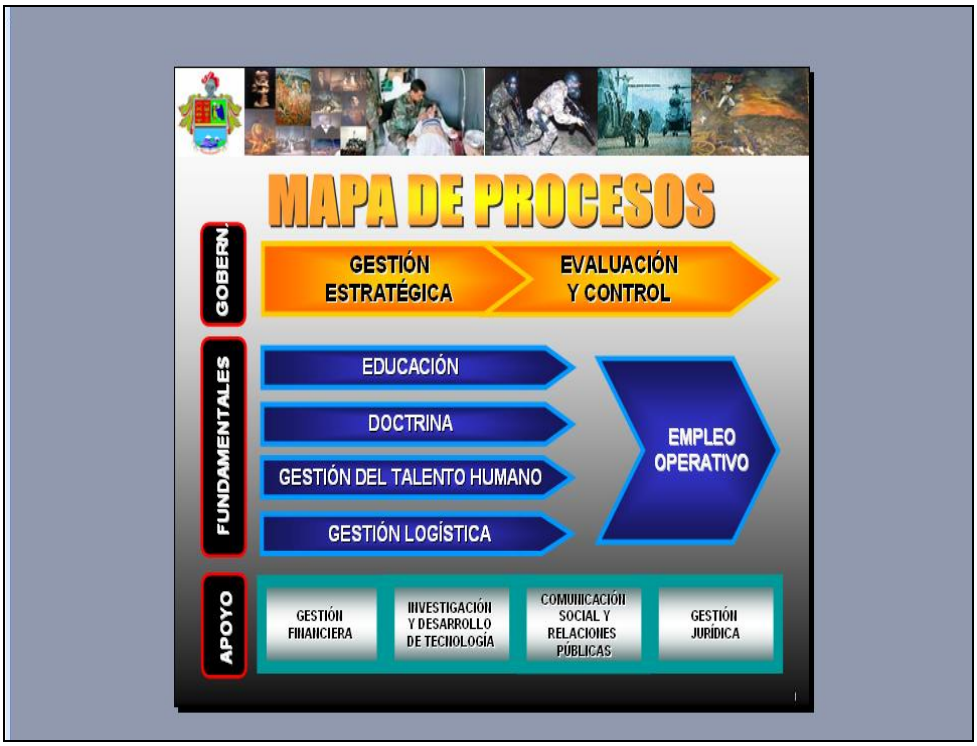


Figura 3.1. Mapa de Procesos de la Fuerza Terrestre

La determinación del Mapa de Procesos, aportará al cumplimiento de la misión de cada una de las direcciones y guiará a identificar los procesos con sus actividades que agregan valor a los productos que van a satisfacer los requerimientos de los clientes tanto internos como externos.

3.1.2 Mapa de Procesos

En este mapa de procesos encontramos tres tipos de procesos:

3.1.2.1 Gobernantes

Los procesos gobernantes son los que orientan, evalúan y hacen seguimiento a la gestión de la entidad.

- Establecimientito de políticas y estrategias
- Fijación de objetivos
- Fijación de lineamientos de comunicación y
- Asegura disponibilidad de recursos

Figura 3.2 Mapa de Proceso de Segundo Nivel – Procesos Gobernantes

3.1.2.2 Fundamentales

Los procesos fundamentales son los que contribuyen directamente al cumplimiento de los objetivos o la razón de ser de la organización.

Estos procesos son los que permiten el buen funcionamiento de las instituciones, es decir son fundamentales para cualquier institución.

Figura 3.3 Mapa de Procesos de Segundo Nivel – Procesos Fundamentales

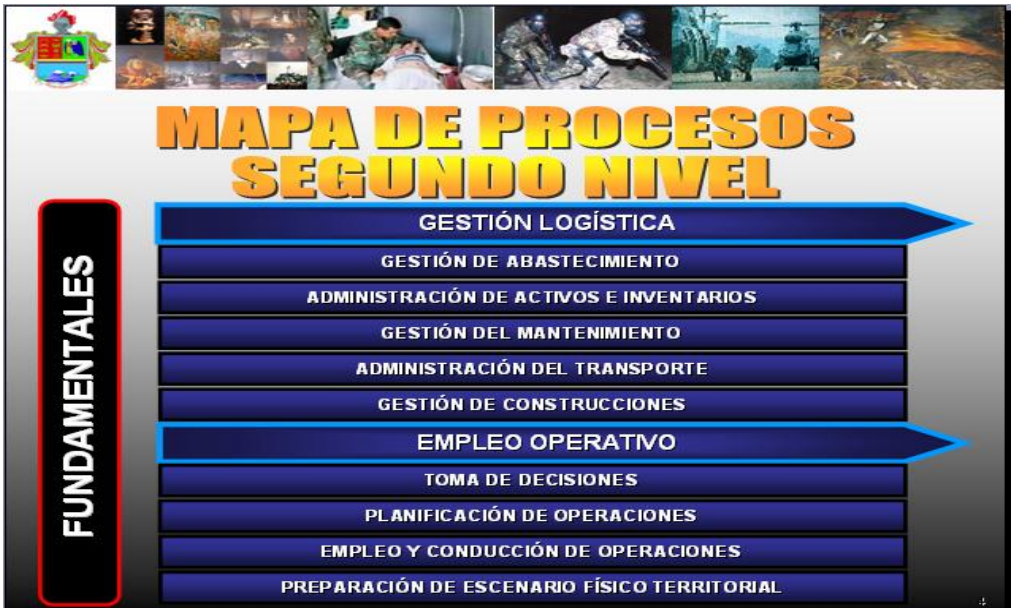


Figura 3.4 Mapa de Procesos de Segundo Nivel – Procesos Fundamentales

3.1.2.3 De Apoyo

Los procesos de apoyo son los que dan soporte para el buen funcionamiento y operación de los procesos gobernantes y fundamentales de la organización.

Figura 3.5 Mapa de Procesos Segundo Nivel – Procesos de Apoyo

3.1.3 Macroprocesos

Dentro de estos procesos definidos, se han desarrollado los siguientes:

- Macroproceso del Educación
- Macroproceso de Gestión del Talento Humano
- Macroproceso Gestión Financiera
- Macroproceso Gestión Logística

Cada macroproceso contiene:

3.1.3.1 Macroproceso de Educación

Dentro de este Macroproceso podemos encontrar los siguientes procesos y subprocesos

Tabla 3.1 Listado de Procesos - Educación

LISTADO DE PROCESOS
MACROPROCESO: EDUCACIÓN

MACROPROCESO	PROCESO	SUBPROCESO
EDUCACIÓN	GESTIÓN DE FORMACIÓN MILITAR	PLANIFICACIÓN GENERAL DE ENSEÑANZA (FORMACIÓN)
		INVESTIGACIÓN EDUCATIVA DE FORMACIÓN
		EVALUACIÓN Y SEGUIMIENTO DEL PLAN GENERAL DE ENSEÑANZA
	GESTIÓN DE PERFECCIONAMIENTO MILITAR	DISEÑO DEL PLAN GENERAL DE CURSOS
		PLANIFICACIÓN GENERAL DE ENSEÑANZA PARA PERFECCIONAMIENTO
		INVESTIGACIÓN EDUCATIVA DE PERFECCIONAMIENTO
		EVALUACIÓN Y SEGUIMIENTO DEL PGC Y PGE DE PERFECCIONAMIENTO
	GESTIÓN ESPECIALIZACIÓN MILITAR	DISEÑO DEL PLAN GENERAL DE CURSOS DE ESPECIALIZACIÓN
		PLANIFICACIÓN GENERAL DE ENSEÑANZA PARA ESPECIALIZACIÓN
		INVESTIGACIÓN EDUCATIVA DE ESPECIALIZACIÓN
		EVALUACIÓN Y SEGUIMIENTO DEL PGC Y PGE DE ESPECIALIZACIÓN
	GESTIÓN DE CONVENIOS EDUCATIVOS Y BECAS EN EL EXTERIOR	DETERMINACIÓN DE CONVENIOS EDUCATIVOS
		GESTIÓN DE CURSOS PARA PAÍSES AMIGOS
		PLANIFICACIÓN ANUAL DE BECAS
		EVALUACIÓN Y SEGUIMIENTOS DE INTERCAMBIOS EDUCATIVOS
	GESTIÓN DE CAPACITACIÓN	DESARROLLO DE PROYECTO DE CAPACITACIÓN
		CAPACITACIÓN A NIVEL DE POSTGRADOS
		EVALUACIÓN DE LA CAPACITACIÓN
		AUTORIZACIÓN DE CONTINUACIÓN DE ESTUDIOS POR MEDIOS PROPIOS

3.1.3.2 Macroproceso de Gestión del Talento Humano

Dentro de este Macroproceso podemos encontrar los siguientes procesos y subprocesos

Tabla 3.2. Lista de Procesos – Talento Humano

LISTADO DE PROCESOS

MACROPROCESO: GESTIÓN DEL TALENTO HUMANO

MACROPROCESOS	PROCESO	SUBPROCESO
GESTIÓN DEL TALENTO HUMANO	ASISTENCIA AL PERSONAL	ASISTENCIA SOCIAL
		ATENCIÓN CASOS DE SALUD
		ADMINISTRACIÓN DE SEGUROS
		ATENCIÓN PSICOLÓGICA
		REHABILITACIÓN SOCIAL
		ADMINISTRACIÓN DE EVENTOS
	SERVICIOS AL PERSONAL	ADMINISTRACIÓN DE SERVICIOS HABITACIONALES
		ADMINISTRACIÓN DE CENTROS INFANTILES
		TRAMITACIÓN DE SUBSIDIOS
		RETIRO, CESANTÍA Y DEVOLUCIÓN DE APORTES

3.1.3.3 Macroproceso de Gestión Logística

Tabla 3.3 Lista de Procesos - Logística

LISTADO DE PROCESOS

MACROPROCESO: GESTIÓN LOGÍSTICA

MACROPROCESOS	PROCESO	SUBPROCESO
GESTIÓN LOGÍSTICA	GESTIÓN DE ABASTECIMIENTO	OBTENCIÓN
		DISTRIBUCIÓN Y CONTROL
	ADMINISTRACIÓN DE ACTIVOS E INVENTARIOS	MANEJO DE INVENTARIO
		CONTROL Y ACTUALIZACIÓN DE INVENTARIOS
		BAJA DE INVENTARIOS
	GESTIÓN MANTENIMIENTO	PLANIFICACIÓN DEL MANTENIMIENTO
		EJECUCIÓN DEL MANTENIMIENTO
		EVALUACIÓN Y CONTROL DEL MANTENIMIENTO
		MANEJO DE ACCIDENTES
		ASIGNACIÓN DE TRANSPORTES
		ASIGNACIÓN DE PASAJES Y VIÁTICOS
	GESTIÓN DE CONSTRUCCIONES	PLANIFICACIÓN DE CONSTRUCCIONES
		EJECUCIÓN DE LA PLANIFICACIÓN
		EVALUACIÓN Y CONTROL DE CONSTRUCCIONES

3.1.3.4 Macroproceso Gestión Financiera

Tabla 3.4. Lista de Procesos - Finanzas

LISTADO DE PROCESOS

MACROPROCESO: GESTIÓN FINANCIERA

MACROPROCESO	PROCESO	SUBPROCESO
GESTIÓN FINANCIERA	GESTIÓN PRESUPUESTARIA	PLANIFICACION PRESUPUESTARIA
		EJECUCIÓN PRESUPUESTARIA
		EVALUACIÓN Y LIQUIDACIÓN PRESUPUESTARIA
	GESTIÓN CONTABLE	REVISIÓN DE CUENTAS
		CONTABILIZACIÓN DE HECHOS ECONOMICOS
		CONSOLIDACIÓN FINANCIERA
	GESTIÓN DE RECAUDACIONES Y PAGOS	GESTIÓN DE PAGOS
		GESTIÓN DE RECAUDACIONES
		GESTIÓN DE GARANTÍAS E INVERSIONES

3.1.4 Análisis Procesos

La implementación de los procesos proporciona la información objetiva de la relación interna entre las direcciones en los temas de desarrollo y estudio, para lo cual se utiliza:

- Macroprocesos
- Procesos
- Sub Procesos

Todos ellos definidos en su Mapa de Procesos, facilitando la administración de recursos, la optimización de tiempos y el mejoramiento de directrices para el control y evaluación del trabajo.

Los procesos están definidos en función de: Procesos Transfuncionales y Procesos Funcionales, convirtiéndose los mismos en una verdadera herramienta técnica y una guía práctica para el usuario lo que le permitirá estandarizar y normalizar todos los procesos ejecutados por los mismos, optimizando los recursos, costos y rendimiento de la gestión administrativa.

Los procesos anteriormente presentados, fueron realizados en un proyecto anterior, ya que el presente proyecto se refiere a un mejoramiento y actualización de los procesos.

Los cuatro procesos desarrollados actualmente son los procesos que se automatizarán ya que contienen actividades específicas para llegar a su objetivo, es decir se facilita su desarrollo dentro de un sistema. Por otro lado los procesos restantes no están aún desarrollados porque su ejecución es más compleja ya que las actividades que contienen no se pueden automatizar, es decir son procesos subjetivos.

Cabe mencionar que aumentar la madurez de los procesos consiste entonces en el grado en que están definidos, administrados, medidos y aplicados.

Aumentar la capacidad de los procesos representa los resultados esperados que pueden ser obtenidos a partir de aplicar el proceso.

Sin embargo una mejora de procesos involucra aspectos de ingeniería y administración, tanto a nivel de proyectos como de la organización.

3.2 Análisis de Debilidades y Fortalezas de los Sistemas Informáticos

3.2.1 Fortalezas

- Recurso Humano Capacitado en el área de Sistemas de Información
- Recurso Económico
- Recurso Tecnológico
- Organización Estable
- Procesos Definidos
- Apoyo Estatal
- Posee un Parque Empresarial
- Posee Asesoramiento Profesional

3.2.1.1 Análisis de Fortalezas

- Recurso Humano Capacitado en el área de Sistemas de Información

  La Comandancia cuenta con un personal capacitado en las diferentes áreas tecnológicas ya sea hardware o software. Por lo que se considera una fortaleza contar con un buen número de personal para desarrollar todas las actividades informáticas necesarias.

- Recurso Económico

  El Ejército Ecuatoriano cuenta con un presupuesto anual por parte del Estado, por lo cual es una institución solvente para fomentar cualquier tipo de cambio dentro de sus instancias, así como para realizar proyectos de mejora que le ayuden a ser una institución estable y organizada.

- Recurso Tecnológico

Disponen de un amplio parque informático, que consta de hardware y software actualizado, además cabe indicar que tienen desplegada una gran red de comunicaciones, y para mejorar todo esto están preparados para asumir la adquisición de nuevos equipos.

El desarrollo de sus aplicaciones se realiza en n capas, orientados a la Web y multiplataforma lo que permite integrar los sistemas que sean desarrollados.

- Organización Estable

  Se han definido los procesos organizacionales establecidos a través de normas, reglamentos, y un organigrama formado por todos sus niveles.

- Procesos definidos

  Se ha desarrollado todos los procesos y procedimientos institucionales, que ayudan al mejor funcionamiento y desempeño de las actividades dentro de la institución, involucrando dentro de cada proceso, sus respectivos formatos, registros, e instructivos que sirven para que el usuario pueda apoyarse en dicha información

- Apoyo Estatal

  La Fuerzas Armadas son una parte importante de todo estado, por lo que se encuentra estrechamente vinculado con este, y siempre tendrán gran apoyo en todos los campos.

- Posee un Parque Empresarial

  El parque empresarial con el que cuenta el Ejército Ecuatoriano, es diverso, encontrando empresas del tipo textil, de cuero, supermercados, artículos electrónicos, salud, educación, entre otras. Debido a que poseen todas estas empresas pueden acceder a mas recursos económicos y dar un servicio social.

- Posee Asesoramiento Profesional

  Existen instituciones y empresas que brindan su asesoramiento ya sea técnico o académico, un caso específico es la ESPE y las diferentes empresas proveedoras de la tecnología. Además de que se realizan gran cantidad de convenios.

Finalmente podemos concluir que la Fuerza Terrestre posee gran variedad de recursos lo que le permite acceder con la gran facilidad a los recursos tecnológicos existentes y además a contar con personas capacitadas para que puedan desarrollar sobre estos recursos, es decir los sistemas de información tiene gran potencialidad e importancia para esta institución.

3.2.2 Debilidades

- Existe rotación continua del personal directivo.
- Posee una Infraestructura muy grande
- Los sistema de información no están culminados ni integrados
- No posee una metodología de desarrollo de software de Calidad
- No poseen una seguridad informática buena

3.2.2.1 Análisis de Debilidades

- No existe Estabilidad Administrativa

  Este es un problema para todo de tipo de proyecto que se quiera implementar, ya que si no es implementado y ejecutado dentro del tiempo que se asigna a una persona como líder de la Comandancia que por lo general es un año, estos proyectos son sometidos a evaluaciones por parte del nuevo líder o comandante lo cual retrasa o cancela dichos proyectos.
- Posee una Infraestructura muy grande

  Al tener una infraestructura muy grande el control sobre la organización se hace mucho más complejo. El objetivo de la Fuerza Terrestre es integrar todos los sistemas de información de todas las dependencias del Ejército a nivel Nacional.

- Los sistema de información no están culminados ni integrados

  Los sistemas no están culminados es decir por ejemplo en el área de procesos solo se han desarrollado e implementado cuatro procesos, y en el área de software se desarrollan aún los sistemas de dichos procesos, por lo no se puede integrar ninguno de los sistemas, sin embargo aunque los sistemas de información no estén

culminados es muy importante que durante su desarrollo se orienten a una integración futura para que sean manejables.

3.3 Hardware de Procesamiento de Datos

La Fuerza Terrestre posee el siguiente equipamiento tecnológico donde se integra el hardware que procesa los datos de las aplicaciones de la institución.

3.3.1 Equipamiento computacional

3.3.1.1 HARDWARE:

Servidores de base datos: RISC, CISC (SUN, HP, DELL)

Servidores IAS: RISC, CISC (SUN, HP, DELL)

Servidores Web: RISC, CISC (SUN, HP, DELL)

Estaciones de trabajo: PC's actualizados (clones con garantía).

3.3.2 Conectividad y redes

3.3.2.1 CONECTIVIDAD:

Servidores de comunicaciones: RISC, CISC (SUN, HP, DELL)

Switch central (primario): Switch router capa 4.

Switch de piso (secundario): Switch capa 3.

Ruteador: Cisco. Protocolo WAN: IP, PPP, Frame Relay y HDLC.

Protocolos Ruteo: RIP versión 1/2, IGRP, OSPF.

Enlace E1: MODE, Troncalizado.

Enlace última milla: Banda ancha inalámbrico con seguridad

Enlace dial up: MODE, ANDINATEL, PACIFITEL

3.3.3 Cableado Estructurado

Cableado vertical (backbone):

Fibra óptica: Monomodo, multimodo

Cableado Horizontal:

Multipar: UTP

Normas de cableado para Voz, Datos, Video y Control:

EIATIA 568: Normas para cableado de telecomunicaciones.

EIATIA 568A: Características del cableado.

EIATIA 569: Normas para enrutar el cable y espacios de telecomunicaciones.

EIATIA 606: Normas para administrar el cableado.

EIATIA TSB-67: Normas para certificar el cableado.

3.4 Software de Desarrollo en Funcionamiento

3.4.1 Programas Monousuario para Uso en los Repartos

a.- Roles de Pago (Militares y Fondos Propios).

Registro de los ingresos y descuentos del personal que se encuentra en la unidad a partir del rol emitido por el Comando General.

b.- Activos Fijos (SAF).

