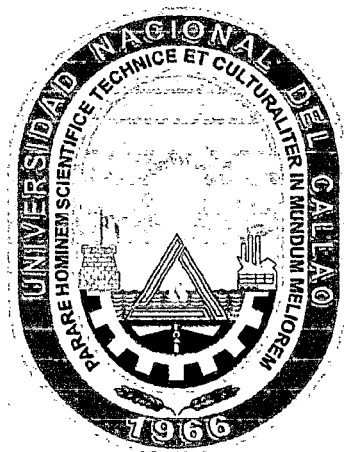


UNIVERSIDAD NACIONAL DEL CALLAO

FACULTAD DE INGENIERIA INDUSTRIAL Y DE SISTEMAS

ESCUELA PROFESIONAL DE INGENIERIA DE SISTEMAS



**"SISTEMA DE GESTIÓN DE CONTROL DE ASIGNACIÓN DE
BIENES INFORMÁTICOS PARA UNA EMPRESA"**

TESIS

PARA OPTAR EL TÍTULO PROFESIONAL DE :
INGENIERO DE SISTEMAS

PRESENTADO POR:
OSCAR EUGENIO VALERIANO LOAYZA

ASESOR:
MG. JUAN RAMIREZ VELIZ

CALLAO - PERÚ
2009

Mi tesis la dedico:

A mis padres que me dieron la
vida y han estado conmigo en
todo momento.

Oscar VALERIANO

AGRADECIMIENTOS

Esta tesis no hubiese sido posible su finalización sin la cooperación desinteresada de todas y cada una de las personas que a continuación citaré y muchas de las cuales han sido un soporte muy fuerte en momentos difíciles.

Primero y antes que nada, dar gracias a **Dios**, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

A mis Padres, por creer y confiar siempre en mí, apoyándome en todas las decisiones que he tomado en la vida.

A mis maestros, por sus consejos y por compartir desinteresadamente sus amplios conocimientos y experiencia.

INDICE

INTRODUCCION.....	1
CAPITULO I: PLANTEAMIENTO DEL PROBLEMA.....	4
1.1. Definición del problema.....	4
1.2. Formulación del problema.....	5
1.3. Objetivo.....	6
1.3.1. Objetivo general.....	6
1.3.2. Objetivos específicos.....	6
1.4. Delimitación del problema.....	6
1.5. Justificaciones de la investigación.....	7
1.6. Hipótesis.....	7
1.6.1. Hipótesis general.....	7
1.6.2. Hipótesis específicas.....	8
CAPITULO II: MARCO TEÓRICO.....	9
2.1 Dirección de telemática.....	9
2.1.1. Reseña histórica.....	9
2.1.2. Misión.....	10
2.1.3. Visión.....	10
2.1.4. Organigrama.....	11
2.2 Bases Teóricas.....	12
2.2.1. Lenguaje de Modelamiento Unificado.....	12
2.2.2. Objetivos de UML.....	13

2.2.3. Orientación a objetos	14
2.2.4. Fases de desarrollo	15
2.2.5. Artefactos para el desarrollo de proyectos	17
2.2.6. El proceso de desarrollo	19
2.2.7. Proceso Unificado de desarrollo RUP	20
2.2.8. Arquitectura Tecnológica	53
2.2.9. FODA	55
CAPITULO III: MATERIALES Y METODOS	58
3.1. Materiales	58
3.1.1. Hardware	58
3.1.2. Software	60
3.1.3. Otros	61
3.2. Métodos	62
3.2.1. Lugar de ejecución del proyecto	62
3.2.2. Cobertura de estudio	62
3.2.3. Metodología de desarrollo	63
CAPITULO IV: RESULTADOS	64
4.1. Situación actual del sistema	64
4.2. Estudio de riesgos y posibles soluciones	64
4.3. Modelo actual	68
4.3.1. Análisis del modelo de contexto	70
4.3.2. Análisis del proceso	92

4.3.3. Análisis normativo y regulatorio.....	93
4.3.4. Modelo Entidad-Relación	93
4.4. Identificación de mejoras (aplicación de la metodología RUP)	94
4.4.1. Mejora de procesos	95
4.4.2. Reestructuración organizacional	96
4.4.3. Priorización de las mejoras	96
4.5. Resultados	97
CAPITULO V: ANÁLISIS COMPARATIVO	98
5.1. Reducción de tiempos	98
5.2. Reducción de documentos.....	99
5.3. Análisis de costo beneficio.....	99
5.3.1. Costos	99
5.3.2. Ahorro	100
CAPITULO VI: CONCLUSIONES Y RECOMENDACIONES.....	101
6.1. Conclusiones.....	101
6.2. Recomendaciones.....	102
CAPITULO VII: BIBLIOGRAFÍA.....	103
CAPITULO VIII: APENDICES.....	106
CAPITULO IX: ANEXOS.....	113

RESUMEN

El presente trabajo de investigación tuvo como propósito fundamental la gestión, control y asignación de bienes informáticos, obteniéndose como resultados importantes la reducción de tiempos en proceso de asignación. A continuación se detallan los resultados:

Reporte	Sin la aplicación	Con la Aplicación
Consolidado Anual del Inventario de Software	6 Meses	1 día
Actualización de Inventario	3 Meses	2 días
Reportes al Alto Mando Naval	1 mes	1 día
Reportes a los Comités de Adquisiciones del Licenciamiento de Software	1 Mes	2 días

Por otro lado se ha obtenido una simplificación de documentos, reducción de costos y optimización de los tiempos de asignación de inventario.

Con el nuevo Sistema se podrán efectuar un debido control de registro y emisión de reporte de inventario de software siendo efectuado en el menor tiempo posible a requerimiento del alto mando naval y los Comités de Licenciamiento para que puedan tener una información precisa y oportuna a fin de proyectar los nuevos licenciamientos, minimizando gastos excesivos de cualquier tipo, permitiendo estar acorde con la tecnología actual en la Institución.

INTRODUCCION

Una adecuada implementación de la tecnología de información, permite la maximización de ingresos, reducción de costos, administración de inversiones, manejo de relaciones con clientes y proveedores, administración del personal y oportunidad de administración integrada. Asimismo la automatización permite suplantar procesos basados en papeles y acceso a información en tiempo real y con tecnología móvil, diseñar procesos estandarizados a seguir, pero al mismo tiempo ser tan flexible como las necesidades del negocio lo demanden.

PLANTEAMIENTO DE PROBLEMA

¿En qué medida afecta el Sistema de Gestión de control y el componente tecnológico de la organización en asignación de los recursos informáticos de la Marina de Guerra del Perú?

PROBLEMAS SECUNDARIOS

1. ¿Cómo facilitar la toma de decisiones de negocios en el subsistema servicios de Internet en la empresa?
2. ¿Cómo influye la disminución del precio en el incremento del número clientes en nuestra empresa?
3. ¿Cómo influye las promociones en el incremento del número clientes para la empresa en el subsistema servicios de internet?

4. ¿Cómo influye el crédito en el incremento del número de clientes para la empresa?

Por otro lado se generaron los siguientes objetivos.

OBJETIVO GENERAL

Determinar si el sistema de gestión de control y el componente tecnológico afecta la asignación de bienes informáticos "MARINA DE GUERRA DEL PERU"

OBJETIVOS ESPECÍFICOS

- Tener un control detallado de los distintos artículos informáticos que se asignan, a las diversas áreas que componen las Fuerzas Armadas.
- Mantener un control sobre el ingreso de nuevo personal, así como de nuevos artículos informáticos en la institución
- Mantener un registro histórico de las asignaciones de artículos informáticos realizadas en la institución.

DELIMITACION DEL PROBLEMA

El problema se encuentra delimitado dentro del sector Militar de la Marina de Guerra del Perú, específicamente en recursos informáticos

JUSTIFICACIONES DE LA INVESTIGACION

El desconocimiento de los bienes informáticos en la Marina de Guerra del Perú, ha traído como consecuencia, la falta de integración de los datos de identificación física y de antecedentes jurídicos y administrativos de los bienes informáticos, para tener un registro adecuado de los activos de esta institución pública.

Con la instalación y puesta en marcha del proyecto de registro y asignación de los artículos informáticos de las Fuerzas Armadas, que sirve para conocer las propiedades con que cuenta dicha institución pública, se podrá utilizar el patrimonio público de una manera ordenada y eficiente, haciendo uso de las tecnologías de Información e implementando este software llamado *Sistemas de Control de Asignación de Artículos Informáticos Marina de Guerra del Perú*

EL AUTOR

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

1.1. DEFINICIÓN DEL PROBLEMA

La Marina de Guerra del Perú es un organismo público, con personería jurídica de derecho privado, propiedad absoluta del estado, autonomía administrativa y económica.

En la actualidad se vive en un mundo globalizado y altamente competitivo, siendo el permanente deseo y necesidad de las empresas, ganar mercados y a la vez obtener rentabilidad en sus negocios. En tal sentido, las empresas se encuentran a la búsqueda de una mayor competitividad, la cual se podrá conseguir a través de obtener rentabilidad en los negocios y reducir sus costos a través de nuevas tecnologías. Esto ha inducido a que el avance de la ciencia en estos últimos años sea mayor a lo progresado en el pasado. La introducción de nacientes tecnologías en las actividades productivas ha generado la aparición de problemas “nuevos” , tal vez en algunas ocasiones no identificadas inmediatamente sino con el tiempo.

En este aspecto confluye la ejecución del proyecto de sistema de gestión y control de asignación de recursos informáticos.

1.2. FORMULACIÓN DE PROBLEMA

¿En que medida afecta el Sistema de Gestión de control y el componente tecnológico de la organización en asignación de los recursos informáticos de la Marina de Guerra del Perú?

Problemas secundarios

1. ¿Cómo facilitar la toma de decisiones de negocios en el subsistema servicios de Internet en la empresa ?
2. ¿Cómo influye la disminución del precio en el incremento del número clientes en nuestra empresa?
3. ¿Cómo influye las promociones en el incremento del número clientes para la empresa en el subsistema servicios de internet?
4. ¿Cómo influye el crédito en el incremento del número clientes para la empresa?

1.3. OBJETIVO

1.3.1. Objetivo general

Determinar si el sistema gestión de control y el componente tecnológico afecta la asignación de bienes informáticos "MARINA DE GUERRA DEL PERU"

1.3.2. Objetivos específicos

- Tener un control detallado de los distintos artículos informáticos que se asignan, a las diversas áreas que componen las Fuerzas Armadas.
- Mantener un control sobre el ingreso de nuevo personal, así como de nuevos artículos informáticos en la institución
- Mantener un registro histórico de las asignaciones de artículos informáticos realizadas en la institución.

1.4. DELIMITACION DEL PROBLEMA

El problema se encuentra delimitado dentro del sector Militar de la Marina de Guerra del Perú, específicamente en recursos informáticos

1.5. JUSTIFICACIONES DE LA INVESTIGACION

El desconocimiento de los bienes informáticos en la Marina de Guerra del Perú, ha traído como consecuencia, la falta de integración de los datos de identificación física y de antecedentes jurídicos y administrativos de los bienes informáticos, para tener un registro adecuado de los activos de esta institución pública.

Con la instalación y puesta en marcha del proyecto de registro y asignación de los artículos informáticos de las Fuerzas Armadas, que sirve para conocer las propiedades con que cuenta dicha institución pública, se podrá utilizar el patrimonio público de una manera ordenada y eficiente, haciendo uso de las tecnologías de Información e implementando este software llamado *Sistemas de Control de Asignación de Artículos Informáticos Marina de Guerra del Perú*

1.6. HIPÓTESIS

1.6.1. Hipótesis general

El uso de las Tic's en el sistema de gestión y control y el componente tecnológico hará que la asignación de los recursos informáticos optimice los recursos informáticos, contribuyendo a la reducción de costos, calidad del servicio y de las utilidades de la organización.

1.6.2. Hipótesis específicas

- a) El desarrollo de un sistema de Gestión permitirá optimizar la toma de decisiones en el subsistema asignación de los recursos de la empresa.
- b) La optimización de la asignación reduce el tiempo de entrega de los recursos informáticos en la empresa
- c) El uso de las Tic's mejorará la toma de decisiones en la organización

CAPITULO II

MARCO TEORICO

2.1 DIRECCIÓN DE TELEMÁTICA

2.1.1 Reseña Histórica

El 28 de Agosto de 1907 los cruceros BAP "Almirante Grau" y BAP "Coronel Bolognesi" recién adquiridos y dotados con equipos de última generación, lograron durante un ejercicio en la mar, materializar por primera vez en el Perú de forma estable una comunicación inalámbrica de la Banda HF a una distancia de 200 Millas entre ellos desde el Puerto de Salaverry al Puerto de Ilo, fecha que es considerada el "Día de las Comunicaciones Navales".

El 25 de Mayo de 1925 se crea la "Oficina de Publicaciones Navales" (OFICONA) como Primer Órgano de Control de Comunicaciones de la Marina, dependiente del Estado Mayor General y situado en la Escuela Naval. A través de los años fue siendo cambiada de lugar y denominación hasta su instalación en la Dirección de Telemática el 23 de Noviembre de 1978, convirtiéndose en la Dirección de Comunicaciones Navales (DIRCOMU).

El 20 de Octubre de 1972 se crea la "Central de Procesamiento de Datos" (CEPRODA) situada en la Dirección de Abastecimiento Naval, la cual fue

siendo trasladada hasta su ubicación actual el 20 Noviembre de 1986 cambiándosele de nombre a “Servicio de Informática de la Marina” (SINFO)

El 1 de Agosto de 1995 se fusionaron la Dirección de Comunicaciones Navales con el Servicio de Informática de la Marina, creándose la actual “Dirección de Telemática” (DIRTEL)

2.1.2 Misión

“Planear, normar, dirigir, operar, controlar y evaluar los sistemas de comunicaciones e informática de la Marina de Guerra del Perú, con el fin de garantizar a las Unidades y Dependencias enlaces seguros e invulnerables, los cuales interconectados mediante sistemas de procesamiento de información contribuyen a la eficiencia en la toma de decisiones y al ejercicio oportuno y efectivo del comando

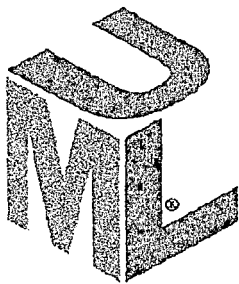
2.1.3 Visión

Con el desarrollo de ésta aplicación se busca que se efectúe un debido control del software licenciado con que cuenta la Marina de Guerra del Perú, logrando con ello que el alto mando naval y los Comités de Licenciamiento cuenten con

2.2 BASES TEÓRICAS

2.2.1 Lenguaje de Modelamiento Unificado

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.



UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo.

El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

UML no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguaje de programación, así como construir modelos por ingeniería inversa a partir de programas existentes.

Es un lenguaje de propósito general para el modelado orientado a objetos. UML es también un lenguaje de modelamiento visual que permite una abstracción del sistema y sus componentes.

Existían diversos métodos y técnicas Orientadas a Objetos, con muchos aspectos en común pero utilizando distintas notaciones, se presentaban inconvenientes para el aprendizaje, aplicación, construcción y uso de herramientas, etc., además de pugnas entre enfoques, lo que generó la creación del UML como estándar para el modelamiento de sistemas de software principalmente, pero con posibilidades de ser aplicado a todo tipo de proyectos.

2.2.2 Objetivos del UML

- UML es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.
- UML no pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.
- Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. UML necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en

un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son la encapsulación y componentes.

- Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.
- Imponer un estándar mundial.

2.2.3 Orientación a objetos

Permite al programador que organice su programa de acuerdo con abstracciones de más alto nivel, siendo estas más cercanas a la manera de pensar de la gente. Ejemplo: Cuentas de bancos, reservaciones de vuelos. Etc.

Los datos globales desaparecen por lo tanto cualquier cambio en la estructura de algunos de los datos sólo debiera afectar las funciones definidas en ese mismo objeto y no en los demás, en general un programa orientado a objetos se define en términos de objetos y relaciones. Los datos y funciones se guardan dentro de objetos. Los datos están ubicados en el centro del objeto, lo cual hace que un cambio en su estructura solo afecte sus funciones del mismo objeto, pero no al resto de la aplicación.

2.2.4 Fases de Desarrollo

Diseño Es un refinamiento y formalización adicional del modelo del análisis

El resultado del modelo de diseño son especificaciones muy detalladas de todos los objetos, incluyendo sus operaciones y atributos. Se requiere ya que el modelo de análisis no es lo suficientemente formal para alcanzar el código fuente. Otro objetivo del diseño es validar los resultados de requisito y análisis.

El diseño añade el ambiente de implementación como un nuevo eje de desarrollo.

El diseño se basará en el diseño de objetos o sea en una extensión directa y detallada del modelo de análisis que describa los atributos y operaciones del sistema. También podemos decir que es una continuación del modelo del análisis.

Implementación Toma el resultado del modelo de diseño para generar el código final. Se adapta al lenguaje de programación y/o a la base de datos, según la especificación del diseño y las propiedades del lenguaje de implementación de base de datos.

Aunque el diseño de objetos es bastante independiente del lenguaje actual, todos los lenguajes tienen sus particularidades, las cuales deben adecuarse durante la implementación final.

Modelado de Objeto Describe los conceptos principales de la orientación a objetos:

Las estructuras estáticas y sus relaciones. Las principales estructuras estáticas son los objetos y clases, los cuales están compuestos de atributos y operaciones, mientras que las principales relaciones entre objetos y clases corresponden a las ligas y asociaciones, respectivamente.

Modelo de Abstracción

Una parte del UML define, entonces, una abstracción con significado de un lenguaje para expresar otros modelos (es decir, otras abstracciones de un sistema, o conjunto de unidades conectadas que se organizan para conseguir un propósito). Lo que en principio puede parecer complicado no lo es tanto si pensamos que uno de los objetivos del UML es llegar a convertirse en una manera de definir modelos, no sólo establecer una forma de modelo, de esta forma simplemente estamos diciendo que UML, además, define un lenguaje con el que podemos abstraer cualquier tipo de modelo.

El UML es una técnica de modelado de objetos y como tal supone una abstracción de un sistema para llegar a construirlo en términos concretos. El modelado no es más que la construcción de un modelo a partir de una especificación.

Un modelo es una abstracción de algo, que se elabora para comprender ese algo antes de construirlo. El modelo omite detalles que no resultan esenciales para la comprensión del original y por lo tanto facilita dicha comprensión.