Registro y actualización informática de todos los bienes muebles que maneja la Fuerza.

c.- Contabilidad – Presupuesto "SIFE":

Registro, proceso de los presupuestos, gastos e ingresos de las partidas presupuestarias asignadas a la unidad.

d.- Personal y Conscriptos:

Registro y actualización informática del personal de la unidad, sanciones, faltas, permisos, vacaciones, etc.

e.- Inspecciones:

Planificación, ejecución y evaluación de las Inspecciones operativas realizadas a las unidades militares de la Fuerza.

f.- Servicios Técnicos:

Ingreso y actualización de órdenes de trabajo para el mantenimiento de los equipos de computación.

g.- Control de Recursos de Comunicaciones y Sistemas "SICOMSI":

Ingreso y actualización de los datos de los recursos de comunicaciones e informáticos que dispone un Reparto. En el área informática: computadores, dispositivos, sistemas, paquetes, etc.

h.- Inventarios de Logística "SILOG-1":

Ingreso y actualización de los recursos logísticos que dispone cada Reparto; tales como: fusiles, morteros, munición, mochilas, etc.

i.- Control de Documentos (SISADE):

Ingreso, actualización y seguimiento de los documentos que ingresan a una Unidad.

j.- Pruebas Físicas:

Ingreso y actualización de los datos de las pruebas físicas de Oficiales y Voluntarios de un Reparto.

3.4.2 Programas para Explotar en los Repartos

a.- <u>Ambiente Cliente Servidor</u>:

1) Inventarios (Power Designer, Power Builder, Informix).-

Procesamiento de datos de Ingreso, actualización y control de los Abastecimientos que dispone los diferentes escalones logísticos.

2) Sistema de Personal "SIPER" (Power Designer, Power Builder, Oracle).-

Procesamiento de datos de la gestión administrativa de Previsión e Ingreso; Estímulos y Remuneraciones; Asignaciones, Pases y Rectificaciones; Gerencial; y, Retiro y Reservas del Personal de la Fuerza Terrestre.

3) Sistema de Bienestar "SIBIE" (Power Designer, Power Builder, Oracle).-

Procesamiento de datos de la gestión administrativa de Asistencia Social, Asistencia Legal, Asistencia Sicológica, Subsidios y Gerencial de Bienestar de la Fuerza Terrestre.

b.- <u>Ambiente Distribuido</u>:

1) Sistema de Educación "SIEDU" (Power Designer, Power Builder, Oracle).-

Procesamiento de datos de la gestión administrativa de Planificación, Investigación, Evaluación y Gerencial de Educación de la Fuerza Terrestre.

2) Sistema de Logística "SILOG" (Power Designer, Power Builder, Oracle).-

Procesamiento de datos de la gestión administrativa de ABASTECIMIENTO (Determinación de Necesidades, Obtención, Distribución y Control de inventarios); y, MANTENIMIENTO (Planificación y Ejecución) de los artículos logísticos de la Fuerza Terrestre.

c.- <u>Ambiente tres capas (Browser)</u>:

Sistema de Finanzas "SIFIN", Educación "SIEDU" y Logística "SILOG" (Power Designer, Power Builder, Oracle).-

Procesamiento de datos de la gestión administrativa de PRESUPUESTO (Formulación Presupuestaria, Evaluación y Liquidación Presupuestaria, y Consolidación Presupuestaria); CONTABILIDAD; TESORERIA; y, ACTIVOS FIJOS de la Fuerza Terrestre.

3.4.3 Sistemas en Desarrollo Pertenecientes al "SIFTE"

Hasta el 2009 se automatizará los siguientes sistemas: Planificación Estratégica Institucional, Auditoria Interna, Inteligencia, Operaciones, Comunicaciones y Sistemas, Comunicación Social y Sanidad.

3.4.3.1 Estándares para Desarrollo

a.- Concepto de Sistemas Distribuidos

Colección de elementos de cómputo autónomo que se encuentran físicamente separados y no comparten una memoria común, se comunican entre sí a través del intercambio de mensajes utilizando un medio de comunicación.

Un sistema distribuido es aquél que está compuesto por varios ordenadores autónomos conectados mediante una red de comunicaciones y equipados

con programas que les permitan coordinar sus actividades y compartir recursos

Los sistemas autónomos pueden tener características no homogéneas.

b.- Características de los Sistemas Distribuidos.

a.- Cada elemento de cómputo tiene su propia memoria y su propio Sistema Operativo.

b.- Control de recursos locales y remotos.

c.- Sistemas Abiertos (Facilidades de cambio y crecimiento).

d.- Plataforma no standard (Unix, NT, Intel, RISC, Etc.).

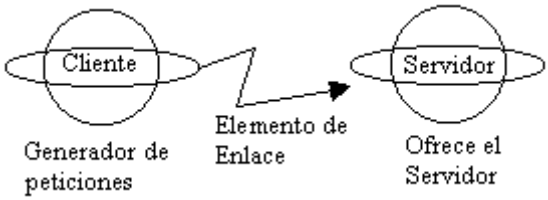
e.- Medios de comunicación (Redes, Protocolos, Dispositivos, Etc.).

f.- Capacidad de Procesamiento en paralelo.

g.- Dispersión y parcialidad.

c.- Arquitectura Cliente/Servidor

Una arquitectura es un conjunto de reglas, definiciones, términos y modelos que se emplean para producir un producto. La arquitectura Cliente/Servidor agrupa conjuntos de elementos que efectúan procesos distribuidos y computo cooperativo.



Cliente.-

Conjunto de Software y Hardware que invoca los servicios de uno o varios servidores.

a.- Características:

- El Cliente oculta al Servidor y la Red.
- Detecta e intercepta peticiones de otras aplicaciones y puede redireccionarlas.
- Dedicado a la cesión del usuario (Inicia...Termina).
- El método más común por el que se solicitan los servicios es a través de RPC (Remote Procedure Calls).

b.- Funciones Comunes del Cliente:

- Mantener y procesar todo el dialogo con el usuario.
- Manejo de pantallas.
- Menús e interpretación de comandos.
- Entrada de datos y validación.
- Procesamiento de ayudas.
- Recuperación de errores.

Servidor.-

Conjunto de Hardware y Software que responde a los requerimientos de un cliente.

a.- Tipos Comunes de Servidores:

- Servidor de Archivos (Unix, Windows).
- Servidor de Bases de Datos (SQL, ORACLE, INFORMIX).
- Servidor de Comunicaciones
- Servidor de Impresión.

- Servidor de Terminal.
- Servidor de Aplicaciones (Unix, Windows).

b.- Funciones Comunes del Servidor:

- Acceso, almacenamiento y organización de datos.
- Actualización de datos almacenados.
- Administración de recursos compartidos.
- Ejecución de toda la lógica para procesar una transacción.
- Procesamiento común de elementos del servidor.

3.4.4 Productos de la Generación de Software Aplicativo

Generar los productos detallados a continuación; los mismos que el grupo de proyecto deberán entregar en original y copia, impreso y digital, una copia para la Unidad y otra para la DICOMSI, para su archivo.

a.- De la mejora de los procesos del sistema:

1) Manual de Procesos actualizado.
2) Memoria técnica actualizada

b.- De la Construcción e Implantación del sistema:

1) Diseño de la Base de Datos.
2) Programas fuente.
3) Instaladores.
4) Ejecutables. Estos productos serán entregados, instalados e implementados en el Canal Piloto destinado para este fin.
5) Documentación actualizada:
   (a) Resultados del análisis de requerimientos.

(b) Resultados del análisis de cada subsistema.

(c) Documentos de estándares de Análisis y Diseño.

(d) Resultados del Diseño

(e) Modelo conceptual y físico

(f) Modelo de Objetos

(g) Diccionario informática debidamente documentado.

(h) Modelo dinámico (escenario) de los componentes del sistema orientado a objetos

(i) Diagrama de casos de usos

(j) Diagrama de clases

(k) Diagrama de secuencia

(l) Diagrama de estados

(m) Diagrama de actividades

(n) Diagrama de componentes

(o) Diagrama de configuración

(p) Prototipo básico y definitivo del Sistema debidamente documentado

(q) Especificaciones del Diseño detallado para la Construcción del Sistema

(r) Plan para la construcción del Sistema

(s) Documento de estándares de Construcción

(t) Manual Técnico del Sistema

(u) Manual del Usuario

(v) Manual de Procedimientos (Documentación Técnica)

(x) Plan para la instalación, puesta en marcha e implantación del Sistema

(y) Plan de Contingencias

6) Carpeta de comunicaciones tramitadas

7) Actas de las sesiones de trabajo

8) Informes de avances del Proyecto

9) Novedades y problemas, etc.

10) Otros documentos de importancia para la Unidad que hayan sido generados durante la construcción del Sistema

3.4.5 Estándares para el Equipamiento de Procesamiento de Datos

SOFTWARE DE BASE (OPERATIVO, APLICATIVO):

Servidores:

Sistema operativo:	UNIX SOLARIS, LINUX, Win2003
Servidor IAS:	De acuerdo al Front End utilizado
Comunicaciones:	TCP/IP
Base de datos:	Oracle
Gestión de Documentos (WorkFlow):	Lotus, Oracle
Sistemas de Información Geográfica:	MAP-INFO, GIS
Interfaces de usuario:	Gráfico, texto

Estación Cliente o Monousuario:

Sistema operativo:	Windows
Comunicaciones:	TCP/IP
Base de datos:	Oracle
Case de Modelamiento:	Power Designer, UML
Herramienta de desarrollo:	Power Builder, Java
Panificador de actividades:	MS-Project
Graficador de diagramas:	Visio
Interfaces de usuario:	Windows
Procesador de palabras:	MS-Word
Hoja electrónica:	MS-Excel
Presentador:	MS-Power Point

Adicionalmente, se incluye lo que se disponga en hardware y software con la capacidad de conexión a nuestras aplicaciones.

3.5 Definición de recursos para el desarrollo de software

Las plataformas utilizadas para desarrollar el software son:

- BackEnd Oracle
- case Power Designer
- FrontEnd Power Builder.

Documentación:

- Seguridades y Pistas de Auditoria
- Manual de Procesos actualizado.
- Memoria técnica actualizada
- Diseño de la Base de Datos.
- Documentos de estándares de Análisis y Diseño.
- Documento de estándares de Construcción
- Plan de Contingencias
- Manual del Usuario
- Manual Técnico del Sistema
- Otros documentos de importancia para la Fuerza Terrestre

Recursos Tecnológicos

- Instaladores.

Recurso Humano

- Los desarrolladores de los sistemas de la fuerza terrestre, que son contratados por la ESPE

Recurso Económico

- Licencias
- Soporte Técnico
- Instaladores

Recurso de Tiempo

- Tiempo invertido en desarrollar los sistemas, actualmente no se ha finalizado este trabajo.

CAPÍTULO 4

ESTABLECEMIENTO DEL NIVEL DE LA FUERZA TERRESTRE DENTRO DEL CMMI Y ANÁLISIS DE REQUERIMIENTOS

4.1 Determinación del nivel de la Empresa dentro de CMMI según Evaluación de Procesos

Los cuatro procesos que la Fuerza Terrestre ha desarrollado y que están siendo actualizados, son lo que han llevado a determinar que la organización tiene las siguientes características:

- Procesos habitualmente inexistentes y localizados en áreas específicas.
- La organización no posee un ambiente estable.
- El éxito depende del compromiso y competencia del personal con la institución y no del uso de procesos probados.
- No existen plantillas definidas a nivel corporativo.
- Frecuentemente no se calculan correctamente el presupuesto de sus proyectos.
- Los procesos no se realizan o no se consiguen los objetivos.
- Los procesos no son controlados.
- No poseen una metodología definida para el desarrollo de software de calidad.

- No se han establecido políticas para administrar proyectos de software y procedimientos para implantar estas políticas.
- La planificación y administración de nuevos proyectos, está basada en los resultados obtenidos con los proyectos actuales.

Un objetivo a cumplir por la Fuerza Terrestre para alcanzar el Nivel 2 es institucionalizar un proceso de administración efectivo para proyectos de software, los cuales permitan a la organización repetir prácticas exitosas desarrolladas en proyectos anteriores, aunque los procesos específicos implementados por los proyectos puedan diferir. Un proceso efectivo puede ser caracterizado como practicado, documentado, cumplido, capacitado, medido, y capaz de mejorar.

Determinación de Niveles

La empresa se encuentra en el nivel Inicial, ya que no existe un ambiente estable en el cual se pueda desarrollar o mantener software. En este nivel es normal no alcanzar las metas definidas ni en tiempo, ni costo, ni recursos planeados. Y si esto llegaría ocurrir, seria gracias a personas excepcionales dentro del equipo de trabajo.

Se trabaja con procedimientos que no están normalizados, es decir, procedimientos tanto del propio desarrollo de software como de su planificación y control, que no están establecidos explícitamente antes de su uso.

Por otro lado las técnicas y/o herramientas que se emplean para el desarrollo del software carecen de una integración entre las mismas y únicamente son empleadas en algunas fases del ciclo de vida del software.

La característica de las empresas que se encuentran en este nivel son: no hay un control de la gestión de proyectos software efectivo, ya que aunque la empresa disponga de procedimientos y técnicas formales, tanto de gestión como del proceso, y de herramientas, estas no se utilizan de manera estándar en todos los proyectos

Por lo que la Metodología a implementarse en esta tesis arrancaría desde el segundo nivel ya que se debe mejorar en:

- Gestión del Requerimientos: Controlar el Alcance
- Planeación de Proyectos: Estimar alcance, tiempo y costo
- Monitoreo y Control de Proyectos: Controlar cumplimiento de alcance, tiempo y costo
- Gestión de Proveedores: Seleccionar proveedores y gestionar el cumplimiento de los acuerdos que se establezcan
- Medición y Análisis: Mediciones del proyecto y de lo procesos
- Gestión de Configuración: Control de Versiones y Cambios
- Aseguramiento de la Calidad del Proceso y del Producto: Asegurar cumplimiento de Procesos y de estándares de entregables

Entonces las metas a perseguir en el nivel dos son:

- Definir las Políticas de Seguridad de la Información de las organización basadas en estándares internacionales
- Formar un grupo interdisciplinario, con el cual se busca trabajar temas de seguridad informática que sean de interés para la organización.
- Plantear una metodología CMMI para alcanzar la calidad en los procesos.
- Plantear diferentes fases para estandarizar cada proceso de ingeniería.
- Formalizar la planificación de proyectos dentro de la organización
- Plantear la capacitación del personal en el tema de CMMI

- Obtener el compromiso del personal en desarrollar sus procesos de acuerdo a los estándares planteados.
- Medir cada uno de los procesos basándose en métricas de calidad

4.2 Análisis de las Áreas del Nivel 2

PA1: Gestión de requisitos

El propósito de ésta área es manejar los requisitos de los productos del proyecto y los componentes e identificar incongruencias entre esos requisitos y los planes del proyecto y productos de trabajo.

Los procesos administrativos manejan todos los requisitos recibidos o generados por el proyecto, incluyendo tanto requisitos especializados y poco especializados, así como también estos requisitos incorporados en el proyecto por la organización.

Metas especificas del área:

SG1: Gestionar Requisitos

Gestionar los requerimientos técnicos y no técnicos pactados en un contrato, estándar o especificación formalmente documentado.

SP1.1: Obtener un Entendimiento de los Requisitos.

SP1.2: Obtener un compromiso con los requisitos.

SP1.3: Gestionar los cambios de requisitos.

SP1.4: Identificar inconsistencias entre el trabajo del proyecto y los requisitos.

PA2: Planificación del Proyecto.

El propósito de esta área es establecer y mantener planes que definen actividades de proyecto.

La planificación comienza con requerimientos que definen el producto y el proyecto. La planificación incluye estimar los atributos de los productos de trabajo y las tareas, determinar los recursos necesitados, negociar compromisos, producir un programa entre estos, e identificar y analizar riesgos del proyecto.

El área de proceso de Planificación de Proyectos implica lo siguiente:

- Desarrollar el plan de proyecto
- Interactuar con intereses del proyecto apropiadamente
- Obtener el compromiso al plan
- Mantenimiento del plan

Metas especificas del área:

SG1: Establecer Estimaciones.

Estimar razonablemente el uso de recursos y tiempo para la realización de un proyecto, debe actualizarse conforme se avance el proyecto y deberá tomar en cuenta escenarios a favor como en contra.

SP1.1: Estimar el alcance del proyecto.

SP1.2: Establecer estimaciones del producto de trabajo y los atributos de tarea.

SP1.3: Definir el ciclo de vida del proyecto.

SG2: Desarrollar un plan de proyecto.

SP2.1: Establecer el presupuesto y el calendario.

SP2.2: Identificar y analizar los riesgos del proyecto.

SP2.3: Planear la Gestión de la información.

SP2.4: Planear los recursos del proyecto.

SP2.7: Establecer el plan del proyecto.

SG3 Obtener responsabilidad hacia el plan

SP 3.1 Revisión de Planes que afectan al proyecto

SP 3.3 Obtener un Plan de Responsabilidades.

PA3 Seguimiento y Control del Proyecto.

El propósito es proveer un seguimiento del progreso del proyecto a fin de tomar las medidas correctivas apropiadas cuando el desempeño del proyecto se desvía significativamente del plan.

El plan documentado de un proyecto es la base para actividades monitoras, comunicar el estado, y tomar acciones correctivas. El progreso se determina comparando productos de trabajo actuales, atributos de tareas, esfuerzo, costo, y el programa para el plan en las actividades prescriptas.

Posibilita la toma de acciones correctiva oportunas cuando el desempeño se desvía significativamente del plan. Una desviación es significativa si, cuando no se resuelve, repercute en que el proyecto no alcance sus objetivos.

Metas especificas del área:

SG1 Seguimiento del Proyecto respecto el Plan.

Basados en el Plan del Proyecto debemos monitorear las acciones a llevarse a cabo, así como aplicar medidas correctivas de ser necesario, tomar en cuenta que estas impactarán en nuestro Plan de Proyecto.

SP 1.1 Parámetros de Planificación del seguimiento del proyecto.

SP 1.2 Seguimiento de las responsabilidades.

SP 1.3 Seguimiento de los riesgos del proyecto.

SP 1.4 Seguimiento de la gestión de la información.

SG2 Gestión de las acciones correctivas a tomar.

SP 2.1 Analizar las Tareas.

SP 2.2 Tomar acciones correctivas.

SP 2.3 Gestión de las acciones correctivas.

PA 4 Gestión de acuerdo con los proveedores.

El propósito de Gestión de acuerdo con los proveedores es manejar la adquisición de productos de proveedores para los cuales ya existe un contrato formal.

El área de proceso Gestión de acuerdo con los proveedores implica lo siguiente:

- Determinando el tipo de adquisición que servirá para que los productos sean adquiridos
- Seleccionar a los proveedores
- Establecer y mantener contratos con proveedores
- Concordar con el proveedor
- Aceptar la entrega de productos adquiridos
- Efectuar el traspaso de productos adquiridos al proyecto

Esta área de proceso primordialmente se aplica a la adquisición de productos y componentes que se entregan al cliente. Para minimizar riesgos para el proyecto, esta área de proceso también puede ser ejercida para la adquisición de productos significativos y componentes que no se entregan al cliente.

Metas especificas del área:

SG1 Establecer acuerdo con el proveedor.

SP 1.1 Determinar el tipo de adquisición.

SP 1.2 Seleccionar los suministradores.

SP 1.3 Establecer un acuerdo con el proveedor.

PA 5 Medida y Análisis.

El propósito es desarrollar y soportar métricas que se usa para dar soporte a las necesidades de información de gestión.

Implica:

- Especificar objetivos de medición y análisis

- Especificar las métricas, datos a recolectar, mecanismos para almacenar datos, técnicas de análisis, reporte y mecanismos de retroalimentación
- Desarrollar la recolección. almacenamiento, análisis y reporte de los datos
- Proveer resultados objetivos que puedan utilizarse para tomar decisiones o aplicar acciones correctivas

La integración de esta área en el proceso del proyecto provee:

- Planeamiento y estimaciones objetivas
- Comparar el rendimiento actual con lo establecido en planes y objetivos
- Proveer una base de métricas para utilizar en futuros proyectos

Metas especificas del área:

SG1 Alinear mediciones y actividades de análisis

Acoplar datos y seleccionar indicadores que permitan medir la evolución de los procesos críticos del proyecto, comparar lo real con lo planificado.

S.P 1.1 Establecer objetivos de medición

SP1.2 Especificar medidas.

SG2 Proporcionar resultados de medición

SP2.1 Recoger mediciones de los datos.

SP2.3 Almacenamiento de datos y resultados

SP2.4 Comunicar resultados.

PA6 Aseguramiento de la calidad de producto y proceso

El propósito es proveer personal y gestión con una visión objetiva puesta en los procesos y productos de trabajo asociados.

Implica:

- Evaluación objetiva de procesos, productos de trabajo, y servicios utilizando descripciones de los procesos, estándares y procedimientos.
- Identificar y documentar defectos
- Proveer retroalimentación con el personal del proyecto y los gerentes resultando en actividades para asegurar la calidad.
- Asegurar que los defectos sean direccionados.

Metas especificas del área:

SG1 Evaluar objetivamente los procesos y productos de trabajo

Hay que formalizar los procesos de calidad, documentar las pruebas de caja blanca y negra.

SP1.1 Evaluar objetivamente los procesos

SP1.2 Evaluar objetivamente los productos de trabajo y servicios

PA7: Gestión de la Configuración

El propósito de Gestión de la Configuración es establecer y mantener la integridad de productos de trabajo identificando configuraciones, controlando configuraciones.

El área de Gestión de la Configuración incluye:

- Identificar la configuración de productos de trabajo seleccionados que componen la línea base en determinados puntos del tiempo
- Controlar cambios en la configuración
- Construir o proveer especificaciones para construir productos de trabajo de Sistema Gestión de la Configuración.

- Mantener la integridad de la línea base
- Proveer un estado preciso y datos actuales de configuración a los desarrolladores, usuarios finales y clientes.

Metas especificas del área:

SG1: Establecer línea base

Acompañar al software de toda la documentación necesaria para seguir su evolución, cambios, mejoras, etc.

SP1: Identificación de elementos de configuración

SP1.2: Establecer un sistema de gestión de la configuración

SP1.3: Crear o poner en marcha las líneas base

SG2: Supervisar y Controlar Cambios

SP2.1: Peticiones de supervisión de cambios.

SP2.2: Control de los elementos de configuración

4.3 Componentes de un Área de Proceso

Los componentes de un área de proceso están agrupados en tres categorías:

- Requerido
- Esperado
- Informativo

Requerido

Los componentes requeridos describen lo que debe lograr una organización para satisfacer un área de proceso. Este logro debe ser visiblemente implementado en los procesos de una organización. Los componentes requeridos en CMMI son las metas específicas y genéricas. El logro de los objetivos es usado en las apreciaciones como la base para decidir si un área de proceso ha sido lograda y satisfecha

- **Objetivo genérico**: Los objetivos genéricos asociados a un nivel de capacidad establecen lo que una organización debe alcanzar en ese nivel de capacidad.

El logro de cada uno de esos objetivos en un área de proceso significa mejorar el control en la ejecución del área de proceso

- **Objetivo específico**: Los objetivos específicos se aplican a una única área de proceso y localizan las particularidades que describen que se debe implementar para satisfacer el propósito del área de proceso.

Esperado

Los componentes esperados describen lo que implementará una organización típicamente para lograr un componente requerido. Guían a las personas que implementan mejoras o realizan mejores prácticas.

- **Práctica genérica**: Una práctica genérica se aplica a cualquier área de proceso porque puede mejorar el funcionamiento y el control de cualquier proceso.
- **Práctica específica**: Una práctica específica es una actividad que se considera importante en la realización del objetivo específico al cual está asociado.

Las prácticas específicas describen las actividades esperadas para lograr la meta específica de un área de proceso

Informativo

Los componentes informativos proveen detalles que ayudan a pensar en la manera en como lograr los componentes requeridos y esperados.

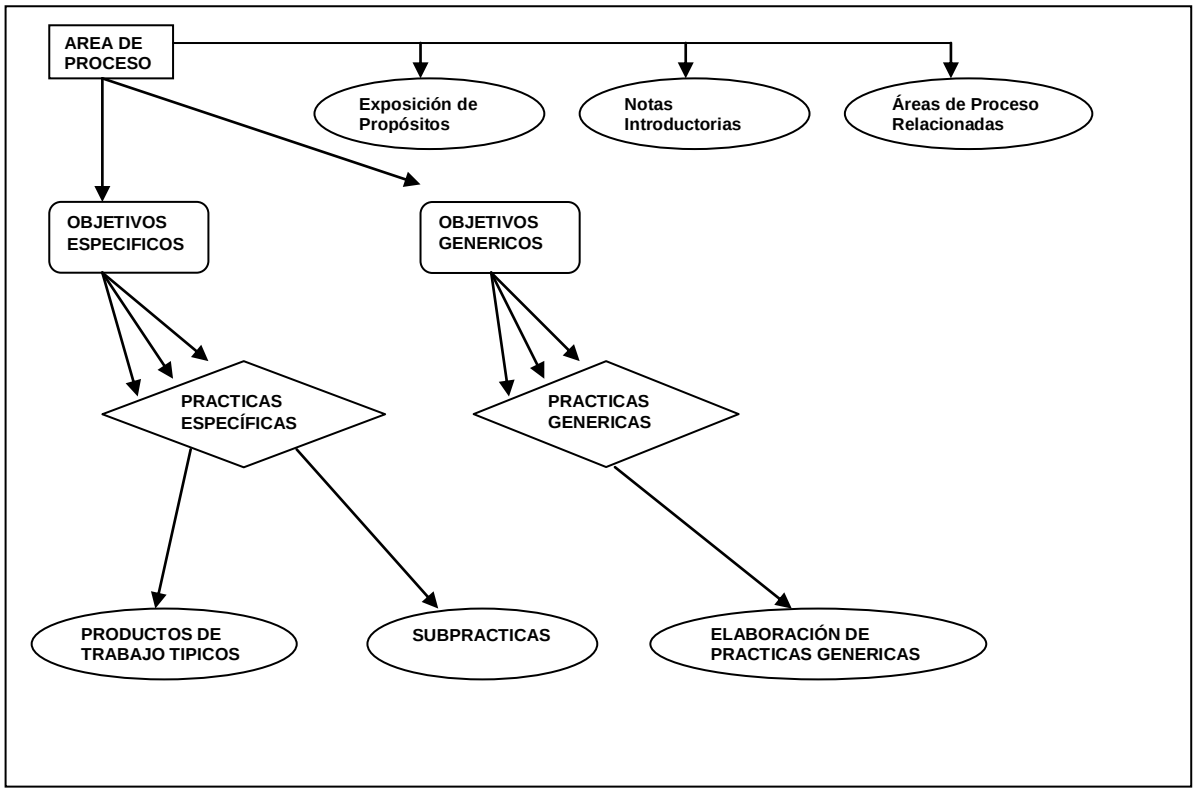


Figura 4.1 Componentes Áreas de Proceso

- **Propósito**
- **Notas introductorias**
- **Nombres**
- **Tablas de relaciones práctica - objetivo**
- **Prácticas**
- **Productos típicos**
- **Sub-prácticas**: Una sub-práctica es una descripción detallada que sirve como guía para la interpretación de una práctica genérica o especifica.
- **Ampliaciones de disciplina**: Las ampliaciones contienen información relevante de una disciplina particular y relacionada con una práctica especifica.
- **Elaboraciones de prácticas genéricas**: Una elaboración de una práctica genérica es una guía de cómo la práctica debe aplicarse al área de proceso.

CAPÍTULO 5

DEFINICIÓN Y DISEÑO DE LA METODOLOGÍA CMMI PARA LA FUERZA TERRESTRE

5.1 Capacitar al Personal en CMMI

La capacitación del CMMI se dio en el periodo de una semana, con la participación de:

- Personal de Sistemas
- Personal de definición, diseño y control de procesos

Como resultado de la capacitación se dio:

- Colaboración del personal para desarrollar la metodología
- Aceptación del CMMI
- Compromiso del personal para aplicar la metodología
- Conocimiento del personal sobre el CMMI

En conclusión la capacitación fue de tipo introductoria, donde se puedo observar el interés de los participantes.

5.2 Definición de los principales procesos de la Organización

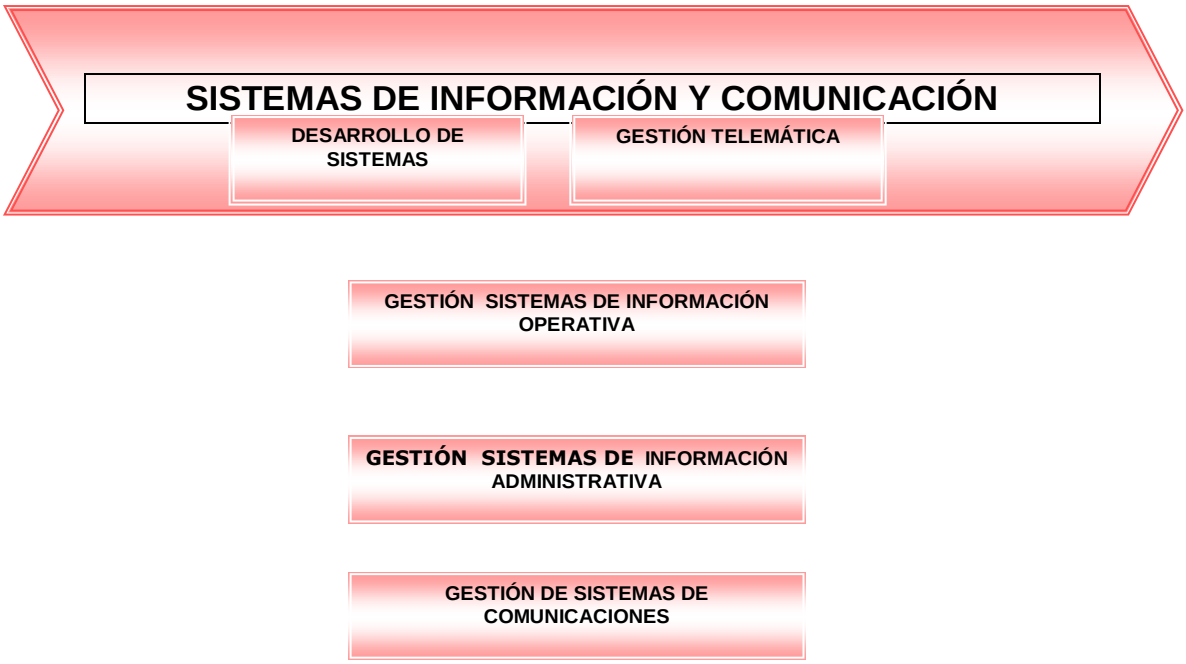


Figura 5.1 Sistemas de Información y Comunicación

5.3 Arquitectura de Procesos

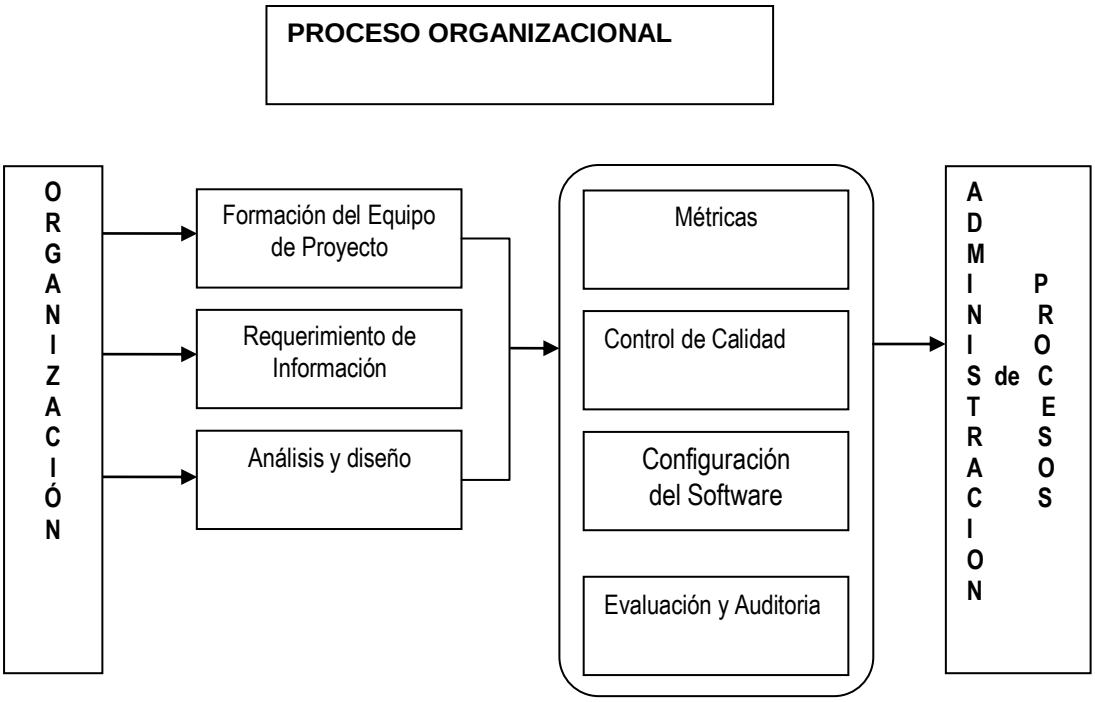


Figura 5.2 Arquitectura de Procesos

A continuación se explica la arquitectura de Procesos:

Organización

Objetivo

- Garantizar el éxito, estableciendo liderazgo, entendimiento y compromiso.

Actividades

- Seleccionar los Procesos
- Formar un Equipo de Mejoramiento
- Establecer un Equipo Ejecutivo
- Dar entrenamiento al equipo
- Comunicar Metas a los empleados

Formación del Equipo de Proyecto

Objetivo

- Formar un equipo, con el cual se busca trabajar en el enfoque de proyectos para la organización.

Actividades

- Recolectar información
- Definir metodología de trabajo
- Definir estándares
- Formar comisiones de trabajo para cada una de las áreas

Requerimiento de Información

Objetivo

- Entender el Proceso actual en todas sus dimensiones

Actividades

- Análisis del Problema : El objetivo de esta actividad es entender las verdaderas necesidades del negocio.
- Evaluación y Negociación: La diversa gama de fuentes de las cuales provienen los requerimientos, hacen necesaria una evaluación de los mismos antes de definir si son adecuados para el cliente.
- Especificación: La especificación de requisitos de software es la actividad en la cual se genera el documento, con el mismo nombre, que contiene una descripción completa de las necesidades y funcionalidades del sistema que será desarrollado; describe el alcance del sistema y la forma en como hará sus funciones, definiendo los requerimientos funcionales y los no funcionales
- Validación: La validación es la actividad que permite demostrar que los requerimientos definidos en el sistema son los que realmente

quiere el cliente; además revisa que no se haya omitido ninguno, que no sean ambiguos, inconsistentes o redundantes.

- Evolución: Los requerimientos son una manera de comprender mejor el desarrollo de las necesidades de los usuarios y cómo los objetivos de la organización pueden cambiar, por lo tanto, es esencial planear posibles cambios a los requerimientos cuando el sistema sea desarrollado y utilizado.

Métricas

Objetivo

- Establecer métricas para medir el producto en desarrollo.

Actividades

- Desarrollar mediciones y objetivos del proceso
- Definir métricas de funcionalidad, seguimiento y control
- Definir métricas de calidad que valoran y modelan los requisitos.
- Definir Técnicas que valoren los estándares.

Control de Calidad

Objetivo

- Implementar un sistema de control del proceso que posibilite el mejoramiento continuo.

Actividades

- Definir una propuesta orgánica.
- Establecer un sistema de retroalimentación.
- Detectar problemas
- Delimitar el área problemática
- Estimar factores que probablemente provoquen el problema
- Determinar si el efecto tomado como problema es verdadero o no
- Prevenir errores debido a omisión, rapidez o descuido

- Confirmar los efectos de mejora
- Detectar desfases

Configuración del Software

Objetivo

- Garantizar el funcionamiento del producto

Elementos de la Configuración del Software

- El plan de proyecto.
- El plan de Gestión de Configuración.
- El documento de definición de requerimientos.
- Estándares de análisis, diseño, codificación, pruebas, y auditoria.
- Documentos de análisis y diseño del sistema.
- Especificaciones de prueba del sistema.
- El plan de pruebas del sistema.
- El Código fuente del programa.
- Documentos de diseño de base de datos.
- Datos de prueba.
- Datos del proyecto.
- Manuales de usuario.

Actividades

1. **Implementación del Proceso**: Se desarrolla un Plan de Gestión de Configuración que describe las actividades de Gestión de Configuración, los procedimientos y el cronograma para su realización, y los responsables de dichas actividades. Dicho plan debe ser documentado e implementado.

2. **Identificación de la Configuración**: Se establece un esquema de identificación de los elementos de software.

3. **Control de la Configuración**: Se identifican y registran las solicitudes de cambio, se analiza y evalúa los cambios, se aprueba o rechaza la solicitud, se implementa, verifica y distribuye el elemento de software modificado.

4. **Contabilidad de Estado de la Configuración**: Se preparan registros de Gestión y reportes de estado que muestren el estado e historia de los elementos.

5. **Evaluación de la Configuración**: Se determina y asegura que los elementos de software sean funcionalmente (versus sus requerimientos) y físicamente completos (es decir, si su diseño y Código reflejan una descripción técnica actualizada).

6. **Gestión de actualización y distribución:** Se controla formalmente la actualización y distribución de los productos de software

Evaluación y Auditoria

- Realizar una evaluación continua y auditoria de procesos

Actividades

- Realizar periódicamente la auditoria del proceso

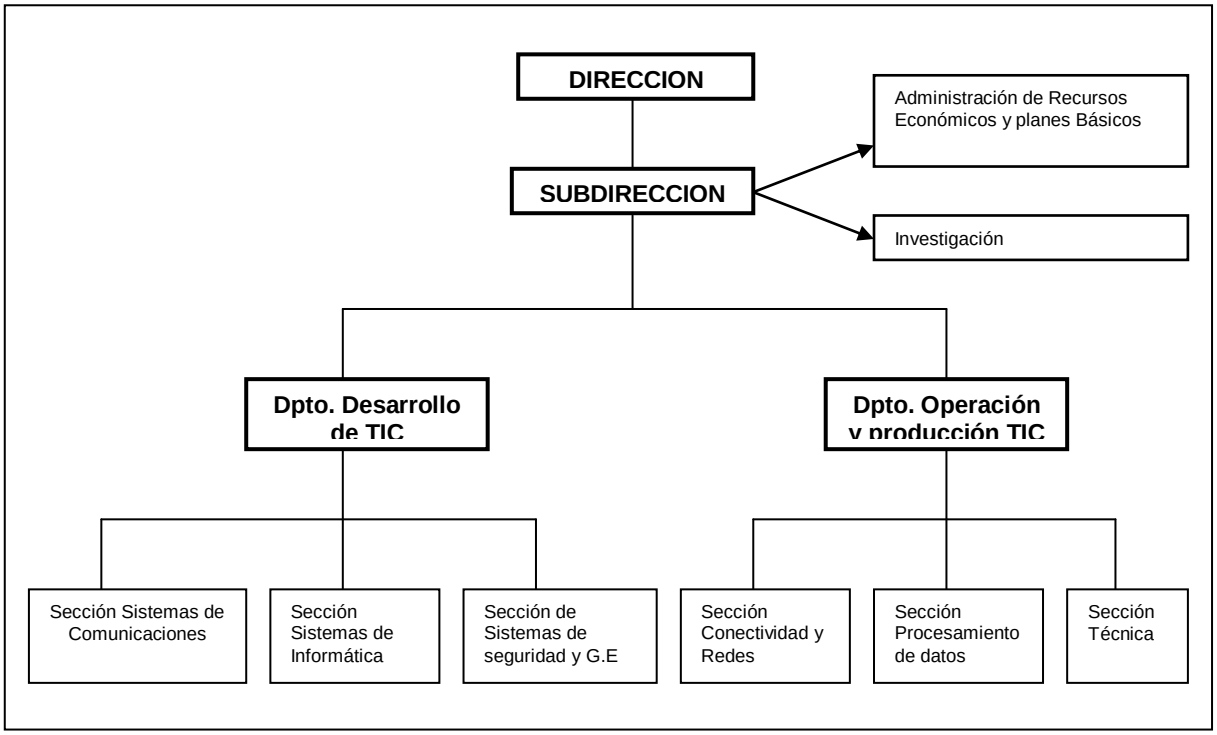
Administración de Procesos

Objetivo

La Administración de Procesos permite mejorar la productividad mediante procesos, productos, clientes claramente definidos, y la coordinación de estos dentro de la organización.