2.2.5 Artefactos para el Desarrollo de Proyectos

Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software. Pueden ser artefactos un modelo, una descripción o un software. Los artefactos de UML se especifican en forma de diagramas, éstos, junto con la documentación sobre el sistema constituyen los artefactos principales que el modelador puede observar.

Se necesita más de un punto de vista para llegar a representar un sistema. UML utiliza los **diagramas gráficos** para obtener estos distintos puntos de vista de un sistema:

- Diagramas de Implementación.
- Diagramas de Comportamiento o Interacción.
- Diagramas de Casos de uso.
- Diagramas de Clases.

Diagramas de Implementación

Los diagramas de implementación muestran los aspectos físicos del sistema. Incluyen la estructura del código fuente y la implementación, en tiempo de implementación.

Diagramas de Comportamiento o Interacción

Muestran las interacciones entre objetos ocurridas en un escenario (Parte) del sistema.

Diagramas de Casos de uso:

Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona una respuesta a eventos que se producen en el mismo. En este tipo de diagrama intervienen algunos conceptos nuevos: un actor es una entidad externa al sistema que se modela y que puede interactuar con él; un ejemplo de actor podría ser un usuario o cualquier otro sistema. Las relaciones entre casos de uso y actores pueden ser las siguientes:

- Un actor se comunica con un caso de uso.
- Un caso de uso extiende otro caso de uso.
- Un caso de uso usa otro caso de uso

Diagramas de Clases

Los diagramas de clases representan un conjunto de elementos del modelo que son estáticos, como las clases y los tipos, sus contenidos y las relaciones que se establecen entre ellos.

2.2.6 El Proceso de Desarrollo

UML no define un proceso concreto que determine las fases de desarrollo de un sistema, las empresas pueden utilizar UML como el lenguaje para definir sus propios procesos y lo único que tendrán en común con otras organizaciones que utilicen UML serán los tipos de diagramas.

UML es un método independiente del proceso. Los procesos de desarrollo deben ser definidos dentro del contexto donde se van a implementar los sistemas.

Rational Rose, es la herramienta CASE que comercializan los desarrolladores de UML y que soporta de forma completa la especificación del UML. Esta herramienta propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software.

2.2.7 Proceso Unificado de desarrollo RUP

Este documento presenta un resumen de Rational Unified Process (RUP). Se describe la historia de la metodología, características principales y estructura del proceso. RUP es un producto comercial desarrollado y comercializado por Rational Software, una compañía de IBM.

Historia

El antecedente más importante se ubica en 1967 con la Metodología Ericsson (Ericsson Approach) elaborada por Ivar Jacobson, una aproximación de desarrollo basada en componentes, que introdujo el concepto de Caso de Uso. Entre los años de 1987 a 1995 Jacobson fundó la compañía Objectory AB y lanza el proceso de desarrollo Objectory (abreviación de Object Factory).

Posteriormente en 1995 Rational Software Corporation adquiere Objectory AB y entre 1995 y 1997 se desarrolla Rational Objectory Process (ROP) a partir de Objectory 3.8 y del Enfoque Rational (Rational Approach) adoptando UML como lenguaje de modelado.

Desde ese entonces y a la cabeza de Grady Booch, Ivar Jacobson y James Rumbaugh, Rational Software desarrolló e incorporó diversos elementos para expandir ROP, destacándose especialmente el flujo de trabajo conocido como modelado del negocio. En junio del 1998 se lanza Rational Unified Process.

La historia de RUP se ilustra en el (*anexo 9.1*).

a) Características esenciales

Los autores de RUP destacan que el proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental.

b) Proceso dirigido por Casos de Uso

Según [KRU00], los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema.

En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los Casos de Uso constituyen un elemento integrador y una guía del trabajo como se muestra en el (*anexo 9.2*).

Los Casos de Uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los

artefactos que son generados en las diferentes actividades del proceso de desarrollo.

Como se muestra en el (anexo 9.3), basándose en los Casos de Uso se crean los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que efectivamente el producto implemente adecuadamente cada Caso de Uso. Todos los modelos deben estar sincronizados con el modelo de Casos de Uso.

c) Proceso centrado en la arquitectura

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo [KRU00].

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de

datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema.

En el caso de RUP además de utilizar los Casos de Uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

Cada producto tiene tanto una función como una forma. La función corresponde a la funcionalidad reflejada en los Casos de Uso y la forma la proporciona la arquitectura. Existe una interacción entre los Casos de Uso y la arquitectura, los Casos de Uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los Casos de Uso requeridos, actualmente y en el futuro. Esto provoca que tanto arquitectura como Casos de Uso deban evolucionar en paralelo durante todo el proceso de desarrollo de software.

En el (anexo 9.4) se ilustra la evolución de la arquitectura durante las fases de RUP. Se tiene una arquitectura más robusta en las fases finales del proyecto. En las fases iniciales lo que se hace es ir consolidando la arquitectura por

medio de baselines y se va modificando dependiendo de las necesidades del proyecto.

Es conveniente ver el sistema desde diferentes perspectivas para comprender mejor el diseño por lo que la arquitectura se representa mediante varias vistas que se centran en aspectos concretos del sistema, abstrayéndose de los demás. Para RUP, todas las vistas juntas forman el llamado modelo 4+1 de la arquitectura [KRU00], el cual recibe este nombre porque lo forman las vistas lógica, de implementación, de proceso y de despliegue, más la de Casos de Uso que es la que da cohesión a todas.

Al final de la fase de elaboración se obtiene una baseline de la arquitectura donde fueron seleccionados una serie de Casos de Uso arquitectónicamente relevantes (aquellos que ayudan a mitigar los riesgos más importantes, aquellos que son los más importantes para el usuario y aquellos que cubran las funcionalidades significativas)

Como se observa en el (*anexo 9.5*), durante la construcción los diversos modelos van desarrollándose hasta completarse (según se muestra con las formas rellenas en la esquina superior derecha). La descripción de la arquitectura sin embargo, no debería cambiar significativamente debido a que la mayor parte de la arquitectura se decidió durante la elaboración. Se incorporan

pocos cambios a la arquitectura (indicados con mayor densidad de puntos) [JBR99].

d) Proceso iterativo e incremental

Según [JBR99] el equilibrio correcto entre los Casos de Uso y la arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

Una iteración puede realizarse por medio de una cascada como se muestra en el (anexo 9.6). Se pasa por los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y Pruebas), también existe una planificación de la iteración, un análisis de la iteración y algunas actividades específicas de la iteración. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones anteriores.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. En el (anexo 9.7) se muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

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

e) Estructura del proceso

El proceso puede ser descrito en dos dimensiones o ejes [RSC02]:

Eje horizontal: Representa el tiempo y es considerado el eje de los aspectos dinámicos del proceso. Indica las características del ciclo de vida del proceso expresado en términos de fases, iteraciones e hitos. Se puede observar en el (anexo 9.8) que RUP consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Como se mencionó anteriormente cada fase se subdivide a la vez en iteraciones.

Eje vertical: Representa los aspectos estáticos del proceso. Describe el proceso en términos de componentes de proceso, disciplinas, flujos de trabajo, actividades, artefactos y roles (*anexo 9.8*)

f) Estructura Dinámica del proceso. Fases e iteraciones

RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un producto. Cada ciclo concluye con una generación del producto para los clientes. Cada ciclo consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase se subdivide a la vez en iteraciones, el número de iteraciones en cada fase es variable. (*anexo 9.9*)

Cada fase se concluye con un hito bien definido, un punto en el tiempo en el cual se deben tomar ciertas decisiones críticas y alcanzar las metas clave antes de pasar a la siguiente fase, ese hito principal de cada fase se compone de hitos menores que podrían ser los criterios aplicables a cada iteración. Los hitos para cada una de las fases son: Inicio - Lifecycle Objectives, Elaboración - Lifecycle Architecture, Construcción - Initial Operational Capability, Transición - Product Release. Las fases y sus respectivos hitos se ilustran en el (*anexo 9.10*).

La duración y esfuerzo dedicado en cada fase es variable dependiendo de las características del proyecto. Sin embargo, el (*anexo 9.11*) ilustra porcentajes

frecuentes al respecto. Consecuente con el esfuerzo señalado, el (*anexo 9.12*) ilustra una distribución típica de recursos humanos necesarios a lo largo del proyecto.

Inicio

Durante la fase de inicio se define el modelo del negocio y el alcance del proyecto. Se identifican todos los actores y Casos de Uso, y se diseñan los Casos de Uso más esenciales (aproximadamente el 20% del modelo completo). Se desarrolla, un plan de negocio para determinar que recursos deben ser asignados al proyecto.

Los objetivos de esta fase son [KRU00]:

- Establecer el ámbito del proyecto y sus límites.
- Encontrar los Casos de Uso críticos del sistema, los escenarios básicos que definen la funcionalidad.
- Mostrar al menos una arquitectura candidata para los escenarios principales.
- Estimar el coste en recursos y tiempo de todo el proyecto.
- Estimar los riesgos, las fuentes de incertidumbre.

Los resultados de la fase de inicio deben ser [RSC02]:

- Un documento de visión: Una visión general de los requerimientos del

proyecto, características clave y restricciones principales.

- Modelo inicial de Casos de Uso (10-20% completado).
- Un glosario inicial: Terminología clave del dominio.
- El caso de negocio.
- Lista de riesgos y plan de contingencia.
- Plan del proyecto, mostrando fases e iteraciones.
- Modelo de negocio, si es necesario
- Prototipos exploratorios para probar conceptos o la arquitectura candidata.

Al terminar la fase de inicio se deben comprobar los criterios de evaluación para continuar:

- Todos los interesados en el proyecto coinciden en la definición del ámbito del sistema y las estimaciones de agenda.
- Entendimiento de los requisitos, como evidencia de la fidelidad de los Casos de Uso principales.
- Las estimaciones de tiempo, coste y riesgo son creíbles.
- Comprensión total de cualquier prototipo de la arquitectura desarrollado.
- Los gastos hasta el momento se asemejan a los planeados.

Si el proyecto no pasa estos criterios hay que plantearse abandonarlo o repensarlo profundamente.

Elaboración

El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos.

En esta fase se construye un prototipo de la arquitectura, que debe evolucionar en iteraciones sucesivas hasta convertirse en el sistema final. Este prototipo debe contener los Casos de Uso críticos identificados en la fase de inicio. También debe demostrarse que se han evitado los riesgos más graves.

Los objetivos de esta fase son [KRU00]:

- Definir, validar y cimentar la arquitectura.
- Completar la visión.
- Crear un plan fiable para la fase de construcción. Este plan puede evolucionar en sucesivas iteraciones. Debe incluir los costes si procede.
- Demostrar que la arquitectura propuesta soportará la visión con un coste razonable y en un tiempo razonable.

Al terminar deben obtenerse los siguientes resultados [RSC02]:

- Un modelo de Casos de Uso completa al menos hasta el 80%: todos los casos y actores identificados, la mayoría de los casos desarrollados.
- Requisitos adicionales que capturan los requisitos no funcionales y cualquier requisito no asociado con un Caso de Uso específico.

- Descripción de la arquitectura software.
- Un prototipo ejecutable de la arquitectura.
- Lista de riesgos y caso de negocio revisados.
- Plan de desarrollo para el proyecto.
- Un caso de desarrollo actualizado que especifica el proceso a seguir.
- Un manual de usuario preliminar (opcional).

En esta fase se debe tratar de abarcar todo el proyecto con la profundidad mínima. Sólo se profundiza en los puntos críticos de la arquitectura o riesgos importantes.

En la fase de elaboración se actualizan todos los productos de la fase de inicio.

Los criterios de evaluación de esta fase son los siguientes:

- La visión del producto es estable.
- La arquitectura es estable.
- Se ha demostrado mediante la ejecución del prototipo que los principales elementos de riesgo han sido abordados y resueltos.
- El plan para la fase de construcción es detallado y preciso. Las estimaciones son creíbles.
- Todos los interesados coinciden en que la visión actual será alcanzada si se siguen los planes actuales en el contexto de la arquitectura actual.

- Los gastos hasta ahora son aceptables, comparados con los previstos.

Si no se superan los criterios de evaluación quizá sea necesario abandonar el proyecto o replanteárselo considerablemente.

Construcción

La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. Durante esta fase todos los componentes, características y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto.

Los objetivos concretos según [KRU00] incluyen:

- Minimizar los costes de desarrollo mediante la optimización de recursos y evitando el tener que rehacer un trabajo o incluso desecharlo.
- Conseguir una calidad adecuada tan rápido como sea práctico.
- Conseguir versiones funcionales (alfa, beta, y otras versiones de prueba) tan rápido como sea práctico.

Los resultados de la fase de construcción deben ser [RSC02]:

- Modelos Completos (Casos de Uso, Análisis, Diseño, Despliegue e Implementación)

- Arquitectura íntegra (mantenida y mínimamente actualizada)
- Riesgos Presentados Mitigados
- Plan del Proyecto para la fase de Transición.
- Manual Inicial de Usuario (con suficiente detalle)
- Prototipo Operacional – beta
- Caso del Negocio Actualizado

Los criterios de evaluación de esta fase son los siguientes:

1. El producto es estable y maduro como para ser entregado a la comunidad de usuario para ser probado.
2. Todos los usuarios expertos están listos para la transición en la comunidad de usuarios.
3. Son aceptables los gastos actuales versus los gastos planeados.

Transición

La finalidad de la fase de transición es poner el producto en manos de los usuarios finales, para lo que se requiere desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto, y en general tareas relacionadas con el ajuste, configuración, instalación y facilidad de uso del producto.

En [KRU00] se citan algunas de las cosas que puede incluir esta fase:

- Prueba de la versión Beta para validar el nuevo sistema frente a las expectativas de los usuarios
- Funcionamiento paralelo con los sistemas legados que están siendo sustituidos por nuestro proyecto.
- Conversión de las bases de datos operacionales.
- Entrenamiento de los usuarios y técnicos de mantenimiento.
- Traspaso del producto a los equipos de marketing, distribución y venta.

Los principales objetivos de esta fase son:

- Conseguir que el usuario se valga por si mismo.
- Un producto final que cumpla los requisitos esperados, que funcione y satisfaga suficientemente al usuario.

Los resultados de la fase de transición son [RSC02]:

- Prototipo Operacional
- Documentos Legales
- Caso del Negocio Completo
- Línea de Base del Producto completa y corregida que incluye todos los modelos del sistema
- Descripción de la Arquitectura completa y corregida
- Las iteraciones de esta fase irán dirigidas normalmente a conseguir una nueva versión.

Los criterios de evaluación de esta fase son los siguientes:

- El usuario se encuentra satisfecho.
- Son aceptables los gastos actuales versus los gastos planificados.

g) Estructura Estática del proceso. Roles, actividades, artefactos y flujos de trabajo

Un proceso de desarrollo de software define quién hace qué, cómo y cuándo. RUP define cuatro elementos los roles, que responden a la pregunta ¿Quién?, las actividades que responden a la pregunta ¿Cómo?, los productos, que responden a la pregunta ¿Qué? y los flujos de trabajo de las disciplinas que responde a la pregunta ¿Cuándo? (ver *anexos 9.13 y 9.14*) [RSC02].

Roles

Un rol define el comportamiento y responsabilidades de un individuo, o de un grupo de individuos trabajando juntos como un equipo. Una persona puede desempeñar diversos roles, así como un mismo rol puede ser representado por varias personas.

Las responsabilidades de un rol son tanto el llevar a cabo un conjunto de actividades como el ser el dueño de un conjunto de artefactos.

RUP define grupos de roles, agrupados por participación en actividades relacionadas. Estos grupos son [RSC02]:

Analistas:

- Analista de procesos de negocio.
- Diseñador del negocio.
- Analista de sistema.
- Especificador de requisitos.

Desarrolladores:

- Arquitecto de software.
- Diseñador
- Diseñador de interfaz de usuario
- Diseñador de cápsulas.
- Diseñador de base de datos.
- Implementador.
- Integrador.

Gestores:

- Jefe de proyecto
- Jefe de control de cambios.
- Jefe de configuración.

- Jefe de pruebas
- Jefe de despliegue
- Ingeniero de procesos
- Revisor de gestión del proyecto
- Gestor de pruebas.

Apoyo:

- Documentador técnico
- Administrador de sistema
- Especialista en herramientas
- Desarrollador de cursos
- Artista gráfico

Especialista en pruebas:

- Especialista en Pruebas (tester)
- Analista de pruebas
- Diseñador de pruebas

Otros roles:

- Stakeholders.
- Revisor
- Coordinación de revisiones

- Revisor técnico
- Cualquier rol

Actividades

Una actividad en concreto es una unidad de trabajo que una persona que desempeñe un rol puede ser solicitado a que realice. Las actividades tienen un objetivo concreto, normalmente expresado en términos de crear o actualizar algún producto.

Artefactos

Un producto o artefacto es un trozo de información que es producido, modificado o usado durante el proceso de desarrollo de software. Los productos son los resultados tangibles del proyecto, las cosas que va creando y usando hasta obtener el producto final.

Un artefacto puede ser cualquiera de los siguientes [RSC02]:

- Un documento, como el documento de la arquitectura del software.
- Un modelo, como el modelo de Casos de Uso o el modelo de diseño.
- Un elemento del modelo, un elemento que pertenece a un modelo como una clase, un Caso de Uso o un subsistema.

Flujos de trabajo

Con la enumeración de roles, actividades y artefactos no se define un proceso, necesitamos contar con una secuencia de actividades realizadas por los diferentes roles, así como la relación entre los mismos. Un flujo de trabajo es una relación de actividades que nos producen unos resultados observables. A continuación se dará una explicación de cada flujo de trabajo.

1. Modelado del negocio

Con este flujo de trabajo pretendemos llegar a un mejor entendimiento de la organización donde se va a implantar el producto.

Los objetivos del modelado de negocio son [RSC02]:

- Entender la estructura y la dinámica de la organización para la cual el sistema va ser desarrollado (organización objetivo).
- Entender el problema actual en la organización objetivo e identificar potenciales mejoras.
- Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización objetivo.
- Derivar los requisitos del sistema necesarios para apoyar a la organización objetivo.

Para lograr estos objetivos, el modelo de negocio describe como desarrollar una visión de la nueva organización, basado en esta visión se definen procesos, roles y responsabilidades de la organización por medio de un modelo de Casos de Uso del negocio y un Modelo de Objetos del Negocio. Complementario a estos modelos, se desarrollan otras especificaciones tales como un Glosario.

2. Requisitos

Este es uno de los flujos de trabajo más importantes, porque en él se establece qué tiene que hacer exactamente el sistema que construyamos. En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que especifiquemos.