Procesos Tecnológicos

Estructura de la Organización de la Dirección de Sistemas de Información y Comunicaciones



Definición de Funciones

La organización interna de los directivos y técnicos informáticos será como se indica a continuación:

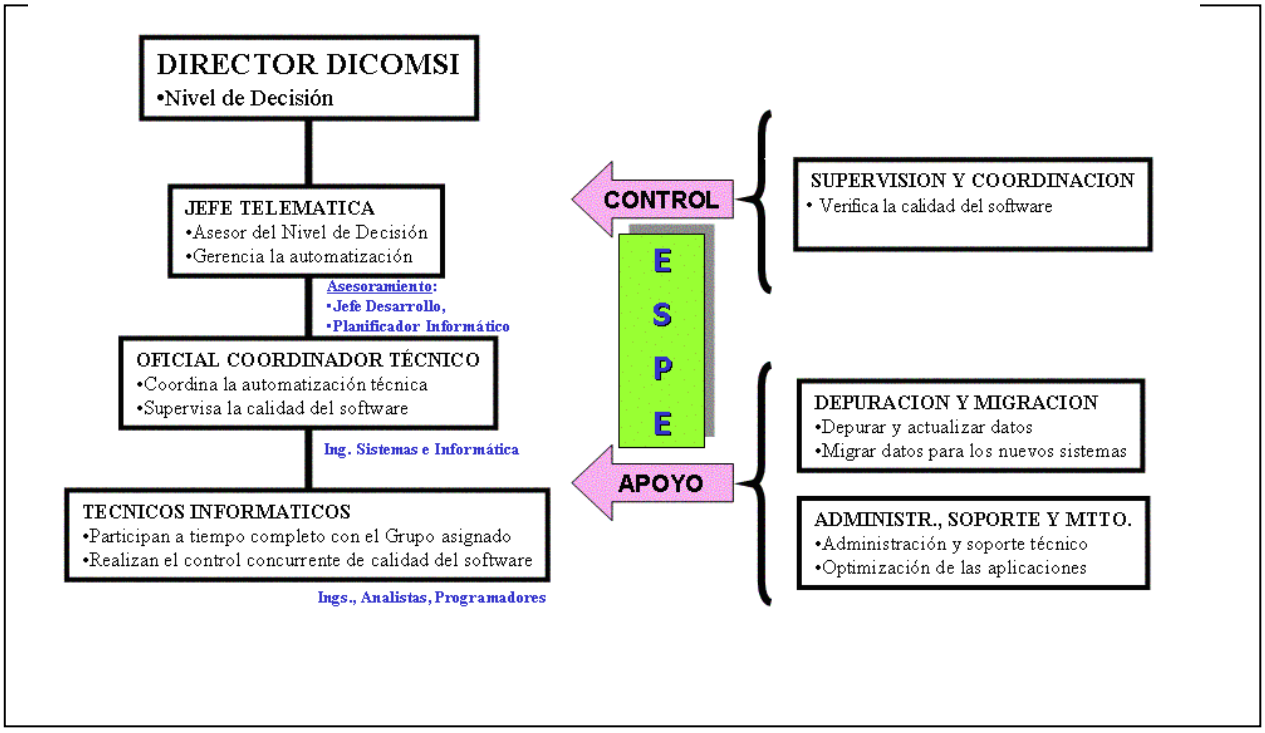


Figura 5.4 Organización Interna

Atribuciones y Deberes del Director de la DICOMSI

a.- Es la máxima autoridad de decisión por parte de la Fuerza Terrestre, la cual es compartida con el Subdirector.

b.- Es responsable de gestionar los proyectos de las dependencias del Comando General y en los niveles administrativos: Divisiones, Brigadas, Repartos, Institutos y más dependencias, para la

generación y procesamiento estratégico de información para la Fuerza Terrestre.

c.- Es responsable directo de que la automatización de la Fuerza Terrestre se cumpla de acuerdo a lo establecido en el Plan Estratégico Institucional.

Atribuciones y Deberes del Jefe de Telemática

a.- Es responsable de asesorar, planificar, coordinar el desarrollo de los diferentes sistemas".

b.- Asegurar el éxito de la automatización con la colaboración permanente del Jefe de la Sección Desarrollo de Telemática y del Planificador de Sistemas Informáticos.

c.- Es responsable directo para que la automatización de los sistemas se cumpla en los plazos previstos para todos los niveles de la Fuerza Terrestre.

d.- Implementar la plataforma tecnológica para los sistemas

e.- Coordinar en forma permanente con el Dpto. de Organización de la Dirección de Operaciones de la F.T., la implantación de los

procesos administrativos y el levantamiento de los procesos tácticos de los diferentes sistemas.

Atribuciones y Deberes del Oficial Coordinador Técnico

a.- Coordinar la automatización de los aplicativos

b.- Verificar que la automatización produzcan información gerencial para el Alto Mando.

c.- Controlar la participación de los técnicos de la Fuerza Terrestre asignados a los equipos de proyecto.

d.- Verificar que la automatización cumpla los estándares definidos por la Fuerza Terrestre.

e.- Solicitar la plataforma tecnológica para los diferentes sistemas.

f.- Presentar el Plan para la implantación de las aplicaciones en todos los niveles de la F.T.

Atribuciones y Deberes de los Técnicos Informáticos de la Fuerza Terrestre

a.- Implementar conjuntamente con los técnicos de la ESPE el sistema asignado, de acuerdo a la metodología, estándares, normas técnicas y normas de calidad para la generación de los productos de software, a entera satisfacción de la Fuerza Terrestre.

b.- Participar a tiempo completo con el grupo asignado para la automatización de un sistema.

c.- Conocer a detalle los procesos manuales y automatizados del sistema en desarrollo.

d.- Coordinar y facilitar la conceptualización de las reglas de negocio del sistema en construcción.

e.- Comprobar que se incluya en el diseño y construcción de los sistemas, los requerimientos para satisfacer en un alto grado las necesidades de los usuarios finales.

f.- Comprobar que el sistema en construcción, se halle integrado con los otros sistemas del SIFTE.

.

g.- Mantener permanentemente informado al Oficial Coordinador Técnico, sobre las novedades que se generen en la ejecución de la automatización.

h.- Participar en la implantación de las aplicaciones y a futuro dirigir el mantenimiento de los programas

i.- Poseer un conocimiento total del modelamiento de procesos; diagramas de: clases, casos de uso, secuencia, actividades, componentes, colaboración, despliegue, objetos y estado; modelo conceptual y físico; diccionario de datos; y, de la programación del sistema asignado; de tal manera de que se halle capacitado para administrar, proporcionar soporte técnico y optimizar su funcionamiento.

Atribuciones y Deberes del Supervisor y Coordinador

a.- Supervisar y coordinar el avance del proyecto, a fin de asegurar que se cumplan los plazos establecidos.

b.- Recibir asesoramiento de oficiales, voluntarios o empleados civiles especialistas del área especifica sobre el funcionamiento de un sistema específico; para alcanzar una alta funcionalidad y calidad del sistema informático construido.

Atribuciones y Deberes del Grupo de Depuración y Migración de Datos.

a.- Conducir la depuración de los datos de sistemas multiusuarios o monousuarios, en coordinación con los técnicos de la ESPE.

b.- Migrar los datos requeridos por el sistema con la asistencia técnica de la ESPE, en forma oportuna y con un alto grado de certeza.

c.- Coordinar el completamiento y la actualización de los datos en los sistemas entregados por la ESPE.

d.- Cooperar en forma permanente con los técnicos de la ESPE, a fin de que puedan realizar las pruebas individuales e integrales, utilizando datos reales.

Atribuciones y Deberes del Grupo de Administración, Soporte Técnico y mantenimiento de los sistemas

a.- Administrar, proporcionar soporte técnico y realizar mantenimiento de los aplicativos.

b.- Instalar, capacitar a los usuarios, coordinar la carga inicial de datos y administrar la explotación del nuevo sistema.

c.- Coordinar y ejecutar la optimización de las aplicaciones orientando a la modernización de los servicios y a satisfacer las necesidades de los usuarios finales del sistema.

d.- Manejar eficientemente la instanciación y ciclo de vida de los componentes; conexiones a la base de datos; transacciones; y, seguridad de los sistemas en explotación.

5.4 Desarrollar políticas organizacionales

Políticas de Ingeniería

Requerimientos

- Los requerimientos del producto deben ser definidos, documentados y aprobados por el cliente antes de su desarrollo.
- Los requerimientos deben ser validados de acuerdo a las especificaciones hechas por el cliente antes de entregarlo.
- La fase de desarrollo no debe iniciarse sin una propuesta y una especificación de requerimientos documentados y aprobados.
- La especificación de requerimientos debe ser establecida para determinar si el desarrollo del producto es factible.

Diseño

- El diseño de los productos de software se realizará bajo el modelo UML y herramientas CASE como Power Designer
- La metodología que se utilizará para el diseño será la Metodología RUP con UML
- Elaboración de diagramas de casos de uso
- Elaboración de diagramas de secuencia

Programación

- La programación de los productos se hará en tres capas con los lenguajes : power builder, JAVA y con las bases de datos informix, oracle y sql Server
- Las bases de datos deberán ser portables para las plataformas UNIX SOLARIS, LINUX y Windows 2003

Pruebas

- Se debe preparar los Archivos de Prueba
- Se deben realizar Pruebas Unitarias
- Se deben realizar Pruebas de Integración
- Se deben realizar Pruebas de Validación

Pruebas de Caja Negra

Pasos:

- Preparación de Archivos de prueba
- Pruebas de Valores Límites

Pruebas de Entornos y Aplicaciones

Pasos:

- Pruebas de Interfaz Gráfica de Usuarios
- Pruebas de Documentación

Políticas de Soporte Técnico

Elaboración del Plan de Implantación

- Elaboración del Plan de Capacitación
- Ejecución del Sistema
- Pruebas de Documentación

Evaluación Periódica de Sistemas

- Plan de Control de Rendimiento del Sistema
- Plan de Control de Integridad del Sistema

- Plan de Mantenimiento
- Elaboración de nuevas solicitudes

Políticas para Administración de Proyectos

- Se debe firmar el Convenio específico para el desarrollo del proyecto.
- El proceso de desarrollo de la especificación técnica debe estar acuerdo a los requerimientos técnicos proporcionados por el investigador responsable.
- Llevar a cabo los procedimientos para cada proyecto de la organización.
- Elaborar y enviar en tiempo y forma los informes sobre el proyecto que correspondan cada etapa.
- Integrar y conservar el expediente completo y actualizado de los asuntos administrativos del proyecto.
- Atender directamente a los auditores y proporcionarles la información y documentación pertinente, así como las facilidades para cumplir con su cometido.

Políticas de aseguramiento de la Calidad

- Realizar auditorias de calidad interna periódicamente
- Las no conformidades de los procesos encontradas en las auditorias internas deben ser analizadas para determinar la causa de las mismas. Las no conformidades están sujetas a una nueva revisión.
- Satisfacer los requerimientos acordados con el cliente.
- Una no conformidad de tipo técnico debe solucionarse antes de continuar con la siguiente fase
- Se definirán procedimientos en aquellos aspectos donde sea posible regular cada uno de los pasos a dar y las alternativas posibles a evaluar.
- Establecer plazos fijos de entrega para realizar las pruebas correspondientes al producto.

Métricas

- Se deben usar métricas de estimación, calidad y control para la efectividad del producto a desarrollarse.

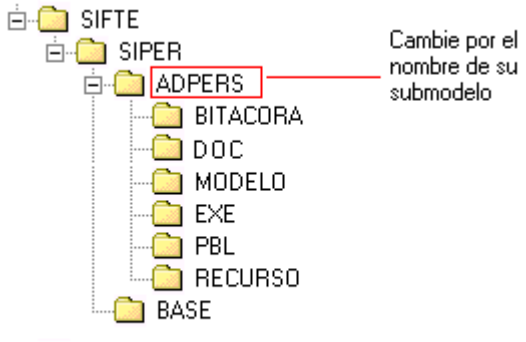
- Mantener un registro de históricos de métricas.
- Definir métricas para el control del desarrollo del producto.
- Tomar acciones sobre los resultados de las métricas.

5.5 Definir especificaciones de productos de software

5.5.1 Estándares Cliente y Componentes

Los estándares[4] establecidos para el desarrollo del sistema se indican a continuación:

Estructura del Directorio



La estructura de directorios para la ubicación de los productos del desarrollo es la siguiente:

C:\siper\admin\pbl*proceso* {librerias del proceso}

[4] Estándares establecidos del proyecto SIFTE de la Fuerza Terrestre

C:\siper\admin\exe {ejecutable de la aplicación}

C:\siper\admin\ini {configuración de inicio de la aplicación}

C:\siper\admin\documentos {manuales, análisis, diseño, actas}

C:\siper\admin\bmp {imagenes}

C:\siper\admin\recursos {componentes, instaladores}

C:\siper\admin\pfc {referencias}

Los nombres de librerías seguirán el siguiente formato p_pi_objeto; donde:

p_i es una abreviación del nombre de los distintos procedimientos

objeto define la siguiente lista de objetos:

aplicación (aplicación, objetos de usuario, estructuras)

datawindow (otros datawindow)

dropdown (datawindow tipo dropdown)

función (funciones)

menú (menús)

otro (otros objetos)

reporte (datawindow tipo reporte)

seguro (objetos del modulo de seguridad y auditoria)

ventana (ventanas de la aplicación)

Los nombres de objetos almacenados dentro de las librerías se escribirán en minúsculas con el siguiente formato:

Ventana: w_nombre, pero también se sugiere la siguiente codificación que puede ser utilizada por el programador a su conveniencia.

wm_nombre	Mantenimiento
wr_nombre	Reporte
wd_nombre	Maestro – detalle
wp_nombre	Principal

Datawindow: d_nombre, pero también se sugiere la siguiente codificación que puede ser utilizada por el programador a su conveniencia.

dm_nombre	Mantenimiento
dr_nombre	Reporte
dt_nombre	Dropdown
de_nombre	Externo
ds_nombre	Maestro
dd_nombre	Detalle

Menú: m_nombre (se utilizará un solo menú para implementar las seguridades)

Aplicación: ap_nombre

Función: f_nombre

Función de ventana: wf_nombre

El programador puede incluir dos letras y un "_" antes del nombre del objeto para identificar de manera única el objeto.

Los nombres de las variables se definen de la siguiente manera at_nombre, donde:

a es el alcance de la variable como se indica en la tabla.

Alcance	**Letra**
Local	L
Instancia	I
Shared	S
Global	G
Argumento	A

t es el tipo de dato de la variable como se indica en la tabla.

Tipo	**Letras**
Integer	I

Long	L
Double	Db
Decimal	Dc
String	S
Date	D
Datetime	Dt
Datawindow	Dw
Datawindowchild	Dwc
Time	T

Inicialmente se puede colocar comentarios en el script pero para la generación de los ejecutables se retiran todos los comentarios. Documentación interna mínima.

En la cabecera del scrip se deberá agregar la siguiente información:

- Descripción de la funcionalidad.
- Funciones que llama, variables importantes, tablas y campos que usa.
- Descripción del proceso.
- Fecha Última modificación
- Responsable original.
- Responsable cambios.

Interfaces de Usuario

Menú

Utilizar un menú como el que se describe a continuación:

Archivo	Edición	Mantenimiento	Procesos	Reportes	Ventana	Ayuda
	Primero Anterior Siguiente Ultimo Insertar Borrar Selección Copiar Pegar Cortar				Horizontal Vertical Cascada Capa Ordenar Iconos	

5.5.2 Estándares Elaboración de base de datos

Convenciones

Se ha considerado conveniente adoptar las siguientes convenciones para identificar los estándares de base de datos.

( ) = Identifica la constante de la categoría del atributo o del objeto a definir en la base de datos.

{ } = Indica que se debe escoger al menos una de las opciones mostradas dentro de las llaves.

[ ] = Es opcional y se puede escoger una o más de las opciones mostradas dentro de los corchetes.

_ = Underscore, usado para separar los elementos identificados como categorías, tipos y opciones de objetos, en este manual.

No se usará como separador en cadenas de caracteres correspondientes a nombres de datos definidos por el usuario.

..... = Indica que el nombre del elemento puede variar en longitud.

nnn = Sustituciones, representado por caracteres itálicos indicando que el elemento está compuesto por caracteres provistos por el usuario.

kkk = Constantes representadas por caracteres tipo romano.

Esta guía usará el término de objeto para referenciar a los distintos componentes de una base de datos.

Las identificaciones de los objetos en la base de datos se deberá sujetar al siguiente esquema:

(nnn),x, y / z, mmmmmmm.

⇓ ⇓ ⇓

Nombre asignado Opciones Categoría

Los parámetros opcionales pueden estar separados entre comas (,) o barras, de acuerdo a la siguiente convención:

Separados con comas, cuando se puede seleccionar uno o más parámetros

Separados con barras, cuando se debe seleccionar uno de ellos únicamente.

Medio Ambiente de Trabajo (RED)

a.- Servidores de Base de Datos

nnn.._uu_yyy donde:

nnn= Identificador de las siglas del reparto

uu= Número del servidor

yyy= Identifica el tipo de servidor

dat= Servidor de datos

bac= Servidor de backup

com =Servidor de comunicación, etc.

En el caso de existir más servidores, abrevie con los 3 primeros dígitos de su tipo de servidor.

Ejemplo : disift_01_dat

b.- Clientes

nnn.._mmm_uu donde:

nnn= Identificador del reparto

mmm= Nombre del área de trabajo

uu= Número de estación

Ejemplo : disift_des__01

c.- Usuarios y Login

n{n}mmmm..... donde:

n= Es la letra del primer nombre del usuario.

{n} = Es la primera o más letras del segundo nombre del usuario, para evitar duplicados.

mmm= Es el apellido del usuario

Ejemplo :pamolina

Para grupos de usuarios: **El nombre del grupo será de acuerdo a la función a definirse**

mmmmm..... donde:

mmm..=Palabras descriptivas y significativas

Ejemplo : administrativo,logístico

Medio Ambiente de Base de Datos

a.- Base de Datos:

mmmm donde:

mmmm= Nombre descriptivo de la base de datos

b.- Esquemas:

nn_mmmm....._ddd donde:

nn = Mnemónico del sistema

mmmm....= Nombre descriptivo del subsistema

ddd = Indica el uso de la base de datos.

des=desarrollo

pru= pruebas

pro= producción

ent= entrenamiento

Objetos de la Base de Datos

Escriba en singular, todo en minúscula, utilice hasta 30 caracteres

a.- Tablas

Nombre de entidades: Nombre representativo de la tabla

Utilice el siguiente estándar.

mnnn_mmm... donde:

m = Es la letra del Código del subsistema , ejm.

p =Personal

nnn= Identificador del objeto (pasa al atributo)

mmm = Nombre del objeto

b.- Tablas temporales:

tmp_ nnnnnnn.... donde:

tmp= Prefijo que identifica tabla temporal

nnnnn..= Nombre de la tabla.

Ejemplo : tmp_logistico

c.- Relaciones

n(r)_nnnn_mmmm.. donde:

n= Es la letra del Código del Subsistema

(r)= Letra que identifica que es una relación

nnn = Identifica entidad padre

mmm= Identifica entidad dependiente

Se puede utilizar agrupaciones de caracteres para lo cual se usará el UNDERSCORE (_) como separador.

Además dentro de la relación documente lo siguiente:

Label: describa la relación existente entre las entidades.

Role: describa utilizando verbos de la siguiente manera:

Emite

Tiene

Corresponde

Pertenece, etc.

d.- Atributo

nnn_mmmm..... donde:

nnn=Identifica los 3 primeros dígitos del nombre del identificador del objeto a la cual pertenece el atributo.

mmmm..= Es el nombre asignado por el usuario para agrupaciones de caracteres se usará el UNDERSCORE como separador. El nombre deberá tener significado para el usuario.

Ejemplos : pro_codigo

e.- Constraints

nnnn...._pk | fk.. donde:

nnnn...=Nombre del dato sobre el cual se aplica el constraint. { pk | fk Representa el tipo de constraint:

El formato para un constraints es el siguiente

pk = de clave primaria

fk = de clave foránea.