Los objetivos del flujo de datos Requisitos es [RSC02]:

- Establecer y mantener un acuerdo entre clientes y otros stakeholders sobre lo que el sistema podría hacer.
- Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema.
- Definir el ámbito del sistema.
- Proveer una base para la planeación de los contenidos técnicos de las iteraciones.
- Proveer una base para estimar costos y tiempo de desarrollo del sistema.

- Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.

Los requisitos se dividen en dos grupos. Los requisitos funcionales representan la funcionalidad del sistema. Se modelan mediante diagramas de Casos de Uso. Los requisitos no funcionales representan aquellos atributos que debe exhibir el sistema, pero que no son una funcionalidad específica. Por ejemplo requisitos de facilidad de uso, fiabilidad, eficiencia, portabilidad, etc.

Para capturar los requisitos es preciso entrevistar a todos los interesados en el proyecto, no sólo a los usuarios finales, y anotar todas sus peticiones. A partir de ellas hay que descubrir lo que necesitan y expresarlo en forma de requisitos. En este flujo de trabajo, y como parte de los requisitos de facilidad de uso, se diseña la interfaz gráfica de usuario. Para ello habitualmente se construyen prototipos de la interfaz gráfica de usuario que se contrastan con el usuario final.

3. Análisis y Diseño

El objetivo de este flujo de trabajo es traducir los requisitos a una especificación que describe cómo implementar el sistema.

Los objetivos del análisis y diseño son [RSC02]:

- Transformar los requisitos al diseño del futuro sistema.
- Desarrollar una arquitectura para el sistema.
- Adaptar el diseño para que sea consistente con el entorno de implementación.

El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales. Por otro lado el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos.

Al principio de la fase de elaboración hay que definir una arquitectura candidata: crear un esquema inicial de la arquitectura del sistema, identificar clases de análisis y actualizar las realizaciones de los Casos de Uso con las interacciones de las clases de análisis. Durante la fase de elaboración se va refinando esta arquitectura hasta llegar a su forma definitiva. En cada iteración hay que analizar el comportamiento para diseñar componentes. Además si el sistema usará una base de datos, habrá que diseñarla también, obteniendo un modelo de datos.

El resultado final más importante de este flujo de trabajo será el modelo de diseño. Consiste en colaboraciones de clases, que pueden ser agregadas en paquetes y subsistemas.

Otro producto importante de este flujo es la documentación de la arquitectura de software, que captura varias vistas arquitectónicas del sistema.

4. Implementación

En este flujo de trabajo se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. Además se deben hacer las pruebas de unidad: cada implementador es responsable de probar las unidades que produzca. El resultado final de este flujo de trabajo es un sistema ejecutable.

En cada iteración habrá que hacer lo siguiente:

- Planificar qué subsistemas deben ser implementados y en qué orden deben ser integrados, formando el Plan de Integración.
- Cada implementador decide en qué orden implementa los elementos del subsistema.
- Si encuentra errores de diseño, los notifica.
- Se prueban los subsistemas individualmente.
- Se integra el sistema siguiendo el plan.

La estructura de todos los elementos implementados forma el modelo de implementación. La integración debe ser incremental, es decir, en cada momento sólo se añade un elemento. De este modo es más fácil localizar fallos y los componentes se prueban más a fondo. En fases tempranas del proceso se

pueden implementar prototipos para reducir el riesgo. Su utilidad puede ir desde ver si el sistema es viable desde el principio, probar tecnologías o diseñar la interfaz de usuario. Los prototipos pueden ser exploratorios (desechables) o evolutivos. Estos últimos llegan a transformarse en el sistema final.

5. Pruebas

Este flujo de trabajo es el encargado de evaluar la calidad del producto que estamos desarrollando, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida.

Esta disciplina brinda soporte a las otras disciplinas. Sus objetivos son [RSC02]:

- Encontrar y documentar defectos en la calidad del software.
- Generalmente asesora sobre la calidad del software percibida.
- Provee la validación de los supuestos realizados en el diseño y especificación de requisitos por medio de demostraciones concretas.
- Verificar las funciones del producto de software según lo diseñado.
- Verificar que los requisitos tengan su apropiada implementación.

Las actividades de este flujo comienzan pronto en el proyecto con el plan de prueba (el cual contiene información sobre los objetivos generales y específicos de las prueba en el proyecto, así como las estrategias y recursos con que se

dotará a esta tarea), o incluso antes con alguna evaluación durante la fase de inicio, y continuará durante todo el proyecto.

El desarrollo del flujo de trabajo consistirá en planificar que es lo que hay que probar, diseñar cómo se va a hacer, implementar lo necesario para llevarlos a cabo, ejecutarlos en los niveles necesarios y obtener los resultados, de forma que la información obtenida nos sirva para ir refinando el producto a desarrollar.

6. Despliegue

El objetivo de este flujo de trabajo es producir con éxito distribuciones del producto y distribuirlo a los usuarios. Las actividades implicadas incluyen:

- Probar el producto en su entorno de ejecución final.
- Empaquetar el software para su distribución.
- Distribuir el software.
- Instalar el software.
- Proveer asistencia y ayuda a los usuarios.
- Formar a los usuarios y al cuerpo de ventas.
- Migrar el software existente o convertir bases de datos.

Este flujo de trabajo se desarrolla con mayor intensidad en la fase de transición, ya que el propósito del flujo es asegurar una aceptación y adaptación sin

complicaciones del software por parte de los usuarios. Su ejecución inicia en fases anteriores, para preparar el camino, sobre todo con actividades de planificación, en la elaboración del manual de usuario y tutoriales.

7. Gestión del proyecto

La Gestión del proyecto es el arte de lograr un balance al gestionar objetivos, riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes y los usuarios.

Los objetivos de este flujo de trabajo son:

- Proveer un marco de trabajo para la gestión de proyectos de software intensivos.
- Proveer guías prácticas realizar planeación, contratar personal, ejecutar y monitorear el proyecto.
- Proveer un marco de trabajo para gestionar riesgos.

La planeación de un proyecto posee dos niveles de abstracción: un plan para las fases y un plan para cada iteración.

8. Configuración y control de cambios

La finalidad de este flujo de trabajo es mantener la integridad de todos los artefactos que se crean en el proceso, así como de mantener información del proceso evolutivo que han seguido.

9. Entorno

La finalidad de este flujo de trabajo es dar soporte al proyecto con las adecuadas herramientas, procesos y métodos. Brinda una especificación de las herramientas que se van a necesitar en cada momento, así como definir la instancia concreta del proceso que se va a seguir.

En concreto las responsabilidades de este flujo de trabajo incluyen:

- Selección y adquisición de herramientas
- Establecer y configurar las herramientas para que se ajusten a la organización.
- Configuración del proceso.
- Mejora del proceso.
- Servicios técnicos.

El principal artefacto que se usa en este flujo de trabajo es el caso de desarrollo que especifica para el proyecto actual en concreto, como se aplicará el proceso, que productos se van a utilizar y como van a ser utilizados. Además se tendrán

que definir las guías para los distintos aspectos del proceso, como pueden ser el modelado del negocio y los Casos de Uso, para la interfaz de usuario, el diseño, la programación, el manual de usuario.

h) Una configuración RUP para proyecto pequeño

En este apartado se describe una posible configuración de RUP para un proyecto pequeño. Por las características del proyecto, se han incluido muy pocos artefactos, roles y actividades de la metodología, manteniendo los más esenciales. Dicha configuración está basada en la siguiente selección de artefactos:

Entregables del proyecto

A continuación se describen brevemente cada uno de los artefactos que se generarán y usarán durante el proyecto.

1. Flujos de Trabajo

Se utilizarán Diagramas de Actividad para modelar los Flujos de Trabajo (workflows) del área problema, tanto los actuales (previos a la implantación de nuevo sistema) como los propuestos, que serán soportados por el sistema desarrollado

2. Características del Producto Software

Es una lista de las características principales del producto, deseables desde una perspectiva de las necesidades del cliente.

3. Glosario

Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada.

4. Modelo de Casos de Uso

El modelo de Casos de Uso presenta la funcionalidad del sistema y los actores que hacen uso de ella. Se representa mediante Diagramas de Casos de Uso.

5. Especificaciones de Casos de Uso

Para los casos de uso que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, postcondiciones, flujo de eventos, requisitos no-funcionales asociados.

6. Modelo de Análisis y Diseño

Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de

implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación). Está constituido esencialmente por un Diagrama de Clases y algunos Diagramas de Estados para las clases que lo requieran.

7. Modelo Lógico Relacional

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Tablas donde se muestran las tablas, claves, etc.

8. Modelo de Implementación

Este modelo es una colección de componentes y los subsistemas que los contienen. Estos componentes incluyen: ficheros ejecutables, ficheros de código fuente, y todo otro tipo de ficheros necesarios para la implantación y despliegue del sistema.

9. Modelo de Pruebas

Para cada Caso de Uso se establecen pruebas de Aceptación que validarán la correcta implementación del Caso de Uso. Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados.

10. Manual de Instalación

Este documento incluye las instrucciones para realizar la instalación del producto.

11. Material de Usuario

Corresponde a un conjunto de documentos y facilidades de uso del sistema.

12. Producto

Todos los ficheros fuente y ejecutable del producto.

Esquema de trazabilidad

El (*anexo 9.15*) ilustra las relaciones de trazabilidad entre artefactos del proyecto, y según la configuración antes mencionada.

Las relaciones de trazabilidad son enlaces entre artefactos que establecen cómo se generan unos a partir de otros. Esto permite por ejemplo asegurar la cobertura de los requisitos o determinar el posible impacto de los cambios. En el (*anexo 9.15*) se ilustran los modelos y artefactos utilizados, indicando las relaciones de trazabilidad entre ellos, lo cual se resume a continuación:

- Se modelarán los procesos de negocio de la situación actual utilizando Diagramas de Actividad para representar Flujos de Trabajo Actuales.

Esto se complementará mediante un Glosario que establecerá la terminología.

- El modelo de procesos de la solución propuesta incluirá Flujos de Trabajo Propuestos junto con una lista de Características del Producto Software.
- Los requisitos serán establecidos mediante un Modelo de Casos de Uso que incluirá Diagramas de Casos de Uso, Prototipos de Interfaces de Usuario y Especificaciones de Casos de Uso.
- El Modelo de Pruebas incluirá las Pruebas de Aceptación establecidas para cada Caso de Uso.
- El Modelo de Análisis y Diseño establecerá el particionamiento interno del sistema. Estará compuesto por un Diagrama de Clases y algunos Diagramas de Estados. Las clases determinarán la estructura y las operaciones necesarias para implementar las funcionalidades descritas en los Casos de Uso. Los Diagramas de Estados detallarán el comportamiento para las clases que lo requieran.
- A partir del Diagrama de Clases, y considerando las clases que requieran persistencia, se derivará el Modelo Lógico Relacional, representado mediante Diagramas de Tablas.
- En el Modelo de Implementación se organizarán las operaciones de las clases en términos de componentes de dicha arquitectura. Esto se representará mediante Diagramas de Componentes.

- La implementación del Modelo Lógico Relacional y de los componentes de la aplicación constituirán el Producto, el cual se complementará con el Manual de Instalación y el Manual de Usuario.

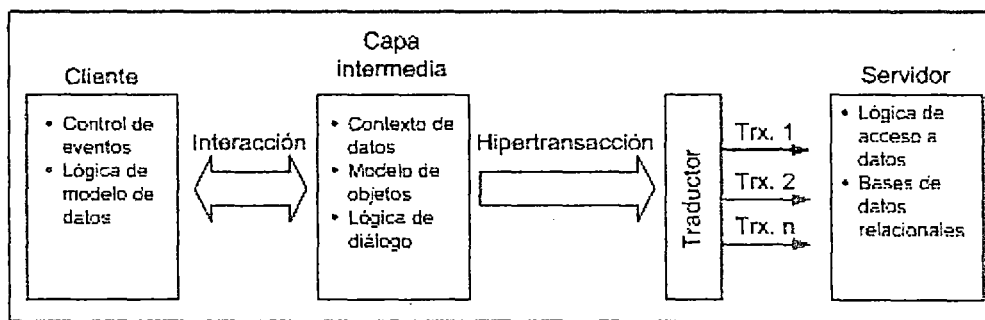
2.2.8 Arquitectura tecnológica

La arquitectura tecnológica básica de un sistema de proceso transaccional se fundamenta en un modelo Cliente-Servidor de dos capas en la que la capa servidora es un sistema remoto y la capa cliente es muy ligera, o sea, casi no tiene capacidad de ejecutar lógica de aplicación. En un modelo transaccional básico, todo el proceso y la gestión del diálogo se ejecuta en la capa servidora.

A continuación propongo los cambios y nuevos elementos a añadir a esta arquitectura:

1. Control de eventos y ejecución de lógica asociada al modelo de datos en la capa cliente: eso permitirá a la interfaz contener objetos con comportamiento propio.
2. Mantenimiento de contexto de datos con orientación a objetos en una capa intermedia para permitir que el usuario disponga de un espacio controlado por él mismo.

3. Ubicar la lógica de diálogo en una capa intermedia permitiendo que las acciones del usuario no afecten a los datos del servidor. De este modo se ayuda a mejorar la reversibilidad de las acciones del usuario.
4. Hipertransacciones que se traduzcan en grupos de transacciones simples, con el objetivo de facilitar la actualización de los datos del servidor a partir de los objetos ubicados en la capa intermedia.
5. Traductor del modelo lógico de objetos al modelo físico relacional, situado, posiblemente, en la capa servidora o en una nueva capa intermedia. Este traductor se basa en un sistema de equivalencias entre los objetos existentes en la interfaz de usuario y las bases de datos relacionales de los servidores.



Propuesta de nueva arquitectura

2.2.9 FODA

El **FODA** o **DOFA**, (**SWOT**, por sus siglas en inglés), es una técnica de planeación estratégica que permite crear o reajustar a una estrategia, ya sea de negocios, mercadotecnia, comunicación, relaciones públicas, etc...

El cual permitirá conformar un cuadro de la situación actual de una empresa u organización; permitiendo, de esta manera, obtener un diagnóstico preciso que permita tomar decisiones acordes con los objetivos y políticas formulados por tal organismo.

El término **FODA** es una sigla conformada por las primeras letras de las palabras:

1. Fortalezas
2. Oportunidades
3. Debilidades
4. Amenazas

Para determinar las fortalezas y debilidades debes de tener conocimiento de la empresa, de sus servicios y/o productos para poder determinarlas. Las fortalezas son los diferenciadores con respecto a la competencia; las cuales se

deben de mantener. Las debilidades son lo que debes de mejorar para convertirlo en fortaleza.

Las fortalezas se clasifican en:

- **Comunes:** cuando es poseída por varias empresas o cuando varias están en capacidad de implementarla.
- **Distintivas:** cuando una misma fortaleza es poseída por un pequeño número de competidores. Son las que generan ventajas competitivas y desempeños superiores a las del promedio industrial. Son poco susceptibles de copia o imitación cuando se basan en estructuras sociales complejas, o aquellas que no pueden ser comprendidas por la competencia o cuando su desarrollo se da a través de una coyuntura única que las demás no pueden seguir.
- **De imitación:** son grandes capacidades de copiar y mejorar las fortalezas distintivas de los demás.

Las **debilidades** se refieren básicamente a desventajas competitivas, las cuales se presentan cuando no se implementan estrategias generadoras de valor que los competidores sí implementan.

Al realizar el análisis externo se deben considerar todos los elementos de la cadena productiva, aspectos demográficos, culturales, políticos e institucionales. Se deben plantear preguntas como:

- ¿En qué áreas es difícil alcanzar altos desempeños y en cuáles se podrían generar altos desempeños?
- ¿Cuáles son las barreras que impiden que este producto alcanza sus metas de participación en el mercado?

El FODA es especialmente importante para el área de marketing debido al análisis externo ya que se considera el mercado, su potencial y los aspectos sobre los cuales se podría ejercer influencia con el fin de producir recompensas para nuestras iniciativas.

CAPÍTULO III

MATERIALES Y METODOS

3.1. MATERIALES

Los materiales necesarios para el desarrollo, en todas sus etapas, del presente proyecto para la implementación del “*Sistema de Control de Software*” para la Dirección de Telemática de la Marina, son los siguientes:

3.1.1. HARDWARE

En lo referente al equipamiento Hardware para el desarrollo de la aplicación, en su mayor parte se ha hecho uso del equipamiento informático con que cuenta el Departamento de Desarrollo (encargado técnicamente para el análisis, desarrollo e implementación de aplicaciones para la Marina de Guerra del Perú) y otra parte del equipamiento informático de la Oficina de Software (órgano técnico encargado del control de software a nivel Marina de Guerra), siendo los principales los siguientes:

Equipamiento hardware de la Oficina de Software:

Nº	Equipo	Cantidad
1	Microcomputadora DeskTop (DualCore, 2 Gb Ram, Hd de 120 Gb, Lector/Quemador de DVD, pantalla crt svga de 17")	1 Unidad
2	Impresora HP1210 (multifuncional a tinta)	1 Unidad
3	Switch 3Com	1 Unidad
4	Cableado estructurado(Cable UTP Cat.6)	100 metros

Asimismo, se debe de indicar que se hará uso del siguiente equipamiento hardware en cada Dependencia y Unidad de la Marina de Guerra del Perú a fin de que la aplicación pueda ser instalada y se generen los reportes anuales que mediante Directiva se soliciten:

Nº	Equipo	Cantidad
1	Microcomputadora DeskTop (DualCore, 1 Gb Ram, Hd de 80 Gb, Lector, pantalla crt svga de 15")	264 Unidades
2	Impresora HP1210 (multifuncional a tinta)	264 Unidades
3	Switch 3Com	264 Unidad
4	Cableado estructurado(Cable UTP Cat.6)	2,640 metros

3.1.2. SOFTWARE

El software necesario para el análisis, desarrollo y posterior implementación de la Aplicación “*Sistema de Control de Software*” consta de los siguientes aplicativos:

Equipamiento software de la Oficina de Software:

Nº	Software	Cantidad
1	Windows XP licenciado	1 Licencia
2	Microsoft Office 2003	1 Licencia
3	McAfee	1 Licencia
4	Microsoft SQL Server	1 Licencia

Asimismo, se debe de indicar que se hará uso del siguiente software en cada Dependencia y Unidad de la Marina de Guerra del Perú a fin de que la aplicación pueda ser instalada y se generen los reportes anuales que mediante Directiva se soliciten:

N°	Software	Cantidad
1	Windows XP licenciado	264 Licencias
2	Microsoft Office 2003	264 Licencias
3	McAfee	264 Licencias
4	Microsoft SQL Server Cliente	264 Licencias

3.1.3. OTROS

Asimismo, para el análisis, desarrollo y posterior implementación de la aplicación antes mencionada, se han usado los siguientes materiales de escritorio y pad:

Material de escritorio y pad usado de la Oficina de Software:

N°	Material	Cantidad
1	Papel Bond A-4 de 75 g.	3 Millares
2	Fólder de Manila A-4	10 Unidades
3	Lapiceros color negro	10 Unidades
4	Cartuchos de tinta negro y color para impresora HP1210	4 Unidades
5	Pizarra acrílica de 1.5 m por 2.5 m	1 Unidad

Material de escritorio y pad, para cada Dependencia y Unidad de la Marina de Guerra del Perú:

Nº	Material	Cantidad
1	Papel Bond A-4 de 75 g.	200 hojas
2	Fólder de Manila A-4	2 Unidades
3	Lapiceros color negro	1 Unidad
4	Cartuchos de tinta negro y color para impresora HP1210	2 Unidades

3.2. MÉTODOS

3.2.1. Lugar de ejecución del proyecto

Marina de Guerra del Perú

3.2.2. Cobertura de estudio

El presente proyecto para el desarrollo e implementación del **“Sistema de Control de Software”** se ha desarrollado para la Oficina de Software perteneciente a la Dirección de Telemática de la Marina, la misma que se encuentra ubicada dentro de las instalaciones de la Estación Naval La Perla en el Distrito de La Perla, Provincia Constitucional Callao, a fin de optimizar el control de software de la Marina de Guerra del Perú, el mismo que cuenta con

diversas Dependencias y Unidades Navales a lo largo y ancho del territorio nacional por lo que el ámbito de aplicación es extenso.

3.2.3. Metodología de desarrollo

- Documento Visión

Documentos fuentes obtenidas de las Dependencias Navales y Unidades así como, de la Oficina de Software, como son: Formato de Control de Inventario, Reportes emitidos por las Dependencia y Unidades y Guía de Instalación de Software.

- Especificación de Requerimientos: Se requiere, básicamente, el desarrollo de una aplicación que permita el control en tiempo real del software y sus licencias a nivel de la Marina de Guerra del Perú.

CAPÍTULO IV

RESULTADOS

4.1. SITUACION ACTUAL DEL SISTEMA

Actualmente nuestro sistema está en la capacidad de llevar un control de la asignación de los artículos informáticos a los usuarios que trabajan en las fuerzas armadas, permitiéndonos conocer la distribución exacta de los equipos, así como también imprimir reportes para de esta manera saber a quien se asigno que artículo, cuánto tiempo permanece en el poder del usuario, registra hora y fecha y de esta manera se trata de dar un servicio eficiente con respecto al control de estos artículos.

4.2. ESTUDIO DE RIESGOS Y POSIBLES SOLUCIONES

Los supuestos o factores exógenos de riesgos son aquellas condiciones que se hallan fuera del control o influencia inmediata del proyecto, pero que son necesarias considerarlas en el presupuesto para lograr los objetivos del Proyecto. El riesgo se encuentra presente desde el inicio hasta finalizar el proyecto, hasta entonces cada miembro del equipo asume una responsabilidad para controlar el riesgo.

El manejo del riesgo del proyecto incluye los procesos que se preocupan con identificar, analizar, y responder al riesgo del proyecto. Este incluye maximizar los resultados de eventos positivos y minimizar las consecuencias de eventos adversos. Se identifican los siguientes procesos:

- **Identificación del Riesgo:** Determinar que riesgos tienen probabilidad de afectar el proyecto y documentar las características de cada uno.
- **Cuantificación del Riesgo:** Es evaluar el riesgo y las interacciones del riesgo para cuantificar el rango de posibles resultados del proyecto.
- **Desarrollo de Respuesta al Riesgo:** Es definir los pasos de mejoramiento para las oportunidades y respuestas a amenazas.
- **Control de Respuesta al Riesgo:** Es responder a cambios en el riesgo a través de la vida del proyecto.

Periodo de la evaluación: Los riesgos van a ser evaluados cada 15 días para su mejor control.

P= PROBABILIDAD

I= IMPACTO

R= RESPONSABLE

Impacto: Estará estimado de desde 1 hasta 5.

* Alto (A): Es de riesgo 5 y 4.

* Moderado (M): El riesgo es de 3.

* Bajo (B): El riesgo es 2 y 1.

RIESGOS Y SOLUCIONES

RIESGO	CONSECUENCIA	IMPACTO (1-5)	PROBABILIDAD	ACCIÓN	CONTINGENCIA
No cumplir los tiempos	Cambio de la planificación del Proyecto.	Alto – 4	75%	Reunión de emergencia para alinear planificación.	Recuperar los tiempos perdidos por las tardes.
Demora en la entrega de componentes claves	Genera sobre costos, retrasos en la programación, pagos de multas, y la entrega de un producto de menor calidad.	Alto – 5	70%	Buscar la persona o empresa, que entregue los componentes claves según lo establecido.	Informe semanales del avance de los componentes claves.

No contar con la data necesaria en los tiempos estimados	Retraso en la presentación de la base de datos.	Alto – 5	70%	Incrementar el apoyo para la búsqueda de la información requerida.	Informe semanales para su evaluación pertinente.
Nuevos requerimientos de diseño	Cambio en la planificación del proyecto.	Medio – 3	50%	Formalizar cambios establecidos en los requerimientos con acta de aceptación.	Actas de aceptación de acuerdo a los avances presentados.
Problemas externos	Retraso en la entrega del proyecto.	Alto – 4	15%	Formalización de retardo y comunicación constante del equipo de trabajo.	Realización de reunión de trabajo en casa de los miembros responsables
Falta de recursos económicos	Retraso en la entrega del proyecto.	Alto 4	60%	Establecer nuevas características de implementación de acuerdo a los recursos.	Pedir parte del pago por adelantado o realizar un préstamo.

Mal integración de los módulos.	Retraso en las etapas del proyecto donde participan.	Alto – 4	50%	Mantener una buena comunicación entre los integrantes de los módulos.	Uso de un estándar para codificar y documentar, para facilitar la integración.
Incumplimiento de la Planificación	Cambio en la planificación del proyecto y retraso en la entrega del proyecto.	Alto-4	70%	Hacer un seguimiento al desarrollo del proyecto.	Ajustar fechas de inicio y fin de tareas, para estimar el nuevo tiempo de las actividades.

4.3. MODELO ACTUAL

FORTALEZAS	DEBILIDADES
OPORTUNIDADES	AMENAZAS

Análisis FODA:

Fortaleza:

- Soporte técnico a todas las UDD
- Personal Calificado
- Conocimiento, dominio e innovación de diversos software
- Asignación presupuestal anual para el licenciamiento de software.

Debilidades:

- Escaso personal para el inventario adecuado.
- El sistema no es el adecuado (manual)
- La información que se tiene no está debidamente actualizada (no se tiene en tiempo real)
- No se cuenta con los roles definidos para el acceso a la información.
- No permite obtener reportes personalizados.

Oportunidades:

- Diversos Proveedores.
- Nuevas y mejores tecnologías.
- Implementación de equipos de última generación.
- Instalación de nuevos programas y/o nuevas versiones de las mismas.
- Brindar capacitación a los encargados de sistema fuera del área de Lima y Callao.
- Acceso a cursos gratuitos por la adquisición de software.

Amenazas:

- Pérdida de información debido a no contar con una copia de seguridad del inventario
- Cambios tecnológicos de software
- No se maneja un adecuado procedimiento para el envío de información actualizado de las UDD
- La información remitida puede ser no confiable (usuarios no brindan confiabilidad al emitir reportes)
- Los usuarios necesitan capacitación constante para el correcto llenado del formato de control de inventario de software.

4.3.1. Análisis del modelo de contexto

1. Análisis

Registrar personal: El operador le solicita sus datos personales al empleado a registrar lo que incluye ingresar los datos al sistema y validar los mismos por parte del sistema. Si los datos son válidos, el sistema guarda los datos y los muestra al operador (*apéndice 8.1*).

Registrar Artículos Informáticos:

El operador recibe la hoja de detalles de los artículos, luego, el operador ingresa los datos al sistema; si el empleado es nuevo (no se encuentra en la base de datos) el operador ingresa un nuevo empleado, si el detalle del artículo no existe el operador ingresa nuevo detalle del artículo. El operador corrige los datos, si los datos son correctos el operador valida los datos; si los datos son válidos el sistema guarda los datos en la base de datos (*apéndice 8.2*).

2. Diagrama de secuencias y colaboración

Registrar personal:

- ✓ El operador selecciona la opción del sistema de ingresar personal.
- ✓ El operador ingresa los apellidos y nombres del empleado, dirección, teléfono, fecha de nacimiento, estado civil, cargo y código de personal al sistema.
- ✓ El sistema verifica los datos ingresados al sistema.
- ✓ Los datos personales se almacenan en la base de datos.
- ✓ La base de datos se actualiza.

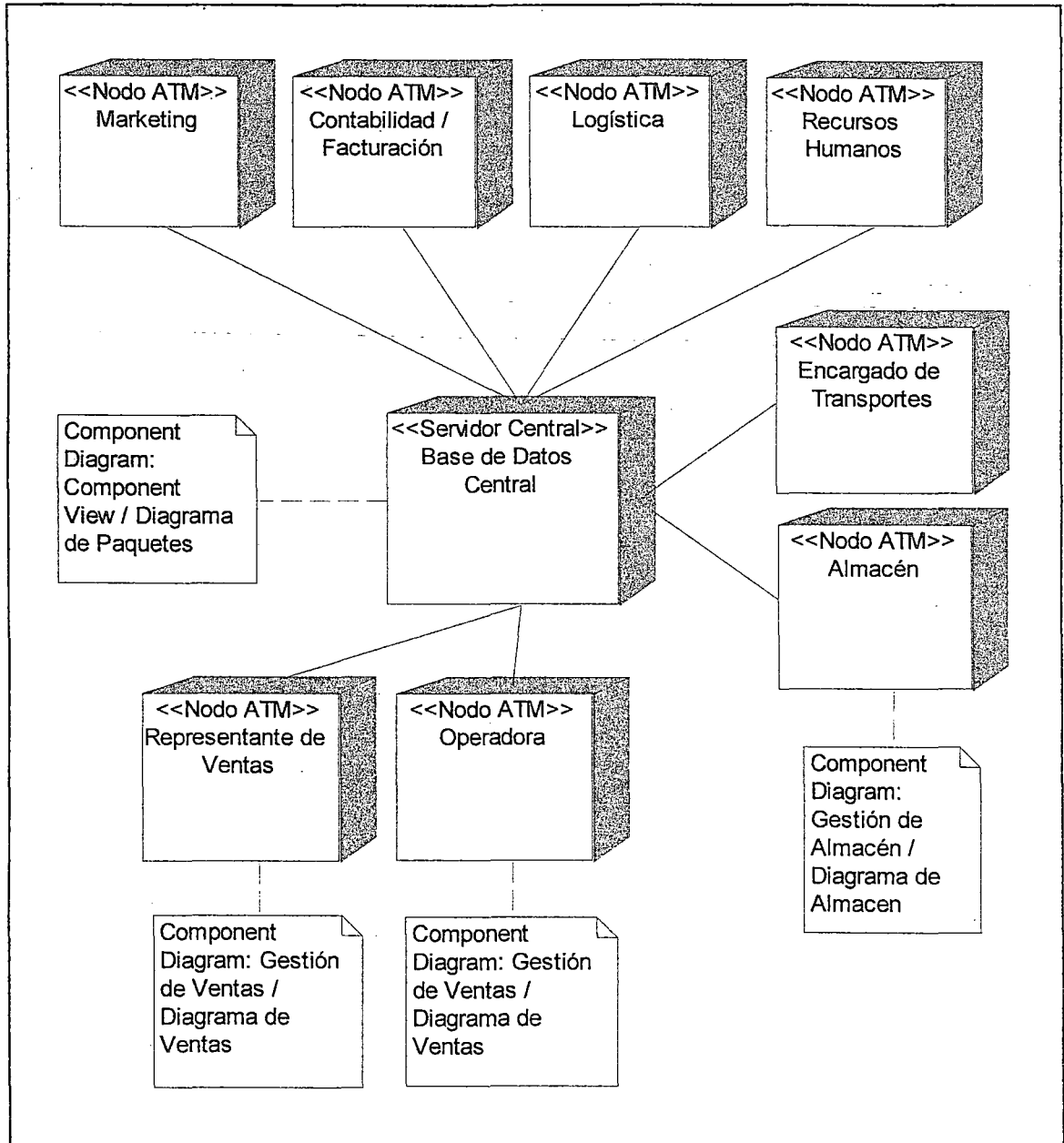
(*apéndice 8.3*)

Registrar Artículos Informáticos:

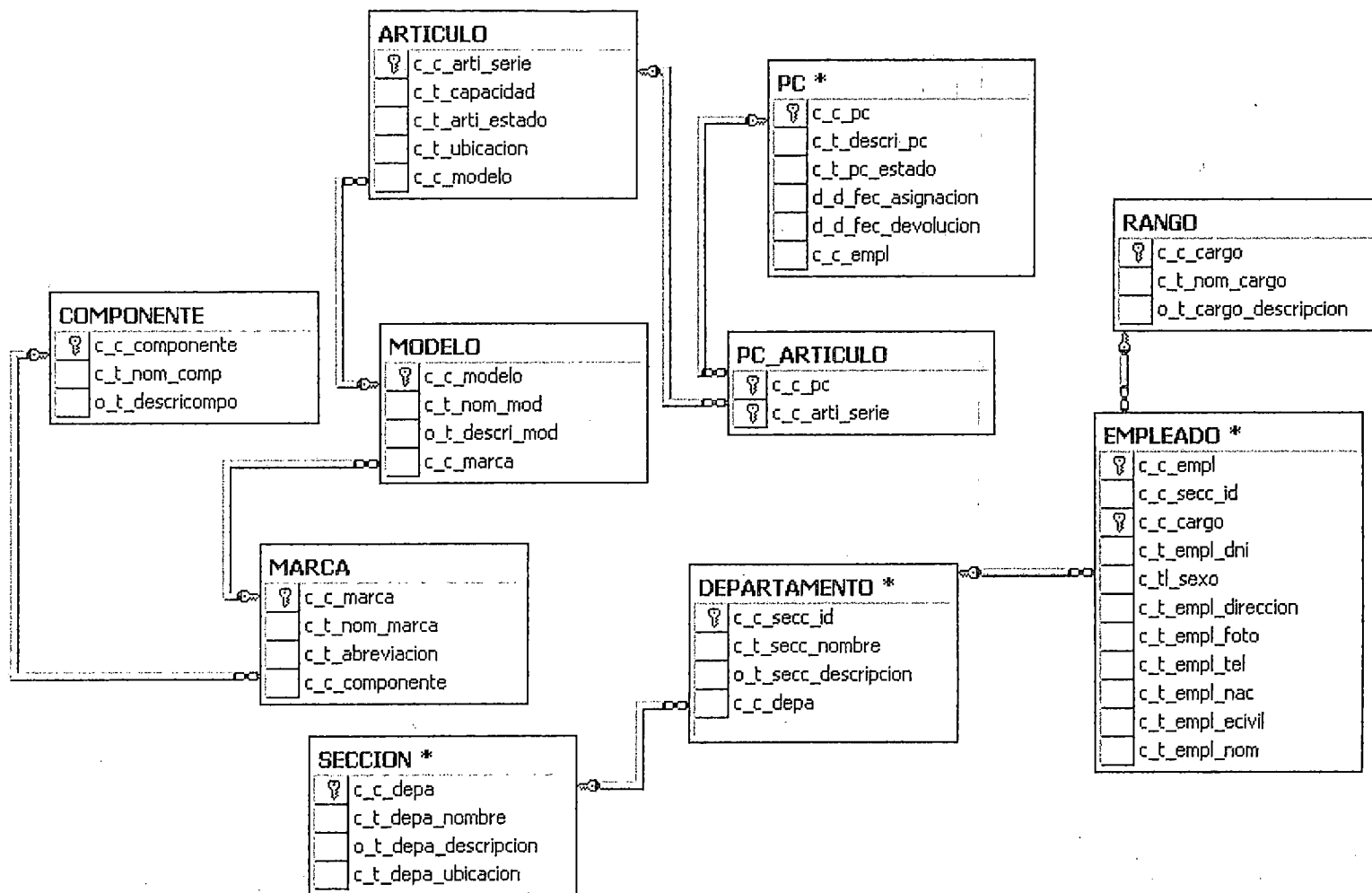
- ✓ El Operador recibe la hoja de asignación que presenta los datos de la computadora y los del usuario que se hará responsable de ella.
- ✓ Se ingresan las características del PC, y se verifica que existan todos los tipos de modelos y marcas, si no fuera el caso se deberán ingresar los modelos y marcas nuevas.
- ✓ Se ingresan los datos del usuario que va a ser responsable de la Pc.
- ✓ Finalmente se guarda la asignación de la Pc a un determinado empleado.