Ejemplo : pro_codigo pk

f.- Índices

Para índices que no han sido generados automáticamente por POWER DESIGNER

idx_ u _c | nc _nnnnn..... donde:

idx =Identificación el índice

u= Identificación de índice único (unique)

c=Indica que es un índice clusterizado (cluster)

nnnn...= Nombre del atributo

Ejemplo : idx_c_cod_cta_cte.

g.- Procedimientos Almacenados (Stored Procedures)

sp_mmm.... donde:

sp= Constante que indica Store Procedure

mmmn...= Es el nombre del stored procedure

Se puede utilizar agrupaciones de caracteres para lo cual se usará el UNDERSCORE (_) como separador.

Ejemplo : rsp_dig_verf

Todo stored procedure deberá tener un encabezado constituido por un conjunto de comentarios que deben ir al inicio del programa y que debe incluir la siguiente información: archivo, stored procedure, base de datos, texto indicativo del producto, nombre del programador, fecha de codificación, descripción del programa, fechas de modificaciones y objetivo de las mismas.

h.- Función (Function)

f_mmm.... donde:

f= Constante que indica Función

mmmn.....= Es el nombre de la función

Se puede utilizar agrupaciones de caracteres para lo cual se usará el UNDERSCORE (_) como separador.

Ejemplo : f_dig_verf

Toda Función deberá tener un encabezado constituido por un conjunto de comentarios que deben ir al inicio del programa y que debe incluir la siguiente información: archivo, función, base de datos, texto indicativo del producto, nombre del programador, fecha de codificación, descripción del programa, fechas de modificaciones y objetivo de las mismas.

i.- Trigger

tr_nnnnn....._mmmm... donde:

tr= Identificación de trigger,

nnn...=Nombre-del-trigger

Se pueden utilizar agrupaciones de caracteres para lo cual se usará el UNDERSCORE (_) como separador.

mmm... Nombre de la tabla.

Ejemplo: tr_sueldo_logistico

j.- Vistas

v_nnnnnn..... donde:

v= identificación de vista

nnn= nombre de la vista

Ejemplo : v_ctasctes

Medio Ambiente de Programación

Comentarios

#nnnnnnnnnnnn..... ó

--nnnnnnnnnnnn.....

Se debe crear una línea de comentario para cada sentencia o grupo de sentencias PL/SQL, que realicen un proceso o trabajo específico.

También se usarán para documentar una sentencia o conjunto de sentencias SQL, o para hacer notar que se ha ejecutado una determinada acción.

El formato para autodocumentar sentencias SQL, es:

Comandos SQL **# objetivo de las sentencias**

Ejemplo: if SQLCODE = 0 then # Verifica ejecución de la transacción

Scripts

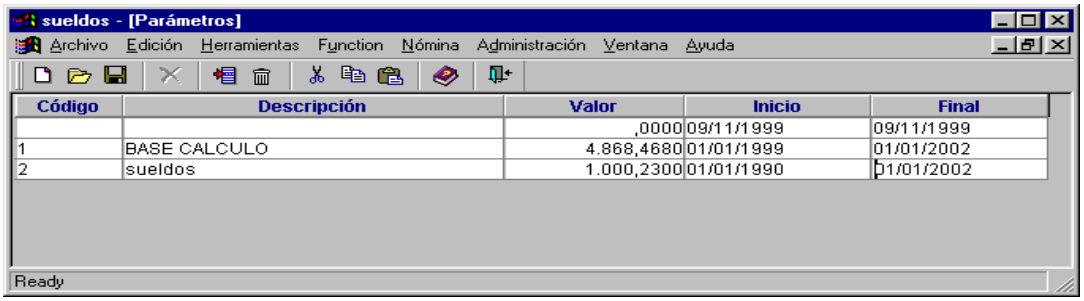
nnnnnnn..... .extensión donde:

nnn.....=Nombre del script, hasta 10 caracteres

extensión:=Identifica la extensión de la base de datos que se este utilizando

Ejemplo : logistico.sql

Ventana de Mantenimiento



Título de la ventana: Tipo oración

Posición: X = 0 , Y = 0, Maximizado

Estilo de presentación: Grid

Etiquetas:

Letra: 9, Arial, Bold, Navy, Centrado,

Fondo: Transparente

Borde: 3D Raised

Posición: Y = 8, Alto = 60

Campos:

Letra: 9, Arial, Regular, Black

Fondo: Blanco

Borde: Ninguno

Posición: Y= 0, Alto = 60

Datawindow:

Color: ButtonFace

Borde: Ninguno

Posición: X = 0, Y = 0 en la ventana.

Cabecera:

Alto: 72

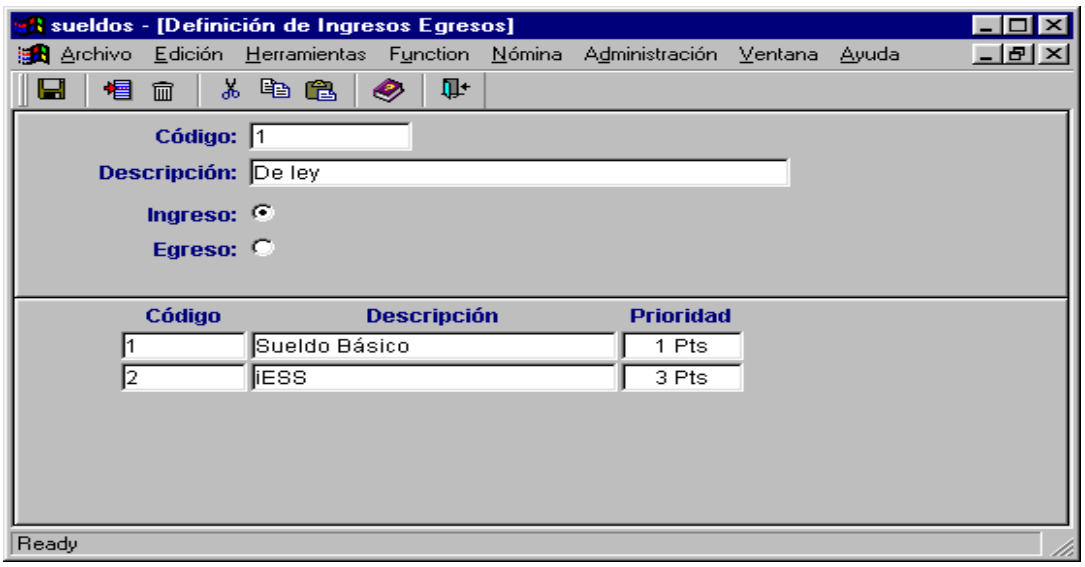
Color: Transparente

Detalle:

Alto: 64

Color: Transparente

Ventana Maestro Detalle



Título de la ventana: Tipo oración

Posición: X = 0 , Y = 0, Maximizado

Maestro:

Estilo de Presentación: FreeForm

Etiquetas: Letra: 9, Arial, Bold, Navy, Alineado a la Derecha, Fondo: Transparente, Borde: Ninguno

Campos: 9, Arial, Regular, Black, Fondo Blanco, Borde 3D Lowered

Datawindow: Color: ButtonFace, Borde: 3D Raised

Detalle:

Estilo de Presentación: Tabular

Etiquetas: Letra: 9, Arial, Bold, Navy, Centrado, Fondo: Transparente, Borde: Ninguno

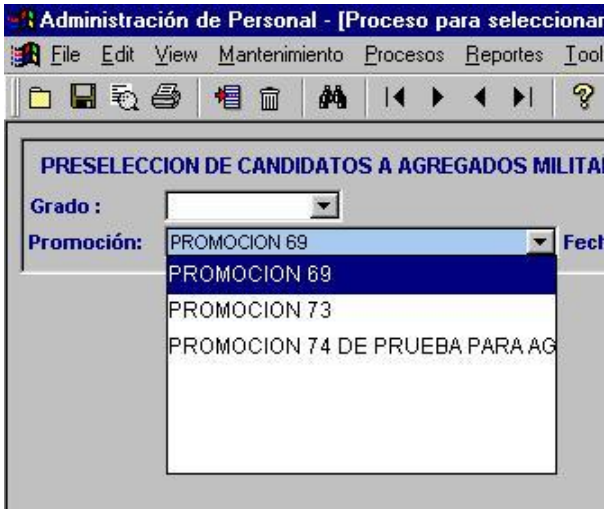
Campos: 9, Arial, Regular, Black, Fondo Blanco, Borde 3D Lowered

Datawindow: Color: ButtonFace, Borde: 3D Raised

Cabecera: Alto: 76, Color: Transparente

Detalle: Alto: 84, Color: Transparente

Dropdown



Sin Etiquetas

Posición del primer campo: X = 5 , Y = 0

Alto de los campos 60

Sin separación entre campos

Estilo de presentación: Tabular

Borde: Ninguno

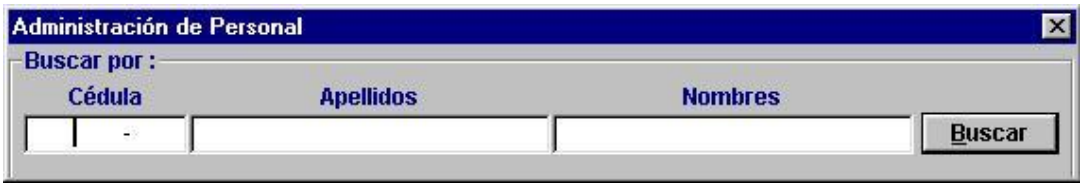
Campos: 9, Arial, Regular, Black, Fondo Blanco, Borde Ninguno

Datawindow: Fondo Blanco

Ordenar el datawindow

Alto de la sección de detalle: 64

Ventana Buscar



Titulo: Nombre del sistema

El diseño será similar al presentado en el gráfico.

El tab order será : Cédula, Apellidos, Nombres, Botón Buscar.

El primer campo permite ingresar la cédula, este campo posee una máscara para separar el último dígito.

El segundo campo permite ingresar el apellido a buscar, en caso de existir varias coincidencias se debe abrir la ventana permitiendo ver la lista de coincidencias.

El tercer campo permite ingresar el nombre a buscar, en caso de existir varias coincidencias se debe abrir la ventana permitiendo ver la lista de coincidencias.

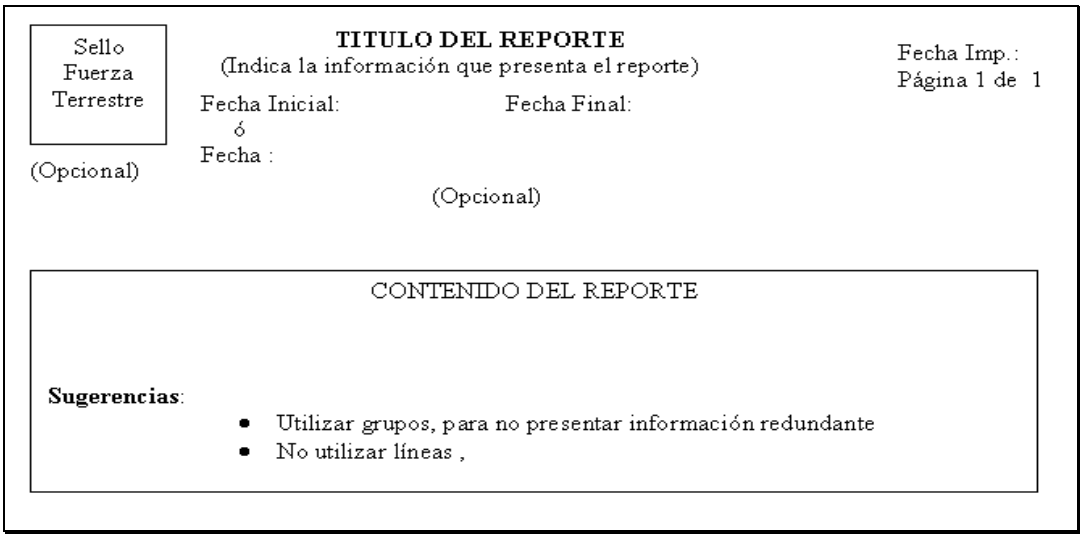
El botón buscar inicia la búsqueda en caso de existir coincidencias exactas devuelve el número de cédula a la ventana que lo solicitó. Caso contrario muestra la lista de coincidencias. De esta lista se puede escoger un nombre o realizar de nuevo búsquedas.

En caso de no existir coincidencias debe dar un mensaje de error.

Reportes

- Título del reporte en mayúsculas y con negrillas.
- El sello de la Fuerza terrestre se pondrá, solo si el reporte impreso representa un documento oficial, en la parte superior izquierda.
- La fecha y hora de Emisión se colocará en la parte superior derecha, sin negrillas

- El número de página debe colocarse en la parte superior derecha, bajo la fecha y hora.
- El nombre del módulo que imprime el reporte debe imprimirse en la Parte inferior izquierda y con mayúsculas.



Tablas

- Comentarios en tablas, en atributos y en relaciones
- Relaciones con nombres y código.
- Los tipos de datos se ajustan en lo posible a los dominios.

Estándares cliente

Pantalla Principal

3 Frames:

Frame Izquierdo:

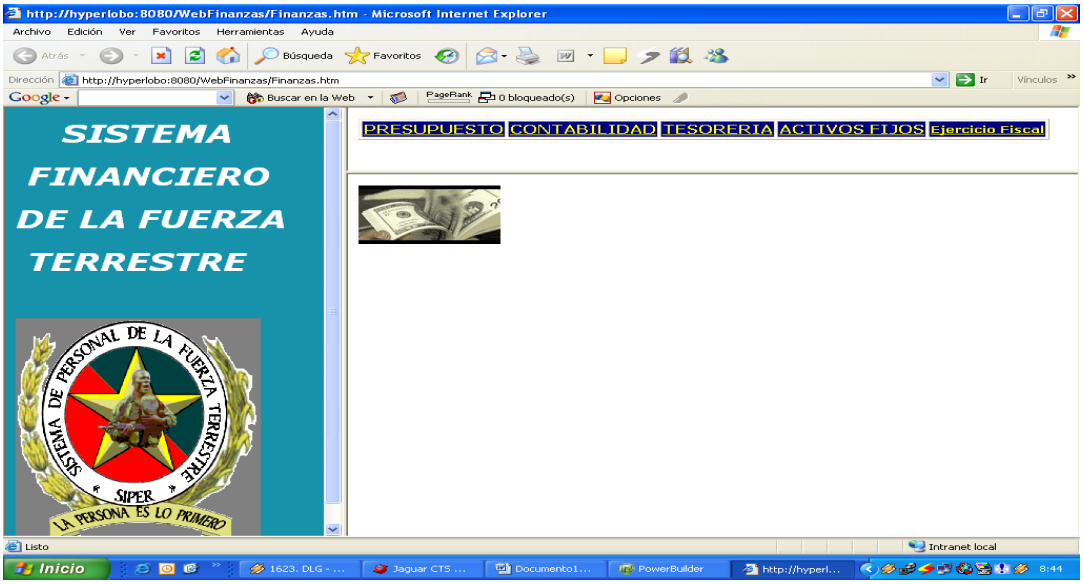
- carátula de presentación del módulo
- Fondo Turqueza Obscuro

Frame Derecho Superior

- Fondo Blanco
- Menú principal con Hypervínculos
- Letra Mayúscula, Amarilla, verdana Normal 12
- Fondo Azul Obscuro.

Frame Derecho Inferior:

- Fondo Blanco
- Gif representativo en la primera carga
- Cada Item del Menú superior con un gif representativo en este Frame



Pantallas de los Items del Menú Principal

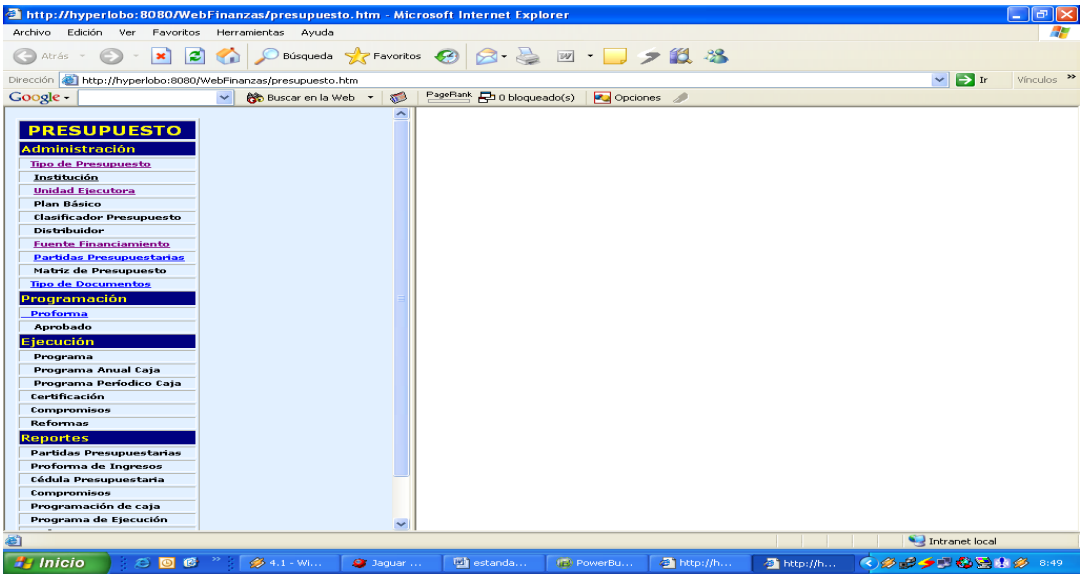
2 Frames

Frame Izquierdo

- Fondo celeste
- Menu encerrado en celdas
- Los Títulos Principales Con fondo en la celda Azul Obscuro
- Los Items con fondo de color igual al fondo del frame
- Las letras de los títulos color amarillo
- Letra Titulo Principal Mayúscula, Amarilla, verdana Heading1 14pts
- Letra títulos inferiores primera letra mayúscula, Amarilla, verdana Heading1 10pts
- Items primera letra de cada palabra mayúscula color Azul claro, verdana Heading1 8pts.
- Items Con Hypervínculos

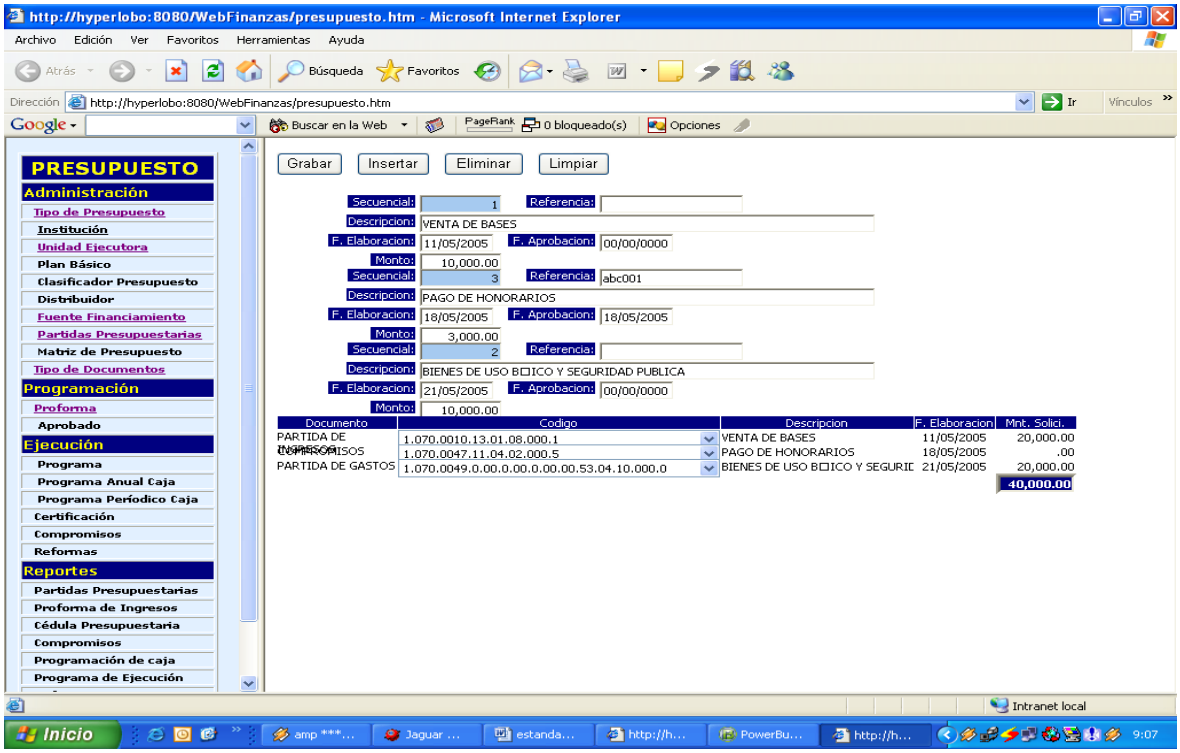
Frame Derecho:

- Fondo Blanco
- Se presenta el despliegue de los Items del Frame Izquierdo



Frame Derecho

- Fondo Blanco
- Botones de elección Grabar, Insertar, Eliminar y Limpiar en la parte superior izquierda. Primera letra mayúscula
- Titulo de detalle en recuadro azul con letra blanca,.
- Despliegue de información fondo blanco, Tahoma 8 pts normal. Borde Lowered(5)
- Maestro detalle cada columna recuadro azul con letra blanca Tahoma 8 pts normal.
- Detalles letra negra Tahoma 8 pts normal.