(apéndice 8.4)

3. Diseño Físico de datos



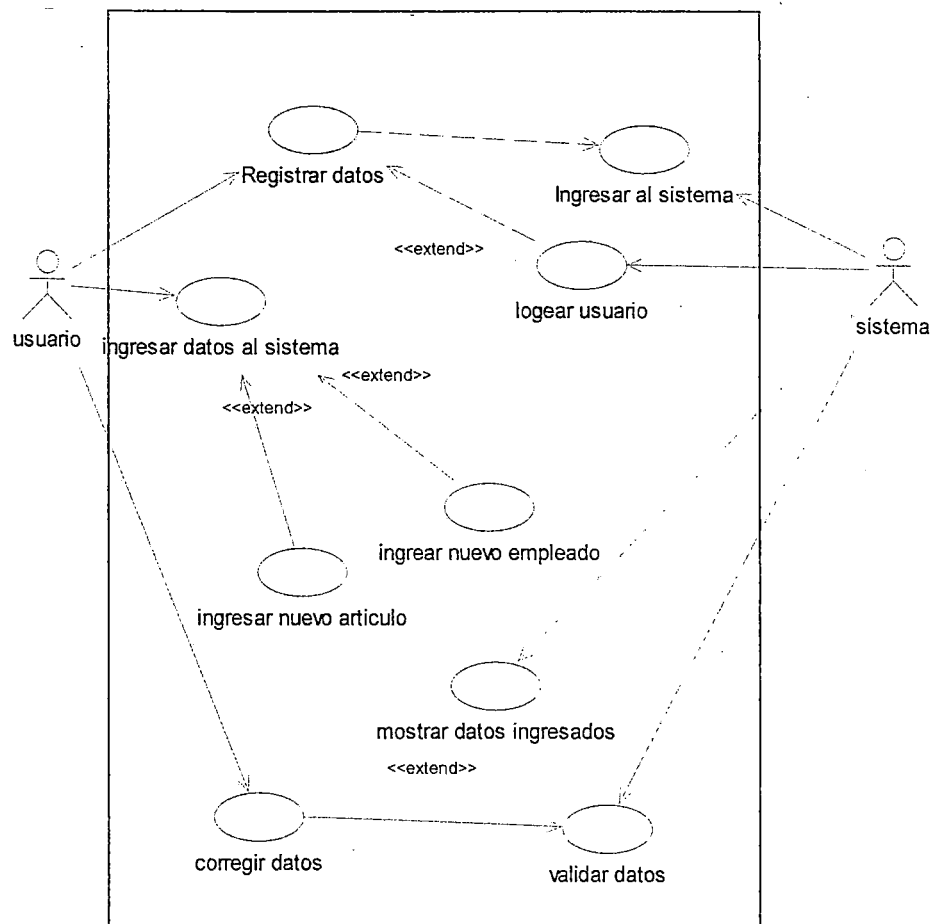
4. Diagrama de Base Datos



5. Modelamiento del Negocio


Este acápite describe las actividades que efectuará el usuario y la manera como éstas serán apoyadas por la solución.

Gestión del Usuario




DESCRIPCIÓN DE LOS CASOS DE USO

INGRESAR AL SISTEMA

1		Proyecto	Sistema de control de asignación de artículos informáticos para las fuerzas armadas
2	Caso de Uso 2	Ingresar al sistema	
3	Destinatario :	Ingeniero de Requerimientos	
4	Descripción :	El sistema necesariamente tendrá que seguir los procedimientos de este caso de uso cuando el usuario se encuentre registrado ya en la Base de Datos del Sistema de control y asignación de artículos informáticos de las FF.AA.	
5	Autores :	Mariella Maxi	
6	Actores :	Sistema	
7	Requerimientos :	Ingresar al sistema	
8	Precondiciones :	El usuario se encuentra registrado en la Base de Datos del Sistema.	
9	Pos-condiciones :	Se habilitan todas las interfaces para realizar las compras.	
10	Flujo Principal :	Paso	Acción
		1	El usuario inicia el proceso de ingreso de datos (ID, password) haciendo clic en el botón registrar.
		2	El sistema captura la información y compara la BD con los datos ingresados.
		3	El sistema habilita todas las ventanas y permisos para el usuario.


11	Excepciones:	1	Si los datos ingresados no son correctos el sistema pedirá nuevamente el ingreso de los datos también recomendará el registro de nuevo usuario.
12	Relaciones:		
13	prioridad		Imprescindible
14	clases		Usuario
15	tablas		Usuario
16	datos		Datos del usuario
17	métodos		Ingresar datos
18	interfaz		Ingresar al sistema

LOGUEAR USUARIO

1		Proyecto	Sistema de control de asignación de artículos informáticos para las fuerzas armadas
2	Caso de Uso 3		Loguear usuario
3	Destinatario :	Ingeniero de Requerimientos	
4	Descripción :	El sistema necesariamente tendrá que seguir los procedimientos de este caso de uso cuando el usuario se encuentre registrado ya en la Base de Datos del Sistema de control y asignación de artículos informáticos de las FF.AA	
5	Autores :	Paola Calderón	
6	Actores :	Sistema	
7	Requerimientos :	Loguear usuario	
8	Precondiciones :	El usuario ya se encuentra registrado	
9	Pos-condiciones :	Puede seleccionar los productos que el desee	

10	Flujo Principal :	Paso	Acción
		1	El sistema muestra la interfaz de login al usuario.
		2	El usuario ingresa su nombre y password.
		3	El sistema informa al usuario que el proceso ha terminado con éxito.
11	Excepciones	1	Si no ha ingresado sus datos correctamente en tres veces , irá a la opción ayuda
12	Relaciones:		
13	prioridad		Imprescindible
14	clases		Usuario
15	tablas		Usuario
16	datos		Login del usuario
17	métodos		Ingresar login , verificar login
18	interfaz		Interfaz de login de usuario


INGRESAR DATOS AL SISTEMA

1		Proyecto	Sistema de control de asignación de artículos informáticos para las fuerzas armadas
2	Caso de Uso 4		Ingresar datos al sistema
3	Destinatario :		Ingeniero de requerimientos

4	Descripción :	El sistema necesariamente tendrá que seguir los procedimientos de este caso de uso cuando el usuario requiera asignar algún implemento en el Sistema de control de asignación de artículos informáticos para las fuerzas armadas	
5	Autores :	Mirian Rodas	
6	Actores :	Usuario	
7	Requerimientos :	Seleccionar Artículo Deportivo	
8	Precondiciones :	El usuario está registrado en la Base de Datos.	
9	Pos-condiciones :	Se muestran los registros anteriores, además de las opciones de ingreso.	
10	Flujo Principal :	Paso	Acción
		1	El usuario elige el proceso de seleccionar implementos.
		2	El usuario selecciona los implementos que sean de su interés
		3	El usuario confirma su interés por adquirir los implementos seleccionados.
		4	El sistema informa la conformidad de la selección y reporta las condiciones totales de las selecciones.
11	Excepciones	1	Si no se selecciona ningún implemento el sistema volverá a pedir la selección.
12	Relaciones:	extend	ingresar nuevo articulo
		extend	ingresar nuevo empleado
13	prioridad	Imprescindible	
14	clases	artículos informáticos	
15	tablas	Artículos	


16	datos	Datos del usuario
17	métodos	Seleccionar artículo
18	interfaz	Interface que muestra seleccionar artículos

INGRESAR ARTICULO INFORMATICO

1		Proyecto	Sistema de control de asignación de artículos informáticos para las fuerzas armadas
2	Caso de Uso 5	Ingresar artículo informático	
3	Destinatario :	Ingeniero de Requerimientos	
4	Descripción :	El sistema necesariamente tendrá que seguir los procedimientos de este caso de uso cuando el usuario quiera agregar algún artículo previamente seleccionada en el Sistema de control de asignación de artículos informáticos para las fuerzas armadas	
5	Autores :	Christian Cuadros	
6	Actores :	Usuario, sistema	
7	Requerimientos :	Deseleccionar producto	
8	Precondiciones :	El usuario ha seleccionado artículo	
9	Pos-condiciones :	El producto se a retirado de la muestra	
10	Flujo Principal :	Paso	Acción
		1	El usuario deselecciona los artículos que aparecen en el reporte generado por el Sistema.
		2	El usuario elige la opción actualizar selección.


		3	El sistema retira los artículos deseleccionados del reporte y actualiza un nuevo reporte.
11	Excepciones	1	No se puede deseleccionar si no hay selección.
12	Relaciones:		
13	prioridad		Imprescindible
14	clases		Artículos informáticos
15	tablas		Artículos
16	datos		Datos de Artículos
17	métodos		Deseleccionar Artículos
18	interfaz		Deseleccionar Artículos

INGRESAR NUEVO PERSONAL

1		Proyecto	Sistema de control de asignación de artículos informáticos para las fuerzas armadas
2	Caso de Uso 6	Ingresar nuevo personal	
3	Destinatario :	Ingeniero de Requerimientos	
4	Descripción :	El sistema necesariamente tendrá que seguir los procedimientos de este caso de uso cuando el usuario quiera agregar algún empleado previamente seleccionada en el Sistema de control de asignación de artículos informáticos para las fuerzas armadas	
5	Autores :	Mariella Maxi	
6	Actores :	Usuario, sistema	


7	Requerimientos :	Deseleccionar producto	
8	Precondiciones :	El usuario ha seleccionado artículo	
9	Pos-condiciones :	El producto se a retirado de la muestra	
10	Flujo Principal :	Paso	Acción
		1	El usuario deselectiona los empleados que aparecen en el repòrte generado por el Sistema.
		2	El usuario elige la opción actualizar selección.
		3	El sistema retira los empleados deselectionados del repòrte y actualiza un nuevo repòrte.
11	Excepciones	1	No se puede deselectionar si no hay selección.
12	Relaciones:		
13	prioridad	Imprescindible	
14	clases	Artículos informáticos	
15	tablas	Empleados	
16	datos	Datos de Empleados	
17	métodos	Deseleccionar Empleados	
18	interfaz	Deseleccionar Empleados	

MOSTRAR DATOS

1		Proyecto	Sistema de control de asignación de artículos informáticos para las fuerzas armadas
2	Caso de Uso 7	Mostrar datos	
3	Destinatario :	Ingeniero de Requerimientos	


4	Descripción :	El sistema necesariamente tendrá que seguir los procedimientos de este caso de uso cuando el usuario haya concluido el ingreso de datos en el Sistema de control de asignación de artículos informáticos para las fuerzas armadas	
5	Autores :	Christian Cuadros	
6	Actores :	Sistema, Usuario	
7	Requerimientos :	Realizar pedido	
8	Precondiciones :	El usuario selecciona los artículos correctamente.	
9	Pos-condiciones :	Muestra de un reporte de los artículos seleccionados más las posibilidades de pago.	
10	Flujo Principal :	Paso	Acción
		1	El usuario elige la opción de realizar pedido con lo que ya ha seleccionado.
		2	El sistema captura la información.
		3	El sistema muestra un reporte completo de los artículos seleccionados, muestra las opciones de compra de los implementos.
11	Excepciones:	1	Si no hay ningún artículo en la selección, no se muestran las opciones de compra.
12	Relaciones:		
13	prioridad	Imprescindible	
14	Clases	Artículos informáticos	
15	Tablas	Artículos	
16	Datos	Datos artículos, cantidad de artículos	
17	métodos	Almacenar artículos	
18	interfaz	Almacenamiento de datos	

CORREGIR DATOS

1		Proyecto	Sistema de control de asignación de artículos informáticos para las fuerzas armadas
2	Caso de Uso 8	Corregir Datos	
3	Destinatario :	Ingeniero de Requerimientos	
4	Descripción :	El sistema necesariamente tendrá que seguir los procedimientos de este caso de uso cuando el usuario haya concluido el ingreso de artículos en el Sistema de control de asignación de artículos informáticos para las fuerzas armadas	
5	Autores :	Paola Calderón	
6	Actores :	Usuario	
7	Requerimientos :	El usuario haya seleccionado por lo menos un artículo.	
8	Precondiciones :	Selección correcta de los pedidos de artículos deportivos.	
9	Pos-condiciones :	Notificación al correo del usuario con toda la información acerca del registro.	
10	Flujo Principal :	Paso	Acción
		1	El usuario selecciona la opción corregir.
		2	El sistema muestra la totalidad de opciones.
11	Excepciones	1	No se muestra la opción de agregar si no se ha elegido ningún artículo.
12	Relaciones:		
13	prioridad	Imprescindible	
14	clases	Artículos informáticos	
15	tablas	Artículos	

16	datos	Datos Artículos
17	métodos	Ingresar datos de Artículos
18	interfaz	Pagar Artículos

VALIDAR DATOS

1		Proyecto	Sistema de control de asignación de artículos informáticos para las fuerzas armadas
2	Caso de Uso 9.	Validar Datos	
3	Destinatario :	Ingeniero de Requerimientos	
4	Descripción :	El sistema necesariamente tendrá que seguir los procedimientos de este caso de uso cuando el usuario haya seleccionado la opción Validar del Sistema de control de asignación de artículos informáticos para las fuerzas armadas	
5	Autores :	Mirian Rodas	
6	Actores :	Usuario, sistema	
7	Requerimientos :	Pago a la entrega del producto	
8	Precondiciones :	El usuario elige la opción comprar mediante pago en entrega.	
9	Pos-condiciones :	Entrega del y cobro del producto en el domicilio con confirmación del cargo.	
10	Flujo Principal :	Paso	Acción
		1	El usuario elige la opción de validar.
		2	El sistema almacena la información en la base de datos.

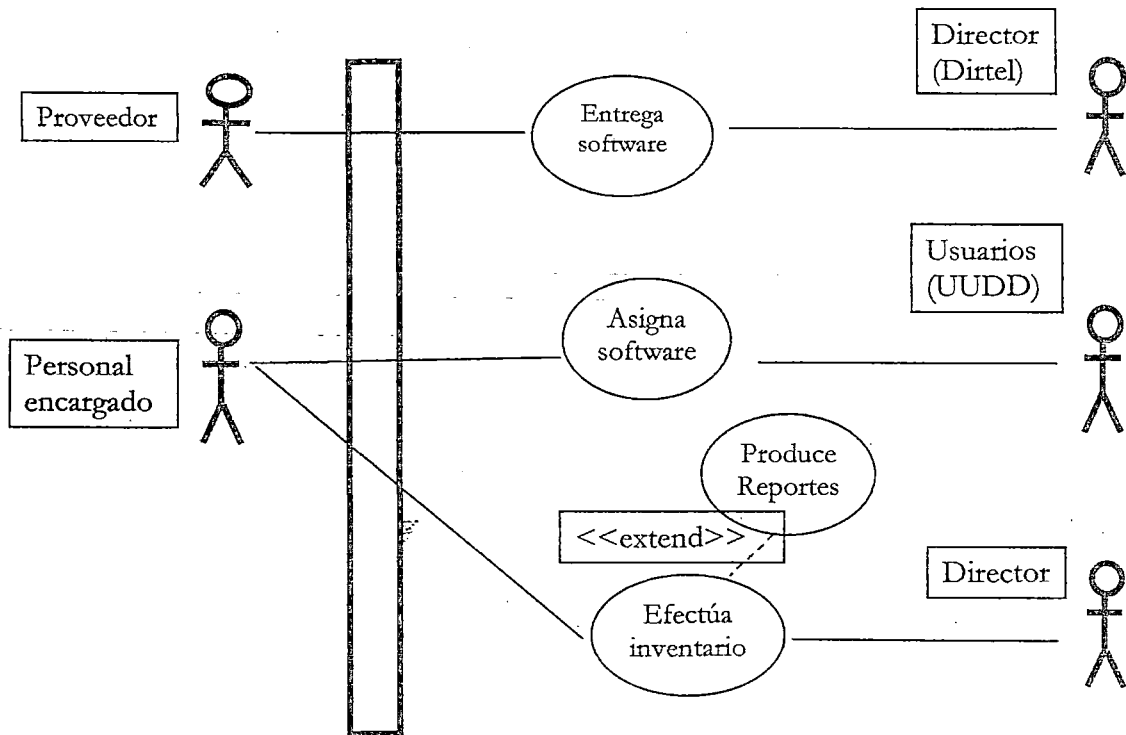
		3	El sistema genera un documento hacia los administradores del sistema.
		4	El sistema informa al usuario que el proceso ha terminado con éxito.
11	Relaciones:		
12	prioridad		Imprescindible
13	clases		Sistema
14	tablas		Sistema
15	datos		Datos usuario

Casos de Uso existentes:

- Entrega de productos, que realiza el proveedor a la Dirección de Telemática de la Marina – Oficina de Software.
- La asignación de productos (software) a las Unidades y Dependencias.
- Inventario y emisión de reporte del software.

Elaboración:

- Diagramas de casos de uso:

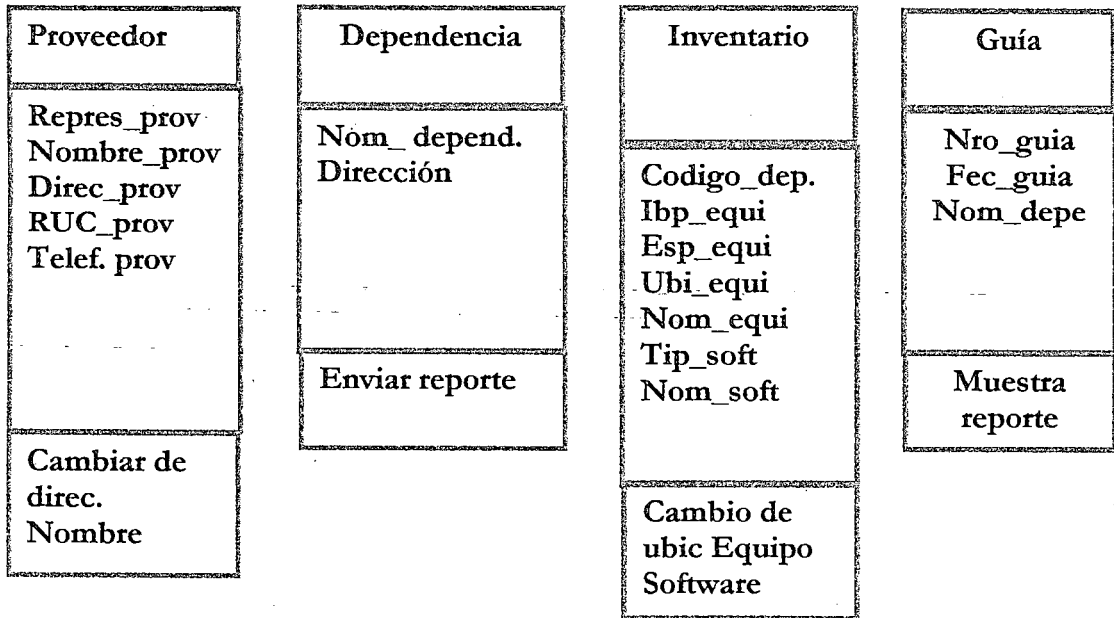


Construcción:

- Documento Arquitectura que trabaja con las siguientes vistas:
 - Vista lógica
 - Vista de implementación

Vista Lógica:

- Diagrama de clases

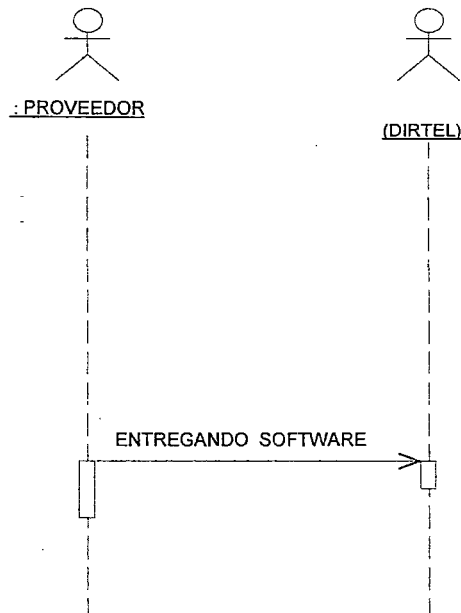


Vista de Implementación:

- Diagrama de Secuencia

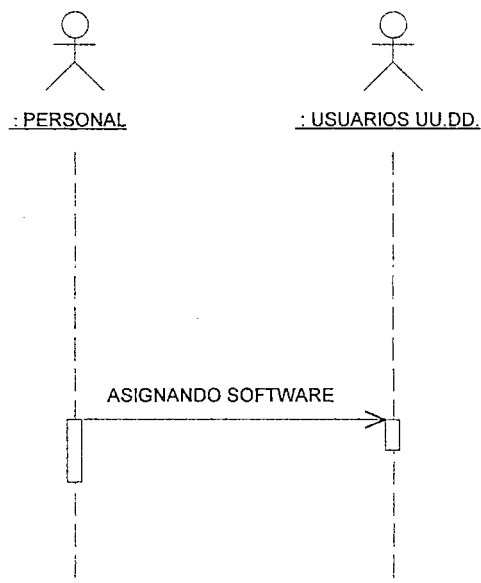
Caso de Uso N° 1: Diagrama de Secuencia

Entrega de Software

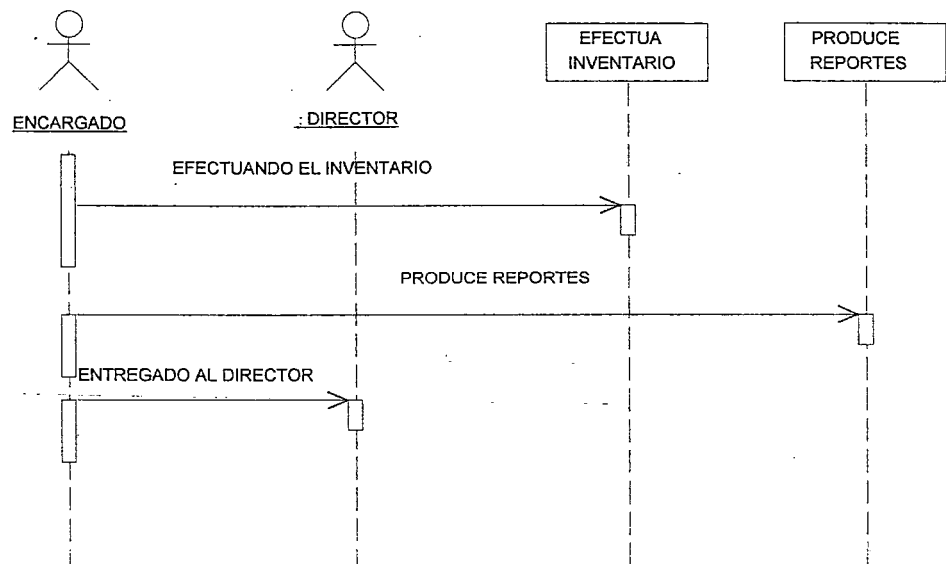


Caso de Uso N° 2: Diagrama de Secuencias

Asignación de Software

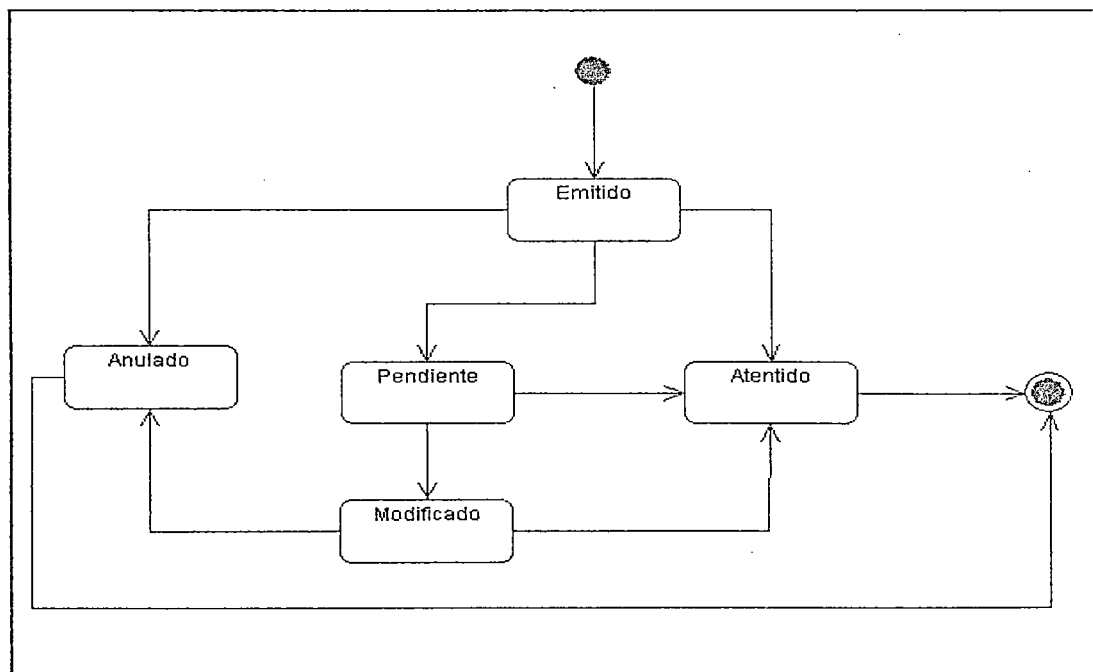


Caso de Uso N° 3: Diagrama de Secuencias
Inventario y Emisión de Reporte del Software

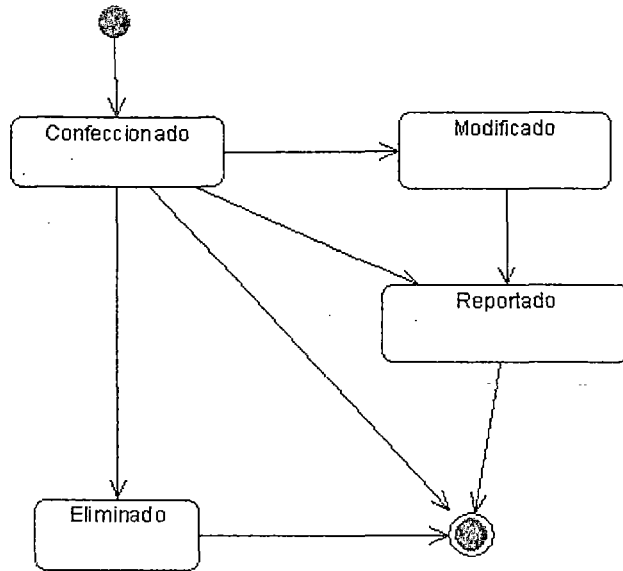


- Diagrama de estados

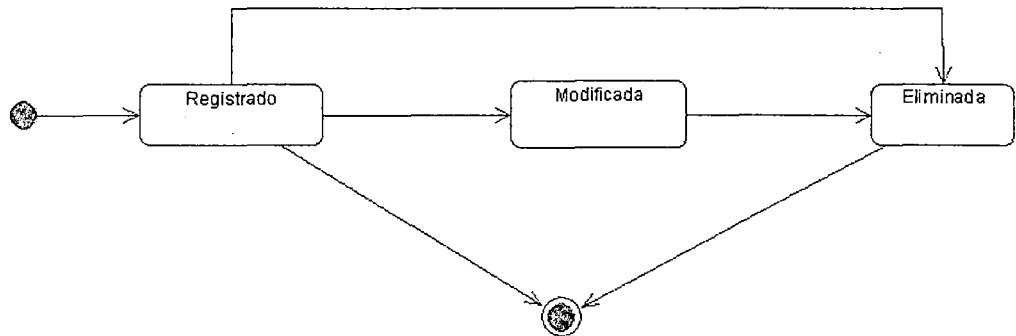
Guía



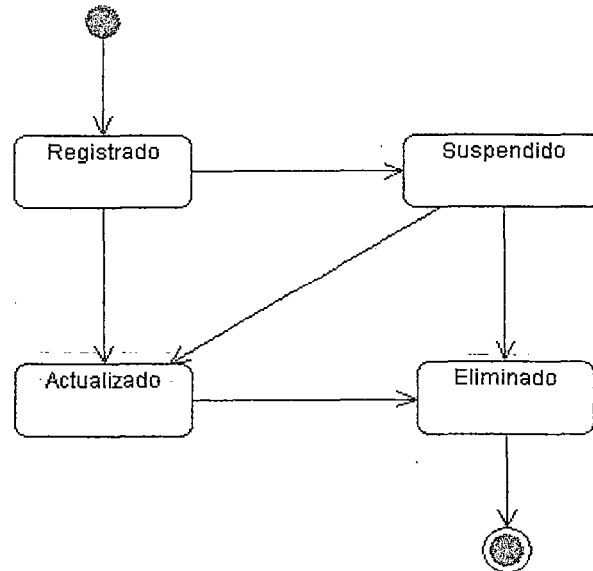
Inventario



Dependencia



Proveedor



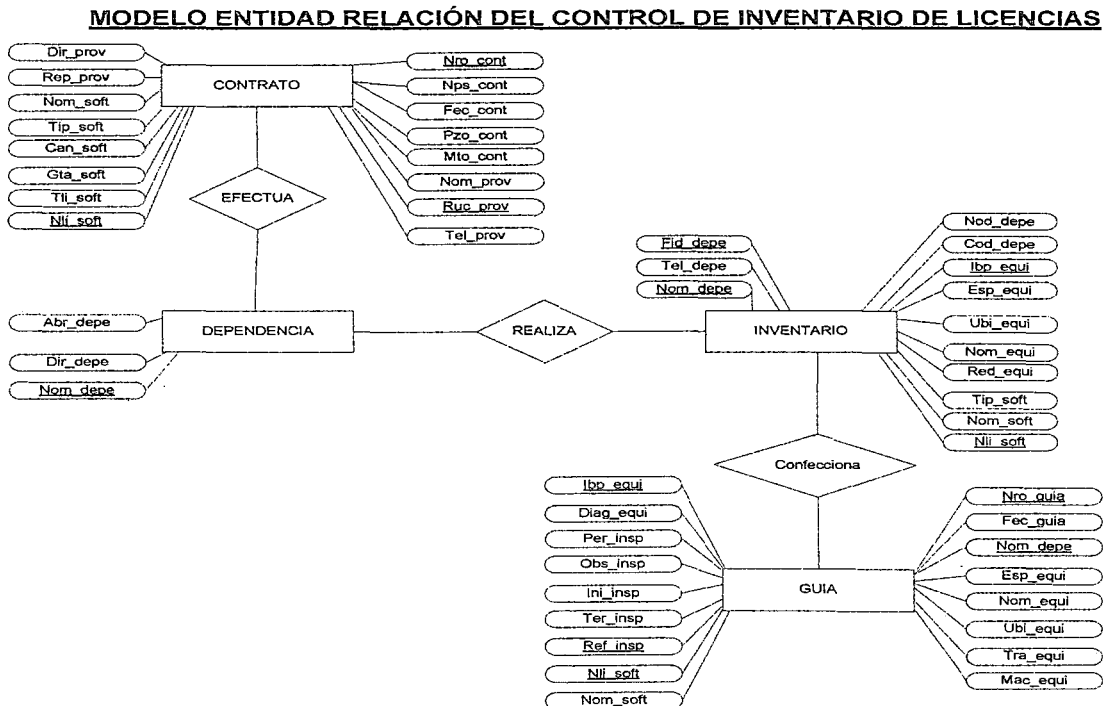
4.3.2. Análisis del proceso

Actualmente el control de inventario de software de la Marina de Guerra del Perú, se realiza manualmente, de la que se deduce que su actualización u obtención de reportes para la toma de decisiones se obtiene con información desfasada y en forma extemporánea, razón por el cual dicha información no es confiable, debido a esta problemática se ha considerado por conveniente que se realice un estudio, la misma que debe de concluir con un proyecto a fin de desarrollar y posteriormente implementar una aplicación capacitando a los operadores de las diferentes Dependencias y Unidades de la Marina de Guerra del Perú.

4.3.3. Análisis Normativo y regulatorio

- a) Reglamento de Doctrina de Informática de la Marina de Guerra del Perú (DOCINFO).
- b) Reglamento de contrataciones y Adquisiciones del Estado
- c) Directivas emitidas por el CONSUCODE
- d) Reglamento de Materiales de la Marina de Guerra del Perú (RECAMAR)
- e) Legislación Nacional e Internacional sobre derechos de autor y Propiedad Intelectual

4.3.4 Modelo Entidad Relación



4.4. Identificación de mejoras (aplicación de la metodología RUP)

El proceso de desarrollo RUP (Rational Unified Process) aplica varias de las mejores prácticas en el desarrollo moderno de software en una forma que se adapta a un amplio rango de proyectos y de organizaciones.

Provee a cada miembro del equipo, un fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas del desarrollo de software. Esta metodología permite que todos los integrantes de un equipo de trabajo, conozcan y compartan el proceso de desarrollo, una base de conocimientos y los distintos modelos de cómo desarrollar el software utilizando un lenguaje de modelado común: UML.

El RUP es un proceso de desarrollo de software:

Provee un enfoque estructurado para realizar tareas y responsabilidades en una organización de desarrollo. Su principal objetivo es asegurar la producción de software de alta calidad, que cumpla las necesidades de sus usuarios finales, que sea realizado en las fechas acordadas y con el presupuesto disponible.

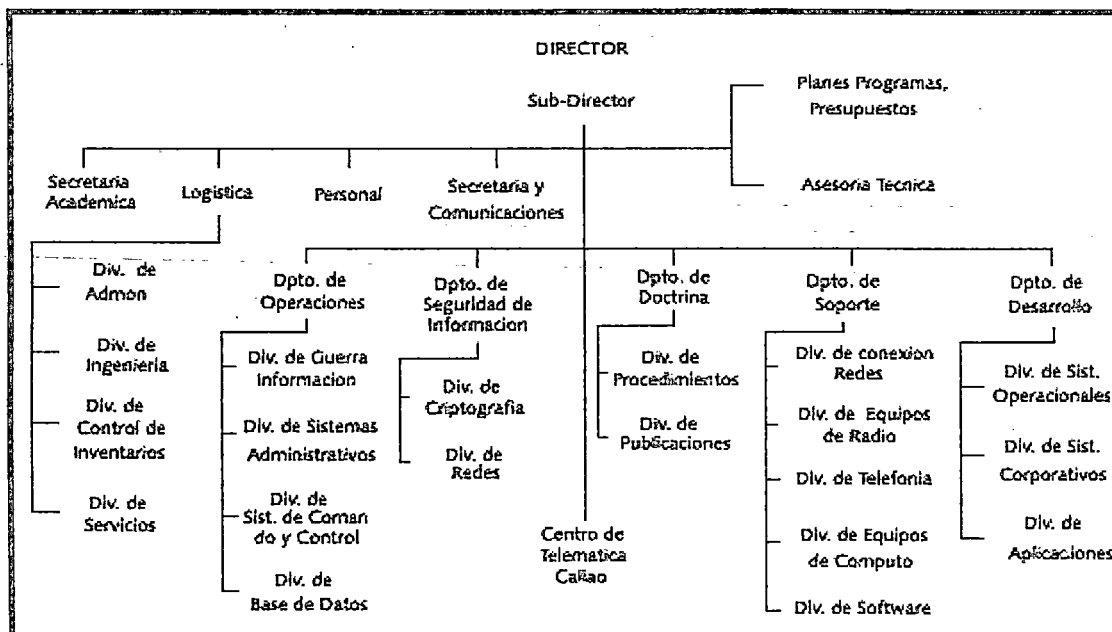
4.4.1. Mejora de procesos

Otras características o ventajas de la aplicación de esta metodología son las siguientes:

- Reconoce que las necesidades del usuario y sus requerimientos no se pueden definir completamente al principio
- Permite evaluar tempranamente los riesgos en lugar de descubrir problemas en la integración final del sistema
- Reduce el costo del riesgo a los costos de un solo incremento
- Acelera el ritmo del esfuerzo de desarrollo en su totalidad debido a que los desarrolladores trabajan para obtener resultados claros a corto plazo
- Distribuye la carga de trabajo a lo largo del tiempo del proyecto ya que todas las disciplinas colaboran en cada iteración.
- Facilita la reutilización del código teniendo en cuenta que se realizan revisiones en las primeras iteraciones lo cual además permite que se aprecien oportunidades de mejoras en el diseño

4.4.2. Reestructuración organizacional

ORGANIGRAMA DE LA DIRECCION DE TELEMATICA DE LA MARINA



4.4.3. Priorización de las mejoras

El desarrollo y posterior implementación de éste sistema para control de software permitiría que en poco tiempo La Marina de Guerra del Perú pueda cumplir con las disposiciones internacionales y nacionales sobre propiedad intelectual -derechos de autor evitando problemas legales sobre piratería o uso ilegal de software, pueda contar con información precisa y en tiempo real sobre la cantidad de software, tipo, faltante y así poder programar su adquisición y/o licenciamiento con la debida anticipación y dando la debida prioridad a los usuarios finales que lo requieran.

4.5. Resultados

INGRESO DE DATOS AL SISTEMA			
Numero de Contrato	001	Numero de proceso de seleccion	012-2000
Fecha de suscripcion de contrato	12/05/2000	Plazo de Entrega de Bienes	15/06/2001
Monto total del contrato	1250.00	Direccion de Proveedor	Av. Faucett 188
Razon Social del Proveedor	Av. Canta Callao 123	Numero de Licencia	15658622
RUC proveedor	45385967425	Tipo de Licencia	Temporal
Telefono de Proveedor	6148592	Garantia de Software	1000
Representante legal del proveedor	Jose Martinez Olaya		
Nombre del Software	Windows 2000		
Tipo de Software (aplicaciones, servidor)	Aplicaciones		
Cantidad de Software	125		

CAPÍTULO V

ANÁLISIS COMPARATIVO

5.1. Reducción de tiempos

Con la implementación del nuevo sistema se ha logrado un óptimo rendimiento en la reducción de los tiempos en los procesos del sistema.

Emisión de Reportes

Reporte	Sin la aplicación	Con la Aplicación
Consolidado Anual del Inventario de Software	6 Meses	1 día
Actualización de Inventario	3 Meses	2 días
Reportes al Alto Mando Naval	1 mes	1 día
Reportes a los Comités de Adquisiciones del Licenciamiento de Software	1 Mes	2 días

5.2. Reducción de documentos.

Durante el manejo del sistema tradicional sin la aplicación, la Oficina de Software emitía los siguientes documentos:

- Guías de Remisión de Software
- Formato de Control de Inventario de Software
- Mensajes Navales para acelerar la instalación, entrega, novedades, etc.
- Oficios, Memorándum, Directivas, etc.