Generalidades:

- Los nombre de botones inician con cb_nombre boton, todo con minúsculas

Requerimientos

- Elaboración de la Solicitud de Automatización
- Aprobación de la Solicitud de Automatización
- Planificación del estudio preliminar
- Investigación preliminar
- Relevamiento del proyecto
- Elaboración de tablas de problemas generales
- Determinación de la factibilidad del proyecto
- Estimar requerimientos de tiempo y costos de inversión para la realización del estudio detallado.
- Elaboración del informe de investigación preliminar
- Evaluación del informe de investigación preliminar
- Formalizar estudio detallado

Diseño

Análisis y Diseño Exterior

- Descripción de los objetivos del sistema
- Identificación de subsistemas
- Identificación de la comunidad de usuarios del nuevo sistema
- Identificación de entidades externas del nuevo sistema
- Elaboración de los Diagramas UML
- Diagrama de casos de uso
- Análisis de requerimientos funcionales
- Modelo de procesos
- Especificación de requerimientos
- Descripción general
- Especificación de requerimientos funcionales
- Requerimientos de interfaces externas
- Requerimientos de rendimiento
- Restricciones de diseño
- Atributos
- Evaluación financiera de alternativas
- Estimar costos de inversión por cada alternativa
- Estimar costos y tiempo de desarrollo por cada alternativa
- Estimar costos de operación por alternativa
- Estimar beneficios tangibles por alternativa
- Estimar beneficios intangibles y vida útil del sistema por cada alternativa
- Análisis de factibilidad financiera por alternativa
- Selección de alternativas
- Métricas de análisis

Análisis y Diseño Interior

- Elaboración de diagramas UML
- Estructuración de manuales de usuario

- Construcción de la estructura lógica de datos
- Describir la lógica detallada de los procesos
- Diseño de entradas y salidas
- Diseño de la base de datos física
- Creación del modelo arquitectónico
- Especificaciones de los procesos que van a ser programados y su jerarquía modular
- Planificación general de la fase de construcción e implantación
- Revisión de la fase de análisis y diseño
- Métricas de diseño

Pruebas

Pruebas de Caja Blanca

- Preparación de Archivos de Prueba
- Pruebas Unitarias
- Pruebas de Integración
- Métricas de Diseño
- Pruebas Generales
- Pruebas de Validación
- Pruebas Alfa
- Pruebas Beta
- Pruebas del Sistema
- Hito de Control

Pruebas de Caja Negra

- Preparación de Archivos de prueba
- Pruebas de Valores Límites
- Pruebas de Comparación
- Hito de Control

Pruebas de Entornos y Aplicaciones

- Pruebas de Interfaz Gráfica de Usuarios
- Pruebas de Documentación

Documentación

Elaboración de la Documentación Operacional

- Manual de Usuario
- Manual Técnico

- Manual de Procedimientos

5.5.3 Metodología de Desarrollo de Software

Para el desarrollo de software aplicativo, la Fuerza Terrestre utilizará la Metodología RUP.

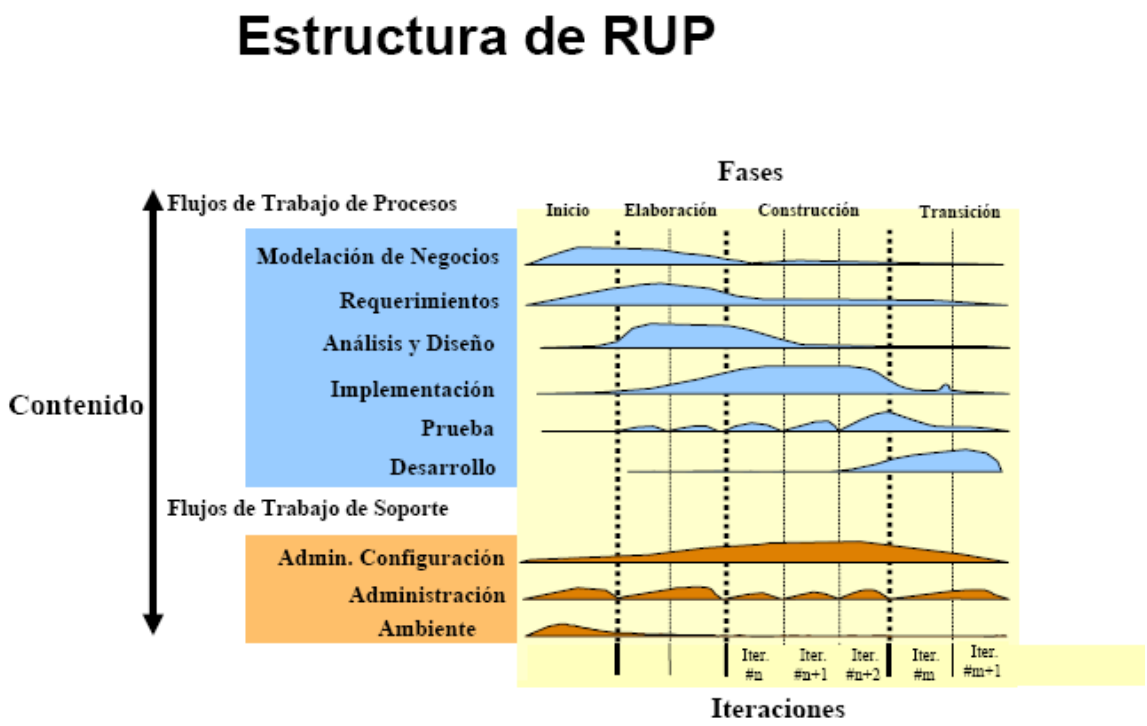


Figura 5.5 Estructura de la Metodología RUP

El RUP es un proceso de ingeniería de software que utiliza el paradigma de orientación a objetos para su descripción.

Es un marco de proceso configurable para satisfacer necesidades específicas e implementar las mejores prácticas de desarrollo de software.

Posee tres características principales:

- Dirigido por casos de uso

  Los casos de uso capturan requerimientos funcionales y representan piezas de funcionalidad que brindan un resultado de valor al usuario.

- Centrado en una arquitectura

Comprende los aspectos estáticos y dinámicos más importantes del sistema.

- Iterativo e incremental

  El trabajo se divide en piezas pequeñas o miniproyectos; cada uno proveyendo un subproducto incremental.

5.5.4 Plan de calidad SQA

1. Introducción

El beneficio primario de un programa de SQA es garantizar a la gerencia ejecutiva que el proceso de desarrollo de software definido formalmente está implementado y en uso. Más específicamente, asegura que:

- Se trabaja con una metodología apropiada de desarrollo.
- El proyecto usa estándares y procedimientos en su trabajo.
- Se conducen revisiones y auditorias independientes.
- Se produce documentación para garantizar el futuro mantenimiento.
- La documentación se produce durante y no después del desarrollo.
- Existen mecanismos, y son usados, para controlar los cambios.
- El testing se enfatiza en las áreas de producto de alto riesgo.
- Cada tarea en el proceso de desarrollo se completa satisfactoriamente antes de que la siguiente comience.
- Las desviaciones de los estándares y procedimientos se exponen lo antes posibles.
- El proyecto es auditable por profesionales externos en cualquier momento.
- La tarea de control de calidad se ejecuta siguiendo estándares previamente establecidos.
- El plan de SQA y el plan de desarrollo son compatibles.

2. Objetivos

Los principales objetivos del Aseguramiento de la Calidad del Software son los siguientes:

- Mejorar la calidad del software monitoreando apropiadamente tanto los productos de software como el proceso de desarrollo que los genera.

- Asegurar el cumplimiento de los estándares y procedimientos establecidos para el software y el proceso de software establecidos.
- Asegurar que cualquier desviación en el producto, el proceso, o los estándares son elevados a la gerencia para poder resolverlas.

3. Rol del SQA

Las personas responsables del proyecto de software (desarrollo y cliente) son las únicas que pueden ser responsables por la calidad. El rol de SQA es monitorear la manera en que estos grupos ejecutan sus responsabilidades. Por lo tanto existen los siguientes peligros latentes:

- Es un error asumir que el personal de SQA puede por sí solo hacer algo por la calidad del proyecto.
- La existencia de una función de SQA no asegura que se siguen los estándares y los procedimientos.
- Sólo si la gerencia demuestra periódicamente su soporte a SQA, siguiendo sus recomendaciones, SQA podrá ser efectiva.
- A menos que la gerencia de línea requiera que SQA trate de resolver sus no-conformidades con la gerencia del proyecto antes de elevarlas, SQA y desarrollo no trabajarán efectivamente.

Todo lo que puede hacer SQA es alertar a la gerencia sobre las desviaciones a los estándares y procedimientos establecidos. La gerencia debe entonces insistir acerca de que los problemas de calidad se solucionen antes de que el producto sea liberado para su uso, sino SQA se transforma en un ejercicio burocrático y costoso.

4. Funciones de SQA

Las principales funciones del rol de SQA, a través de todo el ciclo de vida, son las siguientes:

- Prácticas de QA: se definen y están disponibles herramientas, técnicas, métodos y estándares de desarrollo adecuados para ser usados como estándares de las revisiones de QA.

- Evaluación de la planificación del proyecto de software: si no se planifican prácticas de calidad adecuadas desde el inicio y sincronizadas con el plan del proyecto, luego no serán implementadas.
- Evaluación de los requerimientos: como es extremadamente inusual que se desarrollen productos de alta calidad a partir de requerimientos de baja calidad, los requerimientos iniciales deben ser revisados contra los estándares de calidad establecidos.
- Evaluación del proceso de diseño: se definen los medios para asegurar que el diseño sigue las metodologías planificadas, que implementa los requerimientos y que la calidad del diseño propiamente dicha es revisada independientemente.
- Evaluación de las prácticas de codificación: prácticas apropiadas de codificación deben ser establecidas y usarse.
- Evaluación del proceso de integración y testeo del software: se establece un programa de testing de calidad, el testing es ejecutado por un grupo independiente que es tanto capaz como está motivado para encontrar problemas, la planificación del testing comienza en las primeras etapas del proyecto, y se revisa la calidad del testing propiamente dicho.
- Evaluación del uso del proceso de control y gerenciamiento del proyecto: asegurando que los procesos de gerenciamiento están funcionando, SQA ayuda a garantizar que todo el grupo de proyecto está orientado a producir resultados de calidad.
- Adaptación de los procedimientos de SQA: El plan de SQA debe ser adaptado a las necesidades específicas del proyecto.

5. Destinatarios

Se presenta un cuadro para distribuir la información.

Destinatario	**Departamento**
	Gerente Ejecutivo
	Líder de Proyecto
	Responsable de Calidad
	Miembro de Calidad

6. Organización del equipo SQA

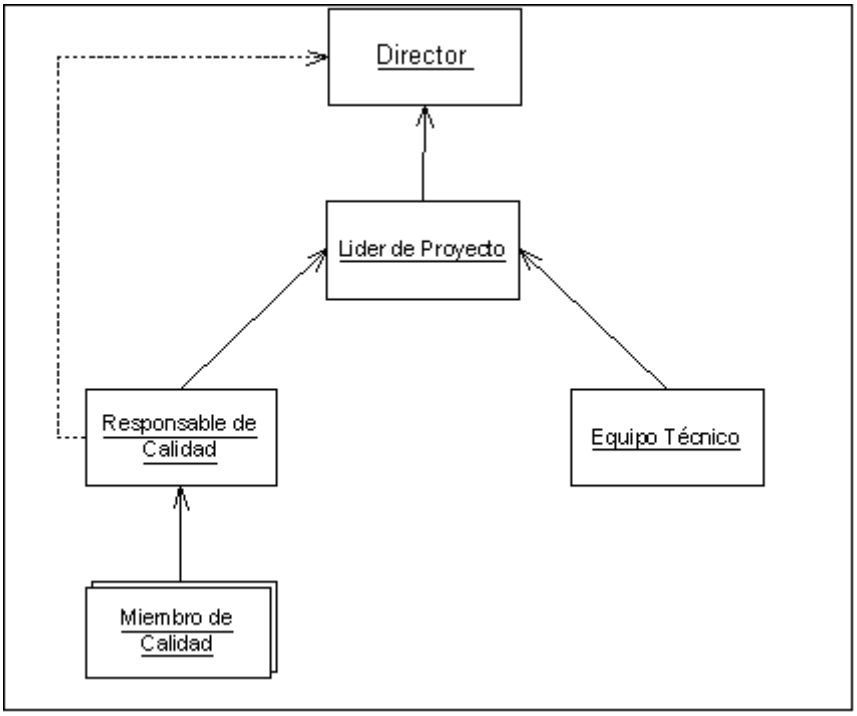


Figura 5.6 Equipo SQA

El Responsable de Calidad tiene un canal secundario de comunicación con un Director para resolver las no conformidades en el proyecto.

7. Responsabilidades

Responsable	Funciones
Responsable de Calidad	• Asegurar que los objetivos de calidad del proyecto se cumplan. • Planificar las actividades de SQA. • Obtener los recursos para la ejecución del Plan de SQA. • Realizar el monitoreo y el seguimiento de las actividades planificadas de SQA.

	• Informar a los Miembros del Equipo de proyecto y al Líder de Proyecto acerca de los resultados de las actividades de SQA. • Reportar periódicamente al Líder de Proyecto sobre las actividades de SQA. • Elevar los casos de no-conformidad en la calidad de los productos y en la ejecución de los procesos a la Dirección. • Proveer al Líder de Proyecto elementos para el análisis del origen de los problemas cuando se detectan diferencias entre los resultados esperados y reales de las mediciones de Calidad.
Miembro de Calidad	• Construir los Activos de Proceso que los Miembros del Equipo de Proyecto deben utilizar para construir los productos. • Ejecutar las tareas definidas dentro del Plan de SQA. • Realizar auditorias sobre los procesos. • Realizar revisiones a los productos para verificar su conformidad a los estándares. • Coordinar y facilitar revisiones.
Líder de Proyecto	• Definir junto con el Responsable de Calidad los objetivos de calidad del proyecto en el Plan de SQA. • Realizar ajustes en el proyecto para cumplir con el proceso definido, según las no-conformidades detectadas por el equipo de Calidad. • Compensar los desvíos detectados ó ajustar el Plan.
Director	• Proveer y garantizar los recursos para el desempeño de las funciones de Calidad. • Definir los casos de no-cumplimiento no resueltos en

	el ámbito del proyecto. • Realizar revisiones periódicas de las actividades y los resultados de SQA.
Equipo Técnico	• Realizar los ajustes necesarios y/ó rehacer el trabajo para cumplir con el producto definido.

8. Estándares

Estándares a utilizar son:

- Estándar de Especificación de Requerimientos
- Estándar de Especificación de Arquitectura
- Estándar de Especificación de Diseño Detallado
- Estándar de Modelado de datos, asignación de nombres y especificación del diccionario de datos.
- Estándar de Modelado de procesos y especificación de casos de uso y/o escenarios.
- Estándar de Nomenclatura para los nombres de componentes, módulos y piezas reusables.
- Estándar de Código Fuente: nombres de variables, estilo de programación.

9. Administración de la Subcontratación de Software

- Plan de Adquisición.
- Solicitud de Propuesta.
- Encuesta a Cliente del Proveedor.
- Encuesta a Proveedor Candidato.
- Checklist de Análisis de Riesgos.
- Evaluación de productos enlatados.
- Matriz de Evaluación.
- Informe de Evaluación Proveedor – Producto.
- Modelos de Contrato.

10. Checklist

- Checklist para Plan de Proyecto
- Checklist para Especificación de Requerimientos
- Checklist para Especificación de Diseño Detallado
- Checklist para Código Fuente

11. Control de Calidad

Se han definido los siguientes controles para el SQA:

- Plan de Proyecto
- Especificación de Requerimientos
- Especificación de Arquitectura
- Especificación de Diseño
- Código Fuente
- Plan de Implementación
- Plan de Migración
- Manuales del Usuario y Técnicos
- Casos de Prueba
- Resultados de Casos de Prueba

12. Historia de Cambios

Revisión	**Autor**	**Fecha**	**Descripción**

Plan de Pruebas

1. Introducción

Este documento se realiza para entregar la estructura del plan de pruebas de aceptación y la forma en que se diseñan, evalúan, ejecutan y reportan las pruebas de aceptación. En base a la fase de desarrollo definida mediante un modelo presentado en el plan general, se plantea

que inicialmente se realiza el Plan de Aceptación ya que para este planteamiento se consideran los requerimientos del usuario.

Las pruebas de aceptación se ejecutan luego de culminar fase de pruebas de sistema, esta ejecución se realiza en la última etapa de pruebas, la cual evaluará el sistema final con miras a su presentación frente al cliente. En esta prueba se verificará si el programa cumple con las especificaciones formales establecidas por el cliente. El aporte de las pruebas se considera como constructivo antes de destructivo, ya que detecta defectos que presenta el sistema pero no considera su corrección.

2. Metas y Objetivos

Metas

La meta esencial es la determinación por parte del cliente de la aceptación o rechazo del sistema desarrollado.

Objetivos

- Comprobar la correcta implementación operativa del entorno de usuario.
- Verificar funcionalidad del sistema completo.
- Verificar rendimiento.

Alcance

Este documento contempla las actividades a realizar durante el desarrollo del sistema, para las pruebas de aceptación.

Luego de finalizar las pruebas de sistema, el programa se encuentra completamente ensamblado, y se han encontrado y corregido los errores entre los módulos, métodos, clases y objetos. En este punto se comienza con la etapa de las pruebas de validación de

requerimientos más conocida como pruebas de aceptación. Éstas se enfocan en las acciones que realiza el usuario además de las salidas del sistema que puedan ser reconocidas por él; dichas acciones y salidas engloban las expectativas del usuario, y están definidas en las especificaciones de los requerimientos del software.

Las pruebas de aceptación, se realizan a los requerimientos funcionales, tomados de los diagramas de casos de uso y a los no-funcionales como facilidad de uso, recuperabilidad, eficiencia, entre otros; y se pretende lograr: corrección, vale decir, carencia de ambigüedad; completitud, es decir, especificación completa y clara del problema; y por último pero no menos importante, consistencia, quiere decir, que no haya requisitos contradictorios.

El plan que a continuación se detalla pretende dar una visión general sobre las actividades a realizar; sobre las pruebas consideradas; además de una explicación global que se consideró para la realización de los documentos a entregar, ya que darán una mayor información relacionada a la evaluación y reportes de este tipo de pruebas.