Actualmente con la Implementación y puesta en funcionamiento del nuevo sistema a nivel institucional los trámites documentarios han disminuido al mínimo, es decir solo se emiten Mensajes Navales y Guías de Remisión de Software.

5.3. Análisis de costo beneficio

5.3.1 Costos

Los costos para el desarrollo, pruebas e implementación del nuevo sistema son mínimos, debido a que para la etapa de desarrollo se ha hecho uso de equipos pertenecientes a la Institución así mismo ha sido desarrollado por Personal Naval, para las pruebas e Implementación cada Unidad y

Dependencia de la Marina cuenta con equipos propios los cuales no son de uso exclusivo.

5.3.2 Ahorro

Se ha optimizado el tiempo para la emisión de los diversos reportes conllevando a que el personal disponga de más tiempo para realizar funciones inherentes a su cargo.

Asimismo, se ha conseguido ahorrar dinero al reducir los trámites documentarios, por que las actualizaciones, reportes y otros se producen en tiempo real, por cada usuario indistintamente por causa de su ubicación.

Resumen del costo beneficio

Actualmente se cuenta con un sistema que nos brinda el servicio de controlar el inventario de Software, ahorrando tiempo, personal y otros.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

Con el desarrollo de este nuevo Sistema se podrán efectuar un debido control de registro y emisión de reporte de inventario de software siendo efectuado en el menor tiempo posible a requerimiento del alto mando naval y los Comités de Licenciamiento para que puedan tener una información precisa y oportuna a fin de proyectar los nuevos licenciamientos, minimizando gastos excesivos de cualquier tipo, permitiendo estar acorde con la tecnología actual en la Institución. El sistema a cargo de la Oficina de Software de la Dirección de Telemática, implementará e instalará en las oficinas encargadas del control de todas las Unidades y Dependencias de la Institución.

6.2. RECOMENDACIONES

El Sistema de inventario de Software debería estar conectado con el Sistema de Inventario de Bienes Patrimoniales (IBP), ya que en este sistema se tiene el consolidado de todo el material informático (hardware y software) de la Marina de Guerra del Perú permitiendo con la unión de estos sistemas un control total, exacto y oportuno a requerimiento del Alto Mando Naval, recomendando la implementación de nuevos equipos para el uso de este sistema.

CAPÍTULO VII

BIBLIOGRAFÍA

- BURCH, John & GRUDNITSKI, Gary. (1996). *Diseño de sistemas de información*. México: Noriega. 985p.
- CHAVARRIA, Marcela y VILLALOBOS, Marveya. (1993). *Orientaciones para la elaboración y presentación de tesis*. México: Trillas. 115 p.
- DEXTRE, José. (1992). *Análisis del planeamiento estratégico de los sistemas de información como instrumento de delimitación de sistemas estables y optimización de la inversión en informática*. Tesis. Lima, Perú, Universidad Nacional de Ingeniería, Facultad de Ingeniería Industrial y Sistemas.
- GILL, Harjinder & RAO, Prakash. (1996). *Data Warehousing; la integración de la información para la toma de decisiones*. México: Prentice Hall Hispanoamericana. 382p.
- JACOBSON, Ivar. BOOCH, Grady & RUMBAUGH, James [JBR99]. (1999). *Unified Software Development Process. 1st Edition*. Massachusetts: Addison-Wesley.
- KENDALL, Kenneth & KENDALL, Julie. (1997). *Análisis y diseño de sistemas. 3ra. ed.* México: Prentice Hall. 913p.
- KRUCHTEN Philippe [KRU00]. (2000). *The Rational Unified Process: an introduction. 2da ed.* Addison-Wesley.
- LUCAS, Henry. (1984). *Sistemas de información: Análisis, diseño, puesta a punto*. Madrid: Paraninfo. 515p.

- MARTIN, James (1993). *Principles of object-oriented analysis and design*. Englewood Cliffs: Prentice Hall. 400p.
- MARTIN, James. (1989). *Information Engineering, BookI: Introduction*. Englewood Cliffs: Prentice Hall.
- MARTIN, James. (1990). *Information Engineering, BookII: Planning and Analysis*. Englewood Cliffs: Prentice Hall.
- MELENDEZ, Rolando. (1996). *Plan integral de sistemas de información para el fondo de compensación y desarrollo social*. Tesis. Lima, Perú, Universidad Nacional de Ingeniería, Facultad de Ingeniería Industrial y Sistemas.
- NAVARRO, Luis. (1994). *Sistemas de información para el planeamiento y control de inventarios en una empresa telefónica*. Tesis. Lima, Perú, Universidad Nacional de Ingeniería, Facultad de Ingeniería Industrial y Sistemas..
- RIVERA, René. (1998). *Sistema de información gerencial en una empresa de producción utilizando la metodología orientada a objetos*. Tesis. Lima, Perú, Universidad Nacional de Ingeniería, Facultad de Ingeniería Industrial y Sistemas.
- RUIZ, Vilca. (1991). *Sistema de información, acreditación y recaudación*. Tesis. Lima, Perú, Universidad San Martín de Porres, Facultad de Ingeniería de Computación y Sistemas. 42p.
- SENN, James. (1990). *Sistemas de información para la administración*. 3ra. ed. México: Grupo Editorial Iberoamérica. 728p.
- SQUIRE, Enid. (1995). *Introducción al diseño de sistemas*. México: Alfaomega. 345p.

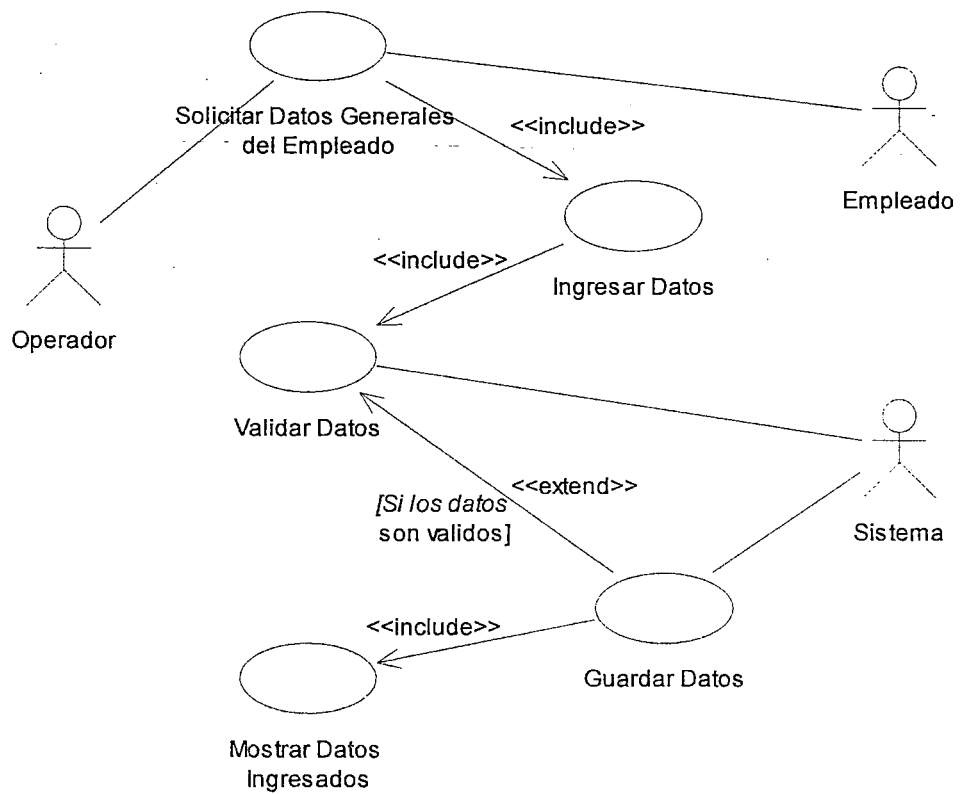
- TAMAYO Y TAMAYO, Mario. (1985). *Metodología formal de la investigación científica*. México: Limusa.
- WHITEN, Jeffrey. BENTLEY, Lonnie & BARLOW, Victor. (1996). *Análisis y diseño de sistemas de información*. 2da. ed. España: Irwin. 907p.