3. Definiciones acrónimos y abreviaciones

Proyecto	PROJECT
Artefacto no UML	NONUMLA
Requerimientos textuales	TEXTREQ
Casos de uso textuales	TEXTUCS
Casos de prueba	TEXTTCS
Historial de requerimientos	HISTREQ
Trazabilidad forward	TRACEFW
Trazabilidad backward	TRACEBW

Diagramas	DIAGRAM
Exportación a Word	EXPWORD
Exportación formato gráfico	EXPGRAF
Imprimir	PRINTING
Exportación a SQL	EXP2SQL
Ayuda	HELPMOD
Exportación a entidad-relación	EXPTOER
Interfaz base de datos	INTERDB
Interfaz diagramas y bases de datos	INTERDG
Base de datos	DATABAS
Calidad	QUALASS
Interfaz	INTERFA
Testing	TESTING

4. Metodología

Consideraciones para la elaboración del plan de prueba

Aspectos metodológicos

Las pruebas de aceptación deben contemplarse desde dos puntos de vista: el constructivo y el de gestión. Este proceso de descripción de las pruebas de aceptación constituye la parte constructiva de las pruebas de aceptación. Una vez que el sistema ha sido construido es necesario disponer de mecanismos que permitan seguir y comprobar que las pruebas se realizan de acuerdo con el plan general.

Elementos constructivos

Los criterios de aceptación: Son elementos de referencia usados en el proceso de prueba para determinar si la solución satisface los requerimientos de usuario. No se puede considerar resuelto un requerimiento hasta que no se haya definido la forma de probar su cumplimiento.

Escenarios de pruebas: Son una técnica descriptiva que permite diseñar los casos de prueba de una forma más expresiva que la puramente textual. Los escenarios tienen un marcado carácter de secuencialidad en el proceso de prueba y se pueden introducir los agentes que realizan físicamente las pruebas.

Atributos de gestión: Una vez que las pruebas han sido diseñadas, es necesario plantear tareas de prueba. Las tareas de prueba deben ser asignadas a ciertas personas y realizarse en ciertas fechas de acuerdo con un plan general de pruebas (en el que no solamente se contemplan las pruebas de aceptación).

Descripción del proceso de diseño de prueba

Para abordar la tarea de definición de los casos de prueba que se van a ejecutar para comprobar que las necesidades de los usuarios han sido satisfechas. Existen dos enfoques posibles:

- **Basado en los requerimientos de usuario:**

1º. Con la información del requerimiento y del criterio de aceptación se empieza a construir directamente el escenario de prueba.

2º. Una vez diseñados el/los escenario/s de prueba, éstos deben quedar automáticamente asociados al criterio de aceptación.

3º. Se permite la posibilidad de asociar el escenario de prueba al servicio que resuelva el requerimiento que estamos probando.

Este procedimiento de trabajo basado en los requerimientos como foco supone un doble objetivo:

1º. El objetivo prioritario es la construcción de un escenario de prueba de aceptación y de validación que se aplicará en su momento al sistema construido para comprobar que la necesidad del usuario ha sido satisfecha.

2º. Un objetivo secundario es la validación de la especificación que ya estaba elaborada, es decir, la constatación de que los servicios son adecuados para resolver los requerimientos de usuario.

- **Basado en las tareas (o servicios) que ofrece el sistema:**

En este caso se centra la atención en la especificación funcional del sistema. Los pasos a dar en este caso son:

1º. Se elige la tarea que resuelve el requerimiento y construir desde dicha tarea los escenarios de prueba. En este caso lo prioritario será tener en cuenta los aspectos funcionales que tiene la implementación del requerimiento.

2º. Cada vez que se crea un escenario de pruebas en cierta tarea es necesario preguntarse si hay que asociarlo a algún criterio de aceptación de alguno de los requerimientos que resuelve la tarea.

Gestión de escenarios de prueba

Una vez construido un escenario de pruebas y pasadas las fases de especificación, diseño, construcción y pruebas, será necesario pasar al sistema construido las pruebas de aceptación para que el usuario pueda dar su conformidad al sistema.

Estas pruebas (de caja negra) deben gestionarse: asignarles un responsable, una fecha, un estado, un entorno físico y un entorno lógico.

Cada escenario tendrá asociados una serie de atributos de gestión que se podrán visualizar y modificar dentro de la herramienta.

El proceso de gestión llevará asociadas una serie de acciones incluidas de carácter administrativas que deben ser descritas en un documento de gestión.

Documentación de la ejecución de las pruebas

Histórico de Pruebas: Documenta todos los hechos relevantes ocurridos durante la ejecución de las pruebas.

Informe de Incidente: Documenta cada incidente (p. ej., una interrupción en las pruebas debido a un corte de electricidad, bloqueo del teclado, etc.) ocurrido en la prueba y que requiera una posterior investigación.

Informe resumen: Resume los resultados de las actividades de prueba y aporta una evaluación del software, basada en dichos resultados.

5. Técnicas y Prácticas

El tipo de prueba a ejecutar en esta etapa son las pruebas funcionales, más conocidas como pruebas de caja negra. Las pruebas de caja negra permiten

detectar funcionamiento incorrecto o incompleto, errores de interfaz, errores accesos estructuras de datos externas, problemas de rendimiento, errores de inicio y terminación. Su criterio se basa en las interfaces y las especificaciones de los módulos.

Las prácticas de este tipo de prueba se inician cuando las pruebas de sistema han sido completadas exitosamente. Las pruebas se realizan basándose en los requerimientos establecidos por el usuario del sistema en forma clara, completa y consistente.

Las pruebas de caja negra que se realizan son las que a continuación se detallan:

Pruebas Alfa (entorno de desarrollo)

Pruebas desarrolladas por personas externas a los desarrolladores. Las pruebas alfa consisten en invitar al cliente a que venga al entorno de desarrollo del software a probar el sistema. Se trabaja en un entorno controlado y el cliente siempre tiene un experto a mano para ayudarle a usar el sistema y para analizar los resultados.

Pruebas Beta (entorno de cliente)

Pruebas realizadas por los usuarios finales. Las pruebas beta vienen después de las pruebas alfa, y se desarrollan en el entorno del cliente, un entorno que está fuera de control de los desarrolladores. Aquí el cliente se queda a solas

con el sistema y trata de encontrarle fallos al producto de los que informa por escrito al desarrollador.

En resumen las pruebas alfa y beta es habitual su realización (preparación, ejecución y documentación) para productos que serán vendidos. Algunos de los potenciales usuarios se prestan a estas pruebas bien por ir entrenando a su personal con tiempo o bien a cambio de alguna ventaja económica (mejor precio sobre el producto final, derecho a mantenimiento gratuito, a nuevas versiones, etc).

6. Actividades del Plan

Ciclo de vida de las pruebas de Aceptación

- Diseñar casos de prueba de aceptación basados en los requerimientos que presenta el cliente.
- Preparar datos de las pruebas de aceptación.
- Validar datos de los casos de prueba de aceptación.
- Procedimiento de la prueba.
- Ejecutar las pruebas de aceptación para validar el análisis de requerimientos del cliente.
- Comparar los resultados de las pruebas con los casos de prueba iniciales.
- Documentar las pruebas de aceptación.

Diseño de casos de prueba de aceptación

El primer paso es evaluar el sistema colocándose en la posición del cliente, verificando la funcionalidad y requerimientos, luego se revisa cuidadosamente la configuración que consiste en ver que todos los elementos se han desarrollado apropiadamente y están suficientemente detallados para soportar las fases de mantenimiento durante el ciclo de vida del software.

Para las pruebas Alfa descritas anteriormente se considera que los encargados de apoyar la labor del usuario mantengan un registro de los eventos que realiza el usuario durante la ejecución del sistema, mediante la siguiente tabla que detalla la información necesaria para la realización de la prueba.

El resultado indicará a los encargados de realizar los cambios el estado en que se encuentra el módulo con defectos, siendo del tipo Urgente, Controlada, Puede esperar, Vital. Finalmente las observaciones especifican de mejor forma para quienes tomen estos detalles para modificarlos, de como se produjo el defecto o fallo y ayudan rápidamente a apuntar al modulo defectuoso.

Tabla 5.1 Pruebas Alfa

Además, se solicitará al usuario que finalizado el testeo de prueba complete un formulario o cuestionario con preguntas relacionadas con su desempeño.

PRUEBAS ALFA				
Encargado :			Fecha :	
Módulo	**Sub-módulo**	**Entorno de prueba**	**Resultado Prueba**	**Observaciones**

Para las pruebas Beta descritas en el punto anterior se considera la participación del usuario frente al sistema, acá el ambiente de prueba no contará con una persona encargada de guiar al usuario, sino que su entorno de prueba será de forma privada y la ejecución la estimará el usuario en un ambiente propicio. La tarea que se sugiere realizar es que complete esta tabla a medida que vaya ejecutando el sistema y se encuentre con ciertas dificultades. Puede incluir además sugerencias en cuanto a la interacción y manipulación del sistema.

Tabla 5.2 Pruebas Beta

PRUEBAS BETA	
Usuario :	
Fecha:	
Prueba realizada	Observaciones

7. Preparación o creación de datos de las pruebas de aceptación

Las Listas de Chequeo están basadas en la identificación de las técnicas de prueba para evaluar cada sub-característica de las características de calidad. Una lista de chequeo es un formulario de preguntas, las cuales dependen del objetivo para el cual son usadas. Estas listas están clasificadas según las etapas del proceso de prueba. Para dar respuesta a cada pregunta se considera una escala del 1 al 5, en donde el uno (1) siempre es la respuesta menos significativa y cinco (5) la más significativa.

Tabla 5.3 Lista de Chequeo para la característica Fiabilidad

Cuestionario		
Característica	**Pregunta**	**Evaluación**
Madurez	1) ¿La información que actualmente entrega el sistema le parece suficiente?	1= Inaceptable 2= Bajo el promedio 3= Promedio 4= Bueno 5= Excelente
Correctitud	1.-¿ La manera de entregar la información por el sistema es la mas adecuada?	1= No 5= Si

9. Procedimiento de prueba

Las pruebas alfa se llevan a cabo con los usuarios en el ambiente de desarrollo, para ello el cliente comenta todo lo que considere como problemas del software y lo da a conocer al equipo de desarrollo. Luego de finalizada la prueba el participante de la prueba completa un formulario de preguntas, las consideraciones hechas serán la base para la evaluación de las pruebas.

La prueba beta se lleva a cabo con los usuarios en el sitio real donde será destinado el software como producto final; estos sitios se conocen como *beta sites*. En estas pruebas, normalmente el implementador o desarrollador no está presente, es decir, el software trabaja en un ambiente que no puede ser controlado por el equipo de desarrollo. El cliente registra todo lo que considere como problemas del software y lo reporta finalmente al equipo de desarrollo.

10. Criterio de aceptación

Para la realización de las pruebas se ha solicitado que en las áreas que consideran requerimientos funcionales como es el caso de los líderes, realicen un ranking de acuerdo a su criterio para cada caso de uso de sus respectivos equipos, con lo cual entregará la categorización y cantidad de pruebas a realizar para cumplir con las metas consideradas para garantizar calidad al producto. La prioridad ALTA, MEDIA o BAJA de los casos de uso asociados con los requerimientos aportarán para las pruebas a realizar ya que se enfocará en aquellos atributos con mayor importancia, estimando para ellos una mayor cantidad de pruebas sin dejar atrás los de prioridad baja.

Tabla 5.4. Evaluación de pruebas

Evaluación de pruebas	
Participantes:	**Fecha:**
Tipo prueba:	**Área /Equipo(a realizar la prueba):**
Porcentaje de satisfacción obtenido:	
Análisis de resultados:	

11. Criterio de evaluación

Para procesar los resultados **s**e establecen los pasos a seguir para obtener los resultados cuantitativos en base a las respuestas dadas en las listas de chequeo. Los resultados están determinados por las respuestas y por la importancia que los involucrados asignan a cada sub-característica. Luego, las sub-características deben ser ponderadas de acuerdo a la evaluación que los involucrados dan un peso (en una escala del 1 al 5) a cada una de ellas a través de la planilla de ponderación, en donde el uno (1) siempre es el peso menos significativo

y cinco (5) el más significativo. La tabla siguiente muestra la planilla de ponderación para la característica Fiabilidad.

Tabla 5.5 Planilla de Ponderación para la Característica Fiabilidad

12. Ejecución de las pruebas de aceptación

La ejecución de las pruebas alfa es realizada por el cliente junto a los desarrolladores, de manera que es el cliente el que utiliza el software en la forma más natural posible, y el implementador o desarrollador sólo observa y registra cualquier error o problema de uso, todo ello en un entorno controlado.

La ejecución de las pruebas beta es realizada por clientes llamados *beta sites,* por lo general son clientes dispuestos a correr los riesgos de aceptar un sistema que todavía no ha sido fielmente comprobado, a cambio de ventajas estratégicas para la empresa, como por ejemplo, un trato preferencial por parte del equipo de desarrollo, y quizás también la reducción de tiempo en el entrenamiento del personal que trabajaría regularmente con el sistema.

Característica de calidad: Fiabilidad					
Sub-característica	**Peso Asignado**				
	1	**2**	**3**	**4**	**5**
Madurez					
Tolerancia a Fallas					
Recuperabilidad					
Correctitud					
Estructurado					
Encapsulado					

13. Análisis de resultados

Los resultados pueden obtenerse en base a dos criterios: respuestas a las preguntas de las listas de chequeo y la ponderación.

- Dar respuesta a las preguntas de las listas de chequeo.

Las respuestas a las preguntas de las listas de chequeo se pueden dar de forma directa o mediante la realización de casos de prueba. De realizarse a través de los casos de prueba, las respuestas dependerán de los resultados obtenidos en los mismos.

Los casos de prueba serán evaluados por medio de la siguiente escala:

Tabla 5.6 Escala de Evaluación

Escala de Evaluación		
Aprobado		
No Aprobado	**Falla menor**	
	Falla Grave	

1= La sub-característica no está presente en esta etapa.

2= La sub-característica se presenta de manera muy deficiente.

3= La sub-característica se presenta medianamente.

4= La sub-característica se encuentra presente.

5= La sub-característica se encuentra altamente presente.

Tabla 5.7. Análisis de resultado general

Análisis de resultados	
Tipo de prueba:	Alfa
Resumen	

Cantidad de pruebas realizadas			**% pruebas satisfactorias**
Número de Pruebas	**Aceptadas**		
	Rechazadas		
	Modificadas		

14. Documento resumen de las pruebas de aceptación

Historial

El historial de las pruebas registra las pruebas realizadas y su estado de ejecución.

Tipo de prueba: Alfa, Beta.

Requerimiento asociado: Nombre del requerimiento.

Estado: Ejecutado, No ejecutado.

Tabla 5.8 Historial de pruebas

Historial de pruebas			
Fecha	**Tipo de prueba**	**Requerimiento asociado**	**Estado**

14.Plan de contingencia:

1. Si el usuario no asiste a la ejecución de la prueba por motivos justificados se debe cambiar la fecha propuesta, procurando una nueva reunión.

2. Si uno de los encargados de la ejecución de las pruebas no asiste, debe ser reemplazado por otra persona que se encuentre capacitada para la actividad.

15. Pruebas

Contenido de cada prueba

- Objetivo de la prueba
- Descripción de la prueba
- Técnica

Características a probar

Para este punto se consideró algunas de las características que se relacionan con los criterios de calidad y serán consideradas para realizar el formulario de pruebas que se entregará a los usuarios basándose en algunos de estos atributos que a continuación se detallan.

Características operativas

- Fiabilidad.
- Eficiencia.
- Eficacia
- Seguridad
- Facilidad de uso.

Capacidad de soportar los cambios

- Facilidad de mantenimiento.
- Flexibilidad.
- Facilidad de prueba.

Adaptabilidad a nuevos entornos

- Portabilidad.

- Reusabilidad.
- Interoperabilidad.

Otras consideraciones

- Rendimiento
- Capacidad

5.6 Planificar elaboración e implementación de procedimientos

La elaboración de un procedimiento se realiza de la siguiente manera:

Tenemos una plantilla ya desarrollada para el encabezado de todos los procesos

CARACTERIZACIÓN	**MACROPROCESO:** OPERACIONES DE LA BRIGADA Nº 23 CENEPA	**CÓDIGO:** GO.01 **VERSIÓN:** 1.0
	PROCESO: MANEJO DE PLANES Y ÓRDENES	**FECHA DE ELABORACIÓN:** 23/10/06 **FECHA DE ÚLTIMA REVISIÓN:** **PAGINA:**

Elaboración de un Procedimiento

A. OBJETIVO DEL PROCESO

B. ALCANCE DEL PROCESO

C. RESPONSABLE DEL PROCESO

D. REQUISITOS LEGALES DEL PROCESO

E. POLÍTICAS DEL PROCESO:

F. SUBPROCESOS

NOMBRE	PERIODICIDAD

Se colocara el nombre y la periodicidad con la que se utilice un subprocesos:

Nombre: Nombre del Subproceso

Periodicidad: Frecuencia con se ejecuta el subproceso

G. INDICADORES DEL PROCESO

NOMBRE	DESCRIPCIÓN	FORMULA DE CÁLCULO	UNID.	FRECUENCIA	FUENTE DE LA INFORMACIÓN

Se colocara los indicadores del proceso:

Nombre: Nombre del Proceso

Descripción: Indica el objetiva el objetivo del indicador

Formula de Calculo: Establece lo parámetros para calcular el valor del indicador

Unidad: Unidad de calculo de del indicador

Frecuencia: Frecuencia con la que se calcula el indicador

Fuente de la Información: Áreas, Secciones, Unidades

H. REGISTROS DEL PROCESO

NOMBRE	UBICACIÓN

Se colocaran los registros del proceso

Nombre: Nombre del Registro

Ubicación: Áreas, Secciones, Unidades o Directivos

I. DOCUMENTOS DEL PROCESO

CÓDIGO	NOMBRE

Se colocara los documentos inmersos en el proceso con un código.

Código: Código del documento

Nombre: Nombre del Documento

5.6.1 Definición de Procedimientos:

Procedimiento de Ingeniería

Requerimientos

1. Levantamiento de Información.
2. Especificación de requerimientos.
3. Validación de requerimientos.

Diseño

1. Describir los objetivos del sistema.
2. Identificar los subsistemas.
3. Identificar usuarios del sistema.
4. Identificar entidades externas del sistema.
5. Elaborar de los Diagramas UML.
6. Análisis de requerimientos funcionales.
7. Modelo de procesos.
8. Especificar requerimientos funcionales.
9. Requerimientos de interfaces externas.
10. Requerimientos de rendimiento.
11. Restricciones de diseño.
12. Atributos
13. Estimar costos y tiempo de desarrollo.
14. Métricas de análisis

Programación

1. Verificar la especificación de requerimientos de software
2. Determinar el lenguaje de programación
3. Determinar el motor de base de datos
4. Determinar los estándares de programación

5. Desarrollar el sistema
6. Implantar el sistema
7. Soporte y mantenimiento del sistema

Pruebas

1. Se realiza las pruebas de caja negra para verificar las fallas que arroja el software
2. Corridas de pruebas del software para determinar lugar de la falla
3. validación y verificación del software

Procedimiento de Soporte Técnico

1. Sugerir el procedimiento apropiado, características del lenguaje, o productos para un particular tipo de análisis o funcionalidad requerida.
2. Responder preguntas acerca de detalles específicos de los procedimientos, tales como discusión de las características especiales disponibles, opciones y limitaciones.
3. Suministrar referencias para formulas y técnicas estadísticas utilizadas por nuestros algoritmos donde sea posible.
4. Suministrar a los clientes una guía y referencias limitadas para la interpretación de las salidas producidas por los procedimientos estadísticos.
5. Aislar, documentar encontrar correcciones para los defectos reportados por el software.
6. Trabajar con el equipo de desarrollo de software para suministrar actualizaciones apropiadas para nuestro software cuando lo consideremos apropiado.
7. Suministrar advertencias estadísticas limitadas y generales como se considere necesario en las bases de un caso específico, sin embargo, es la responsabilidad del cliente determinar que tipo de análisis estadístico es apropiado para sus necesidades.
8. Dar soporte a las inquietudes del cliente con documentación impresa o documentos online, suministrando ejemplos o explicaciones adicionales para el esclarecimiento de conceptos requeridos.
9. Proveer ayuda durante la instalación del software.
10. Proveer pequeñas asistencia con la lógica de programación.

11. Proveer una descripción amplia de ajustes del hardware, modificaciones del sistema operativo.

Procedimiento de Aseguramiento de la calidad

DESARROLLO

1. Procedimiento para el análisis de la situación actual.

Para el análisis la situación actual se propone el procedimiento representado en el siguiente esquema:

1.1. Análisis externo

La guía para el análisis externo debe tener en cuenta los siguientes factores:

- Culturales
- Tecnológicos
- Políticos
- Económicos
- Legales
- Características del mercado
- Demanda actual y proyectada
- Clientes actuales y clientes potenciales
- Determinantes estructurales de la competencia. Fuerzas de Porter.

  - Amenazas de nuevos ingresos. Barreras para el ingreso

  - Intensidad de la rivalidad entre los competidores existentes

  - Poder de negociación de los proveedores

  - Presión de productos sustitutos

- Poder de negociación de los clientes

Los resultados de su aplicación se clasifican en amenazas y oportunidades.

1.2. Confección de la matriz FODA

Para la confección de la matriz FODA siguen los siguientes pasos:

1. Listado de las fortalezas, debilidades, oportunidades y amenazas.
2. Reducción del listado hasta un número que permita trabajar con comodidad la matriz.
3. Elaboración de la matriz de interacción.

2. Procedimiento para la implantación del sistema de la calidad

Para llevar a cabo la estrategia seleccionada se proponen los siguientes pasos:

ETAPA 1. Implantación de un proceso de cambio organizacional encaminado a crear cultura de calidad.

- Identificación de la necesidad del cambio y diagnóstico de la situación actual.
- Planificación del cambio.
- Ejecución de las acciones y obtención de resultados.

ETAPA 2. Creación del grupo gestor de calidad y procesamiento de los resultados del diagnóstico interno.

- Creación del grupo gestor de calidad con personal de todas las áreas de la organización.
- Procesamiento de los resultados del diagnóstico interno
- Confección del diagrama de Pareto para estimar el grado de cumplimiento de cada requisito.

ETAPA 3. Diseño del sistema de la calidad de acuerdo con la estrategia seleccionada.

- Identificación de los procesos estratégicos, operativos y de apoyo y sus interrelaciones.
- Confección de los mapas de proceso y flujogramas.
- Elaboración de los procedimientos generales del Sistema de la Calidad.
- Elaboración del Manual de Calidad
- Elaboración de los documentos específicos (procedimientos específicos de trabajo, especificaciones, expedientes maestros de los productos)

ETAPA 4. Implantación del Sistema de la Calidad

- Puesta en práctica de lo establecido en los documentos.
- Recopilación de la evidencia documentada de lo anterior

ETAPA 5. Ejecución de auditorias internas y revisión del Sistema de la Calidad

- Ejecución de auditorias internas
- Ejecución de acciones correctoras y preventivas
- Revisión del sistema por la Dirección

ETAPA 6. Certificación.

- Solicitud de la auditoria
- Ejecución de la auditoria
- Certificación de acuerdo con los resultados de la auditoria

Procedimiento de Administración de Proyectos

1. Determinar un conjunto de necesidades para establecer un proyecto.
2. De acuerdo a las necesidades, determinar el proyecto que se va a realizar.
3. Determinar los requerimientos del proyecto.
4. Determinar el equipo de trabajo.
5. Establecer un cronograma de acción.

6. Desarrollo del proyecto.
7. Implementación del proyecto.
8. Seguimiento y control.

5.7 Definir Métricas de Software

Métricas de Estimación

La planificación de un proyecto se basa en una buena estimación del esfuerzo requerido para realizarlo, y para apoyar esta difícil tarea, se han desarrollado varios métodos que han encontrado aceptación comercial en forma creciente en la planificación del desarrollo de software.

La mayoría de estos métodos incluyen modelos empíricos de estimación y poseen como variable manejadora del costo principal, el tamaño de la aplicación a desarrollar, lo que es suficientemente difícil de estimar como para que se justifique pensar en automatizar o apoyar fuertemente esta tarea con la generación de un método fácil de usar.

Para el caso de los modelos basados en líneas de código, se puede observar que en la actualidad, las herramientas de desarrollo proveen la capacidad de disminuir substancialmente el esfuerzo de codificación, pues la tendencia actual ya no es codificar, sino generar código. Por el lado de las técnicas basadas en el enfoque de puntos de función, el problema radica en que la estimación sólo puede realizarse con un diseño externo acabado de la aplicación, y si consideramos la utilización de herramientas de generación de código.

Líneas de código

La métrica de tamaño tradicional para estimar el esfuerzo de desarrollo y productividad ha sido LDC (Líneas de código) o SLDC (Líneas de código fuente). Se han propuesto varios modelos

de estimación, la mayoría de ellos son funciones de las líneas de código o de las miles de líneas de código que tendrá el software a desarrollar. Generalmente, el modelo de estimación de esfuerzo consiste de dos partes.

La primera provee una base de estimación como una función del tamaño del software, y es de la siguiente forma:

$$E= A+B * (KLDC)^{c}$$

donde E es el esfuerzo estimado en meses hombre, A, B y C son constantes y KLDC es el tamaño estimado del sistema final en miles de líneas de código.

La segunda parte del modelo modifica esta estimación en base a cuantificar la influencia de factores de ambiente, por ejemplo la utilización de diferentes metodologías, habilidad del equipo de desarrollo y restricciones de hardware.

Se debe procurar estimar la cantidad de LDC teniendo en cuenta el tamaño que tendrá la aplicación en su etapa inicial.

Puntos de Función

El análisis por puntos de función es un método para cuantificar el tamaño y la complejidad de un sistema software en términos de las funciones de usuario que este desarrolla (o desarrollará). Esto hace que la medida sea independiente del lenguaje o herramienta utilizada en el desarrollo del proyecto.

Para desarrollar la métrica de puntos de función se debe tomar en cuenta

1. Números de entrada de usuario: se cuenta cada entrada del usuario que proporcione al software diferentes datos orientados a la aplicación. Las entradas deben ser distinguidas de las peticiones que se contabilizan por separado.

2. Número de salida del usuario: se encuentra cada salida que proporciona la usuario información orientada ala aplicación. En este contexto las salidas se refieren a informes,

pantalla, mensajes de error. Los elementos de datos individuales dentro de un informe se encuentran por separado.

3. Números de peticiones al usuario: una petición esta definida como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida interactiva. Se cuenta cada petición por separado.

4. Número de archivos: se cuenta cada archivo maestro lógico, o sea una agrupación lógica de datos que puede ser una parte en una gran base de datos o un archivo independiente.

5. Número de interfaces externas: se cuentan todas las interfaces legibles por la maquina por ejemplo: archivos de datos, en cinta o discos que son utilizados para transmitir información a otro sistema.

Factores de complejidad:

- Son 14 factores que completan la visión externa de la aplicación.
- Toman un valor entre 0 y 5.

1) Comunicación de Datos

2) Proceso Distribuido

3) Objetivos de rendimiento

4) Configuración de Explotación usada por otros sistemas

5) Tasa de transacciones

6) Entrada de datos en línea

7) Eficiencia con el usuario final

8) Actualizaciones en línea

9) Lógica de Proceso Interno Compleja

10) Reutilización del Código

11) Contempla la conversión e instalación

12) Facilidad de operación

13) Instalaciones múltiples

14) Facilidad de cambios

Cuando han sido recogidos los datos anteriores se asocian el valor de complejidad a cada cuenta. Las organizaciones que utilizan métodos de puntos de función desarrollan criterios para determinar si una entrada es denominada simple, media o compleja. No obstante la determinación de la complejidad es algo subjetivo.

Para calcular los puntos de función se utiliza la siguiente relación.

PF = CUENTA_TOTAL * [0.65 + 0.01 * SUM(fi)]

Donde CUENTA_TOTAL es la suma de todas las entradas de PF obtenidas de la tabla anterior.

Fi donde i puede ser de uno hasta 14 los valores de ajuste de complejidad basados en las respuestas a las cuestiones señaladas de la siguiente tabla.

Evaluar cada factor en escala 0 a 5.

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

Tabla auxiliar de los puntos de Función

Parámetro	Simple	Medio	Complejo
Nº Entradas de Usuario	3	4	6
Nº de Salidas de Usuario	4	5	7
Nº de Peticiones al usuario	3	4	6
Nº de Archivos	7	10	15
Nº de Interfaces externos	5	7	10

Los valores constantes de la ecuación anterior y los factores de peso aplicados en las encuestas de los ámbitos de información han sido determinados empíricamente.

Una vez calculado los puntos de función se usan de forma analógica a las LDC como medida de la productividad, calidad y otros productos del software.

Productividad = PF / persona-mes

Calidad = Errores / PF

Costo = Dólares / PF

Documentación = Pags. Doc / PF

COCOMO

- Estima meses/hombre de esfuerzo de desarrollo en término de miles de líneas de código fuente liberadas.

Su formula es :

$$ESFUERZO = C1\ FA(Tamaño)^{P1}$$

Esfuerzo= numero de meses / hombre

C1: Factor de escalamiento del esfuerzo

FA: Factor de Ajuste

Tamaño: Tamaño del producto en líneas de código

P1: Efectividad del proceso

- Estima el tiempo de desarrollo (calendario)

Su formula es :

$$TIEMPO = C2\ (Esfuerzo)^{P2}$$

Tiempo: Número total de meses

C2: Factor de escalamiento calendario

Esfuerzo: Número de meses/hombre

P2: Inercia y paralelismo de la gerencia

Puntos de Característica.

Debido a que el análisis por Puntos de Función fue diseñado para software de negocios y no es fácil de generalizar a aplicaciones científicas, de tiempo real y otras, Caper Jones propuso ampliaciones a este método, generando una métrica que denominó Puntos de Característica. Ésta da cabida a aplicaciones cuya complejidad algorítmica es alta.

Este método considera los mismos elementos que considera Albrecht en su análisis por puntos de función, sólo que añade la variable "número de algoritmos" y elimina los niveles de complejidad, así, cada cuenta es pesada por un valor único para ese componente (es decir, se le asigna complejidad media).

Métrica para Estimación Temprana del Tamaño del Software.

Para poder realizar una estimación temprana, se propone la utilización de la métrica de puntos de función modificada.

Este método consiste en clasificar, listar y contar a partir de la especificación de requisitos todos los componentes, desagregados en:

a. Entradas Externas.

b. Salidas Externas

c. Archivos Lógicos

d. Consultas.

De este modo, la fórmula para obtener los puntos de función para estimación temprana (PFET) es la siguiente:

$$PFET = (4 \times IN + 5 \times OUT + 4 \times INQ + 10 \times FILE) \times ACP$$

Donde

PFET : Puntos de función para estimación temprana.

IN : Número de Entradas Externas

OUT : Número de Salidas Externas

INQ : Número de Consultas

FILE : Número de Archivos Lógicos

ACP : Factor de Ajuste de Complejidad de Proceso

Una forma de clasificarlo el tamaño es atendiendo al número de líneas de código, como se muestra en la tabla:

Tabla 5.9: Categoría de un proyecto en función de sus líneas de código

Categoría	*Programadores*	*Duración*	*Líneas de código*	*Ejemplo*
Trivial	1	0 – 4 semanas	< 1k	Utilidad de ordenación
Pequeño	1	1 – 6 meses	1k – 3k	Biblioteca de funciones
Media	2 – 5	0,5 – 2 años	3k – 50k	Compilador de C

Grande	5 – 20	2 – 3 años	50k – 100k	SO pequeño
Muy grande	100 – 1000	4 – 5 años	100k – 1M	Grandes SO
Gigante	1000-5000	5 – 10 años	> 1M	Sistema de Distribución

Método de Regresión Lineal.

Se han de registrar datos de Esfuerzo y Tamaño de proyectos concluidos. En base a estos registros se obtendrá un modelo de la forma

Esfuerzo = f (tamaño del software)

donde el esfuerzo es medido en horas hombre, y el tamaño del software está medido en puntos de función para estimación temprana.

Para fines estadísticos, hay que notar que la variable predictora o independiente es el tamaño del software, y la variable respuesta o dependiente es el esfuerzo de desarrollo.

Así, el esfuerzo *E* es función del tamaño *T*, de la forma

$$E = a + b \times T$$

Esta ecuación corresponde a un modelo de regresión lineal simple, donde los parámetros a y b quedan determinados por:

$$b = \frac{n \times \sum_{i=1,n} E_i T_i - \sum_{i=1,n} E_i \times \sum_{i=1,n} T_i}{n \times \sum_{i=1,n} (T_i)^2 - (\sum_{i=1,n} T_i)^2}$$

$$a = \frac{\sum_{i=1,n} E}{n} - \frac{\sum_{i=1,n} T_i}{n} \times b$$

Métricas de Calidad

Métrica de Tiempo de Respuesta en Servicios de Asesoría.

$$TRS = TRNE / TCE$$

donde:

TRNE = Total de respuesta no efectuadas.

TCE = Total de consultas efectuadas.

Métricas de Funcionamiento.

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 = número de errores encontrados antes de la entrega del software al usuario final.

D = número de errores encontrados después de la entrega al usuario final.

Métricas de funcionalidad

- **Métricas de Adecuidad**

Nombre:	Completitud de implementación funcional
Propósito:	Qué tan completa está la implementación funcional.
Método de aplicación:	Contar las funciones faltantes detectadas en la evaluación y comparar con el número de funciones descritas en la especificación de requisitos.
Medición, fórmula:	X = 1 - A/B A = número de funciones faltantes B = número de funciones descritas en la especificación de requisitos
Interpretación:	0 <= X <= 1 Entre más cercano a 1, más completa.
Fuente de medición:	Especificación de requisitos Diseño Código fuente Informe de revisión

Métricas de Fiabilidad

- **Métrica de Madurez**

Nombre:	Suficiencia de las pruebas
Propósito:	Cuántas de los casos de prueba necesarios están cubiertos por el plan de pruebas.
Método de aplicación:	Contar las pruebas planeadas y comparar con el número de pruebas requeridas para obtener una cobertura adecuada.
Medición, fórmula:	X = A/B A = número de casos de prueba en el plan B = número de casos de prueba requeridos
Interpretación:	0 <= X Entre X se mayor, mejor la suficiencia.
Fuente de medición:	A proviene del plan de pruebas B proviene de la especificación de requisitos

Métricas de Usabilidad

- **Métricas de Entendibilidad**

Nombre:	Funciones evidentes
Propósito:	Qué proporción de las funciones del sistemas son evidentes al usuario.
Método de aplicación:	Contar las funciones evidentes al usuario y comparar con el número total de funciones.
Medición, fórmula:	X = A/B A = número de funciones (o tipos de funciones) evidentes al usuario B = total de funciones (o tipos de funciones)
Interpretación:	0 <= X <= 1 Entre más cercano a 1, mejor.
Fuente de medición:	Especificación de requisitos Diseño Informe de revisión

Métricas Eficiencia

- **Métricas de comportamiento en el tiempo**

Nombre:	Tiempo de respuesta
Propósito:	Cuál es el tiempo estimado para completar una tarea.
Método de aplicación:	Evaluar la eficiencia de las llamadas al SO y a la aplicación. Estimar el tiempo de respuesta basado en ello. Puede medirse: • Todo o partes de las especificaciones de diseño. • Probar la ruta completa de una transacción. • Probar módulos o partes completas del producto. • Producto completo durante la fase de pruebas.
Medición, fórmula:	X = tiempo (calculado o simulado)
Interpretación:	Entre más corto, mejor.
Fuente de medición:	Sistema operativo conocido Tiempo estimado en llamadas al sistema

Métricas de Mantenibilidad

- **Métricas de Cambiabilidad**

- **5.2. Ejemplo de Métrica de Cambiabilidad**

Nombre:	Registrabilidad de cambios
Propósito:	¿Se registran adecuadamente los cambios a la especificación y a los módulos con comentarios en el código?
Método de aplicación:	Registrar la proporción de información sobre cambios a los módulos
Medición, fórmula:	X = A/B A = número de cambios a funciones o módulos que tienen comentarios confirmados B = total de funciones o módulos modificados
Interpretación:	0 <= X <= 1 Entre más cercano a 1, más registrable. 0 indica un control de cambios deficiente o pocos cambios y alta estabilidad.
Fuente de medición:	Sistema de control de configuraciones Bitácora de versiones Especificaciones

Métricas de Transportabilidad

- **Métricas de conformidad de la transportabilidad**

Nombre:	Conformidad de transportabilidad
Propósito:	Qué tan conforme es la transportabilidad del producto con regulaciones, estándares y convenciones aplicables.
Método de aplicación:	Contar los artículos encontrados que requieren conformidad y comparar con el número de artículos en la especificación que requieren conformidad.
Medición, fórmula:	X = A/B A = número de artículos implementados de conformidad B = total de artículos que requieren conformidad
Interpretación:	0 <= X <= 1 Entre más cercano a 1, más completa.
Fuente de medición:	Especificación de conformidad y estándares, convenciones y regulaciones relacionados. Diseño Código fuente Informe de revisión

Métricas de calidad y productividad

Métricas de Calidad: proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. Es decir cómo voy a medir para que mi sistema se adapte a los requisitos que me pide el cliente.

Métricas de Productividad: Se centran en el rendimiento del proceso de la ingeniería del software. Es decir que tan productivo va a ser el software que voy a diseñar.

En los rendimientos del sistema y los datos se puede desarrollar, para cada proyecto un conjunto de métricas sencillas de productividad y calidad orientadas al tamaño. Se obtienen las siguientes formulas:

Productividad = KLDC/persona-mes

Calidad = errores/KLDC

Documentación = pags. Doc/ KLDC

Costo = $/KLDC

Métricas de Control y Seguimiento

Métricas de Control

- Desvíos de tamaño.
- Desvíos de esfuerzo.
- Desvíos de costo.
- Desvíos en el camino crítico.
- Desvíos en las actividades de SCM planificadas.
- Desvíos en las actividades de SQA planificadas.

Métricas de Seguimiento

- Horas reales insumidas en cada tarea vs horas estimadas.
- Porcentaje de desvío de esfuerzo registrado mensualmente.
- Porcentaje de desvío de cronograma registrado mensualmente.
- Porcentaje de desvío de costo registrado mensualmente.

- Capacitaciones realizadas en el mes.
- Magnitud final del proyecto.
- Tasa de errores del proyecto.
- Cantidad final de requerimientos de cambio.
- Cantidad final de casos del proyecto.

CAPÍTULO 6

CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

- La adopción de CMMI como metodología basada en procesos de calidad, orienta a la estandarización dentro de la organización en todos sus niveles y áreas.
- La calidad del producto depende directamente de la calidad del proceso.
- El mejoramiento y estandarización de los procesos es un camino seguro a alcanzar el nivel de calidad deseado
- El CMMI no puede ser implementado por una única persona o un equipo. Requiere el compromiso y dedicación de todas las partes y niveles de la empresa.
- Es necesario analizar cual es el nivel de la empresa dentro del modelo CMMI para saber cual será el siguiente nivel a alcanzar y los pasos a seguir
- El segundo nivel del CMMI es un nivel del modelo orientado a contar con procesos estandarizados, para realizar la definición y el diseño de la metodología.
- La fase de definición y diseño de la Metodología CMMI implica la elaboración de: una arquitectura de procesos, políticas organizacionales, implementación de procedimientos y definición de métricas.
- Cada fase de la metodología es un proyecto a largo plazo, donde se debe estandarizar procesos.
- Al ser una tesis de definición y diseño de la metodología CMMI la implementación queda a cargo de la Fuerza Terrestre.

6.2 Recomendaciones

- Es necesario definir el marco para el mejoramiento de procesos dentro de la organización.
- Es recomendable desarrollar planes de calidad y planes de prueba para el aseguramiento de la calidad del producto y la satisfacción del cliente.

- Se debería contemplar que el líder del proyecto se mantenga, para evitar el estancamiento del mismo, ya que todas las fases son construidas a largo plazo.
- Se recomienda establecer una capacitación de CMMI.
- La organización debe proveer los recursos necesarios en tecnologías, y personas capacitadas.
- Se recomienda actualizar tanto los procesos como los procedimientos a fin de obtener mayor calidad en los mismos.
- En la implementación la metodología se la debe aplicar de acuerdo al proyecto a realizar.