CAPÍTULO VIII

APÉNDICES

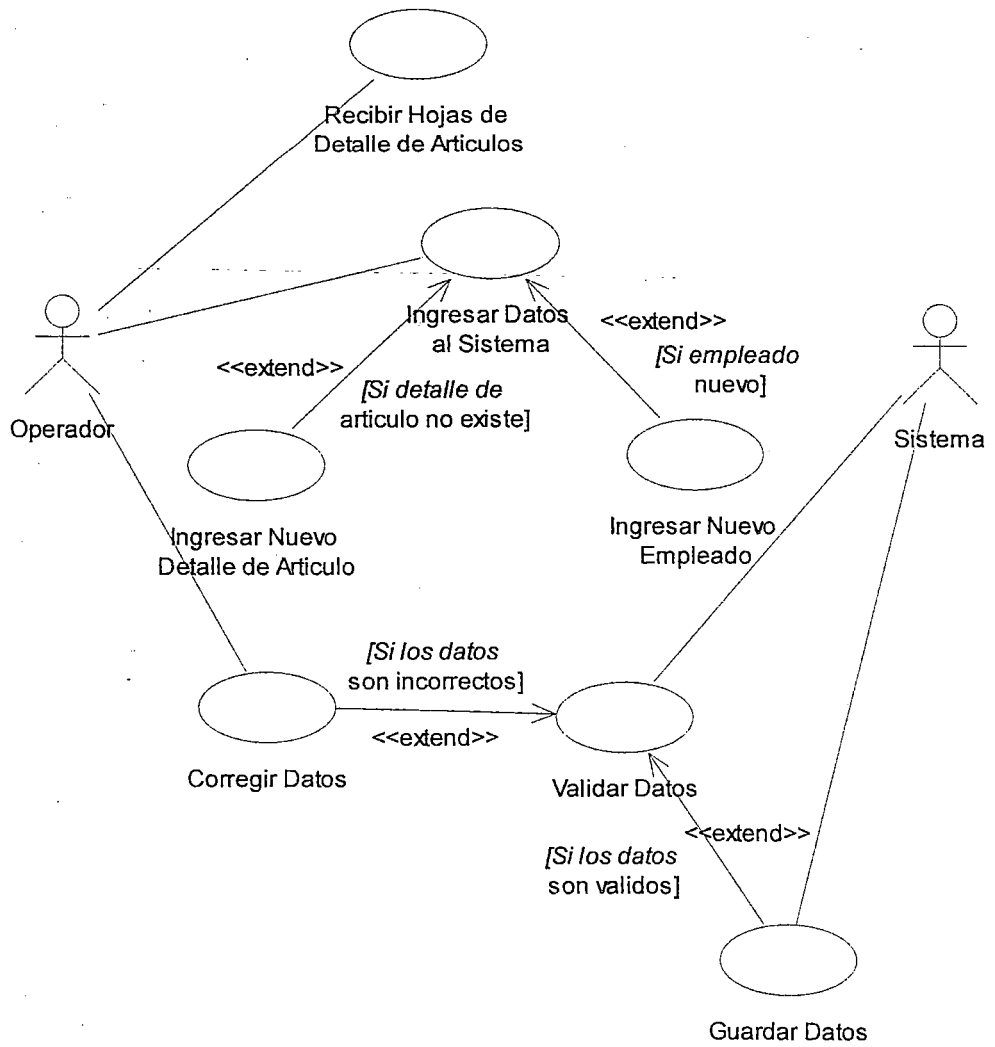
APÉNDICE 8.1

Registrar personal



APÉNDICE 8.2

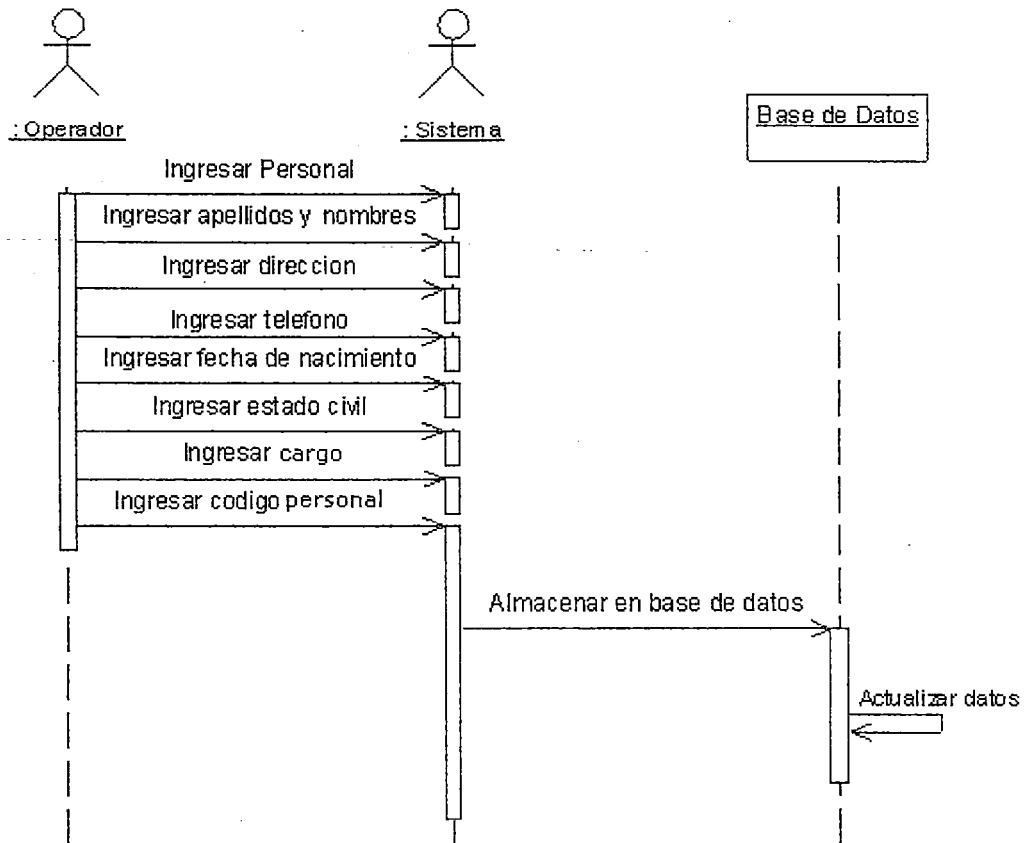
Registrar artículos informáticos

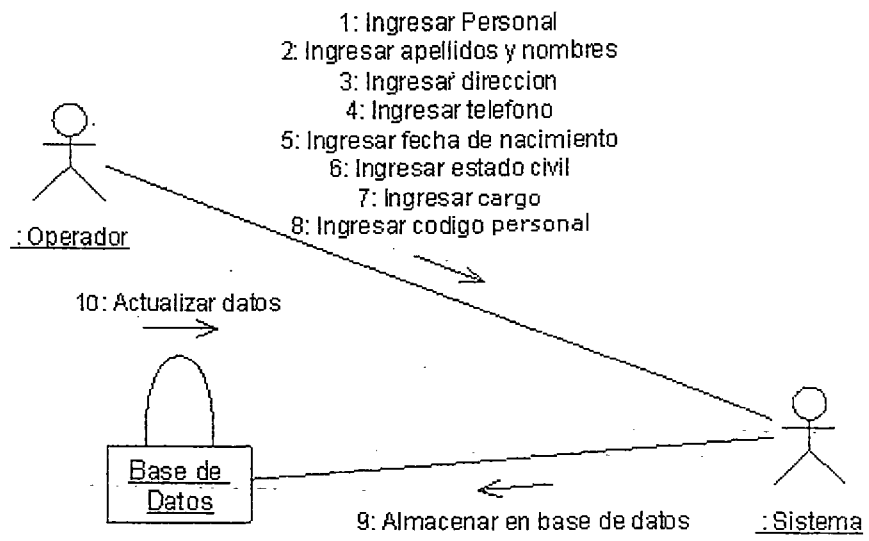


APÉNDICE 8.3

Diagramas de secuencias y colaboración

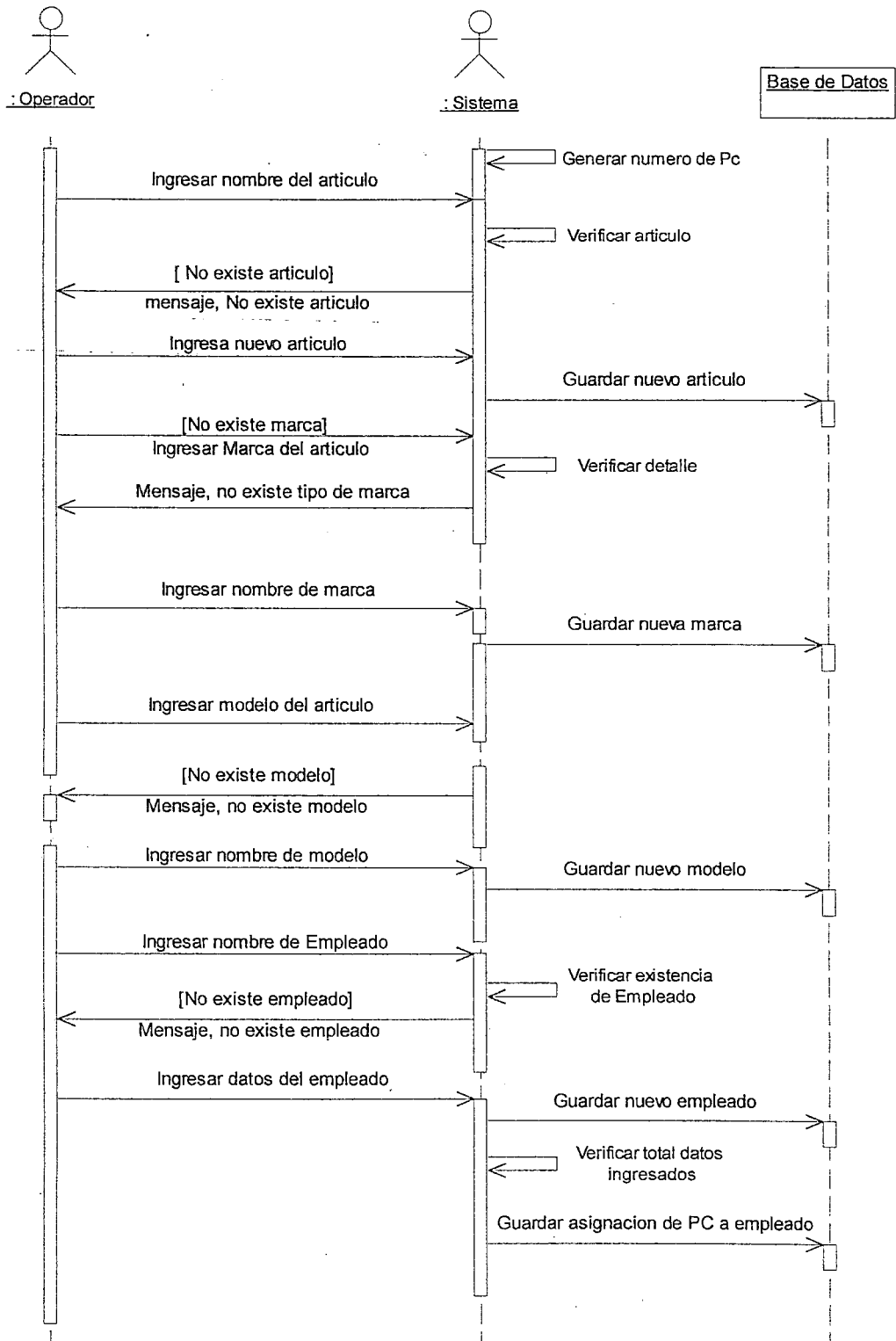
Registrar personal

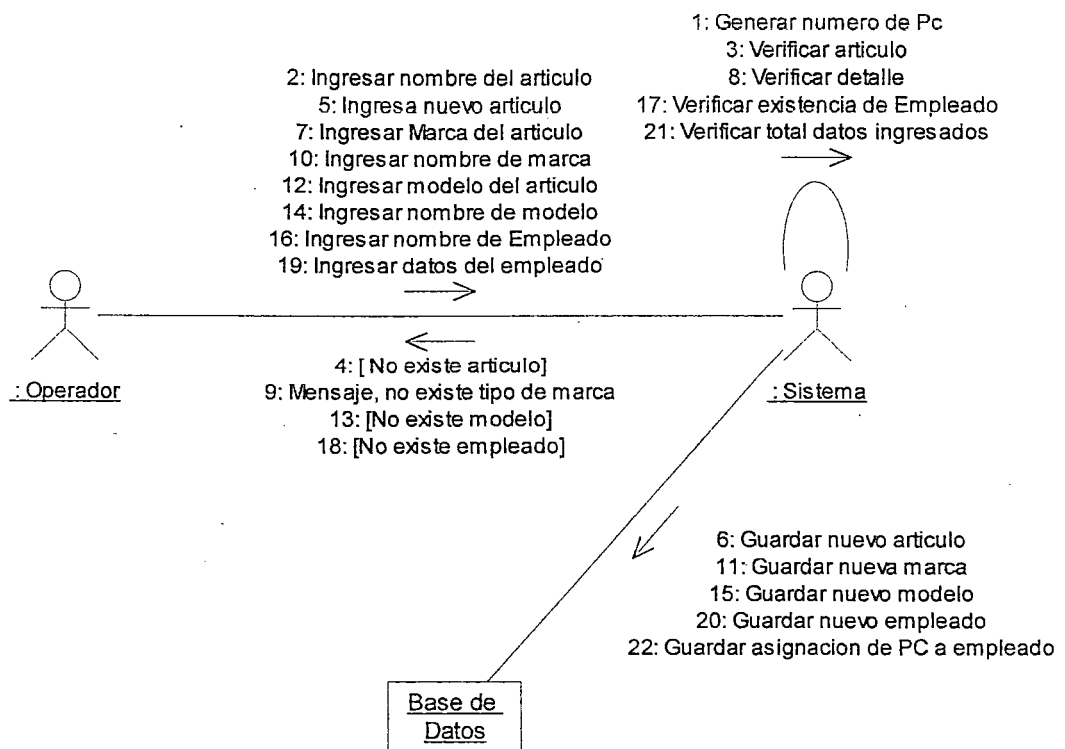




APÉNDICE 8.4

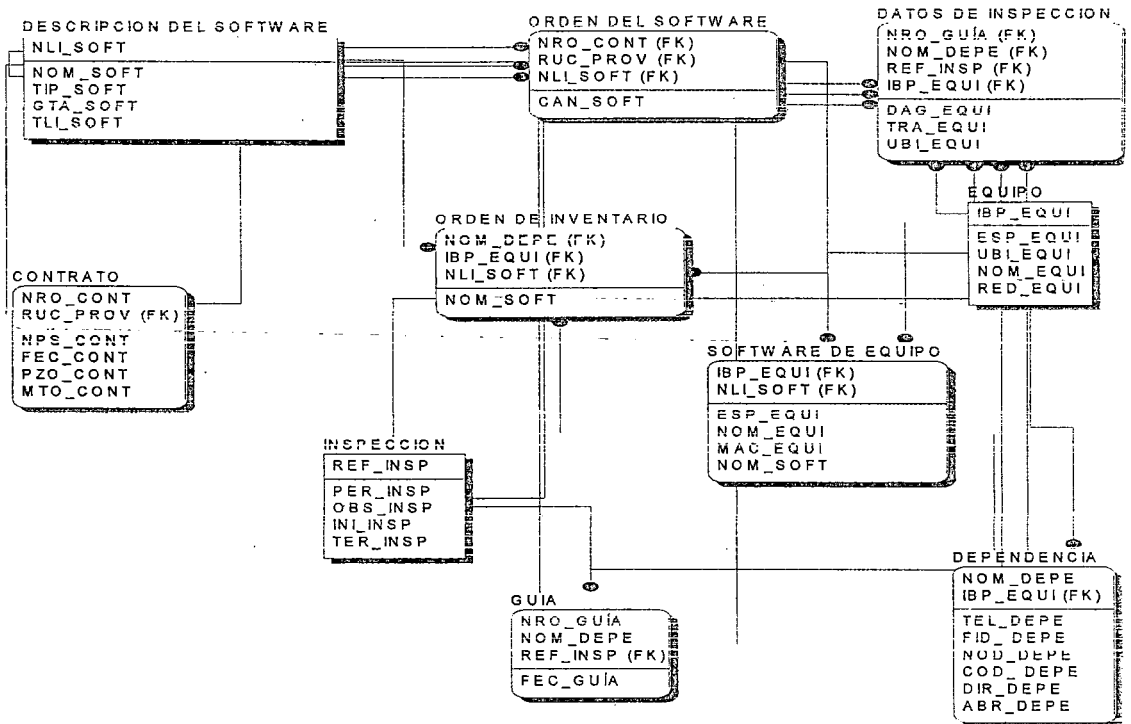
Diagramas de secuencias y colaboración: registrar artículos informáticos





APÉNDICE 8.5

Modelo lógico de la Base de Datos

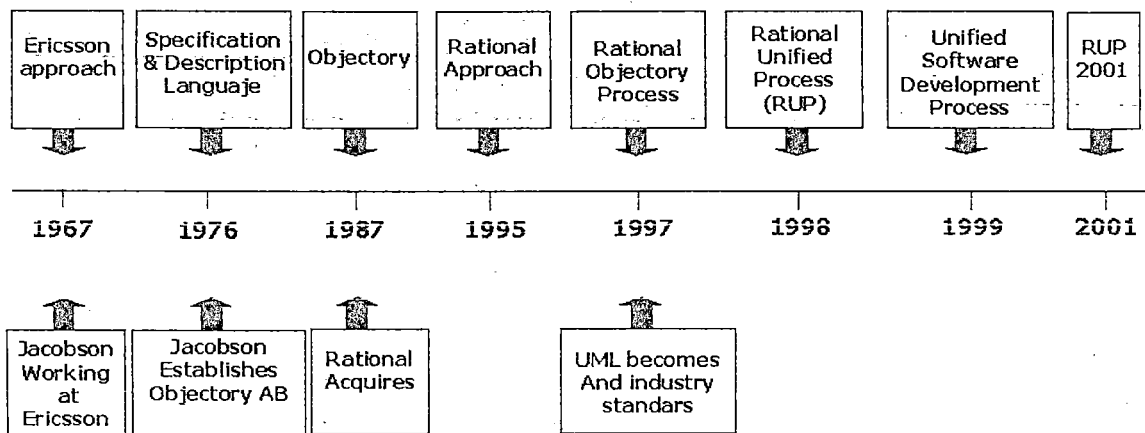


CAPÍTULO IX

ANEXOS

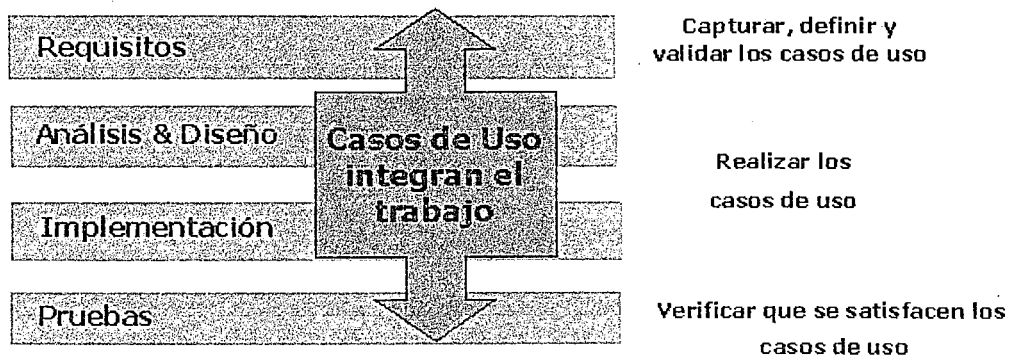
ANEXO 9.1

Historia de RUP



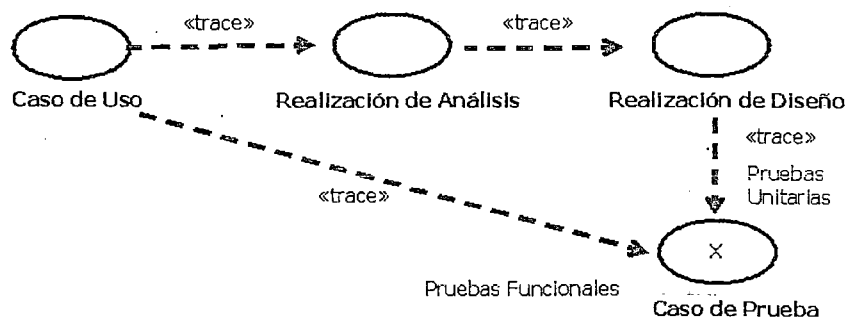
ANEXO 9.2

Los casos de uso integran el trabajo



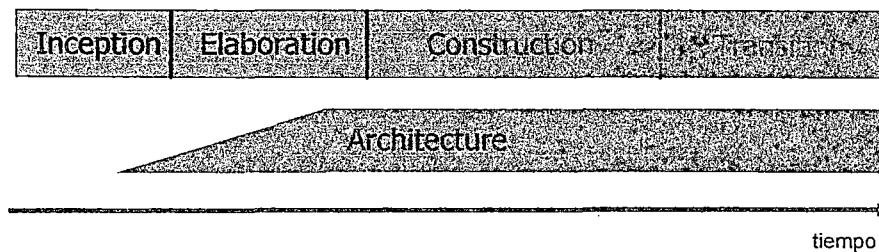
ANEXO 9.3

Trazabilidad a partir de los Casos de Uso



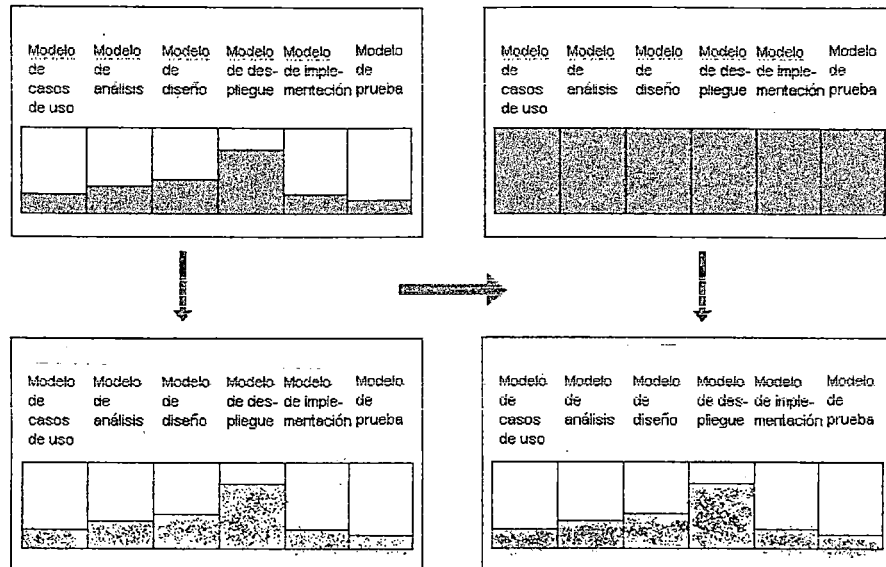
ANEXO 9.4

Evolución de la arquitectura del sistema



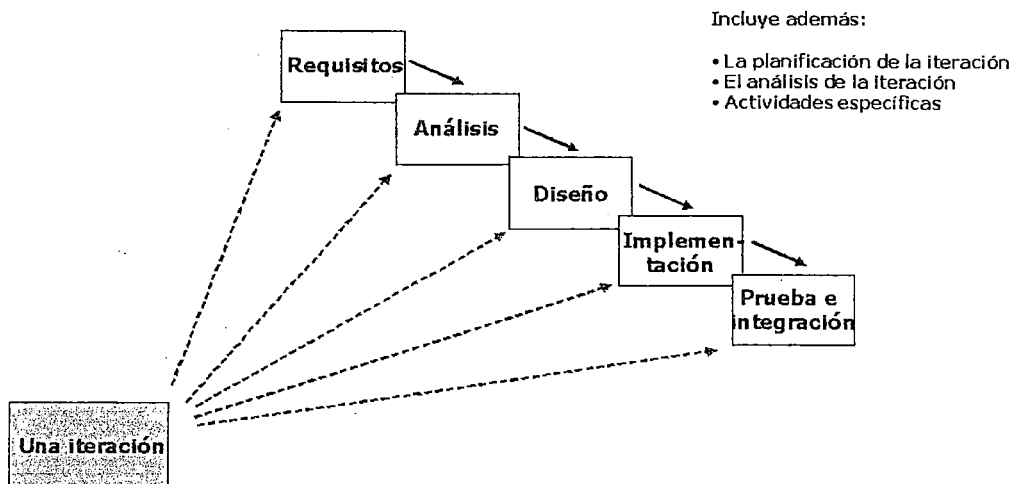
ANEXO 9.5

Los modelos se completan, la arquitectura no cambia drásticamente



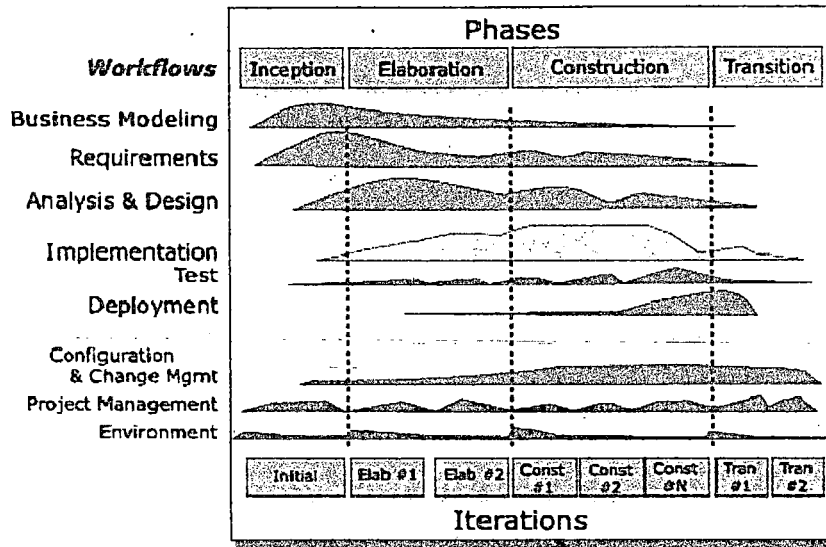
ANEXO 9.6

Una iteración RUP



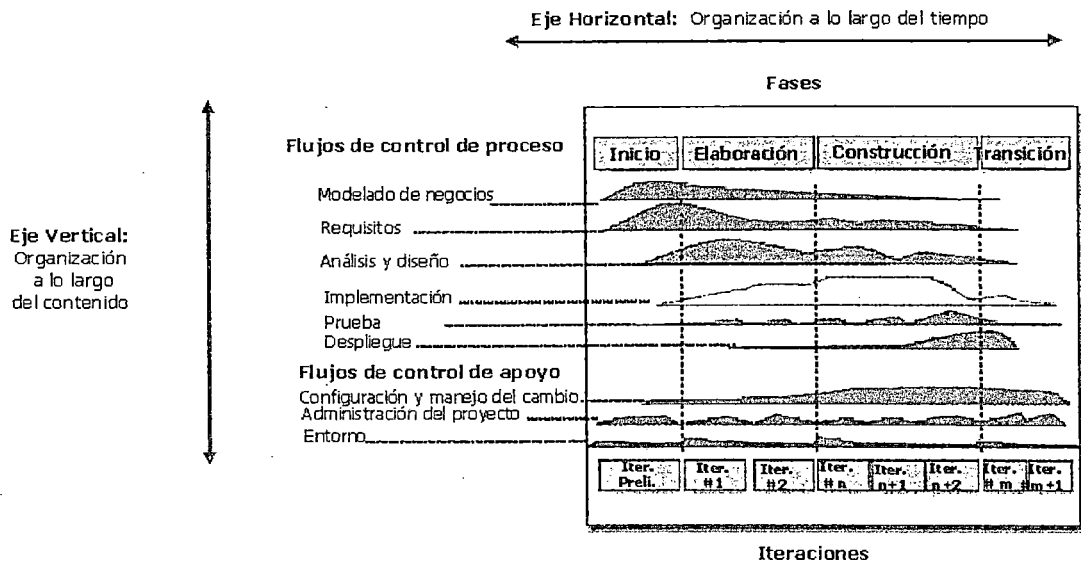
ANEXO 9.7

Esfuerzo en actividades según fase del proyecto

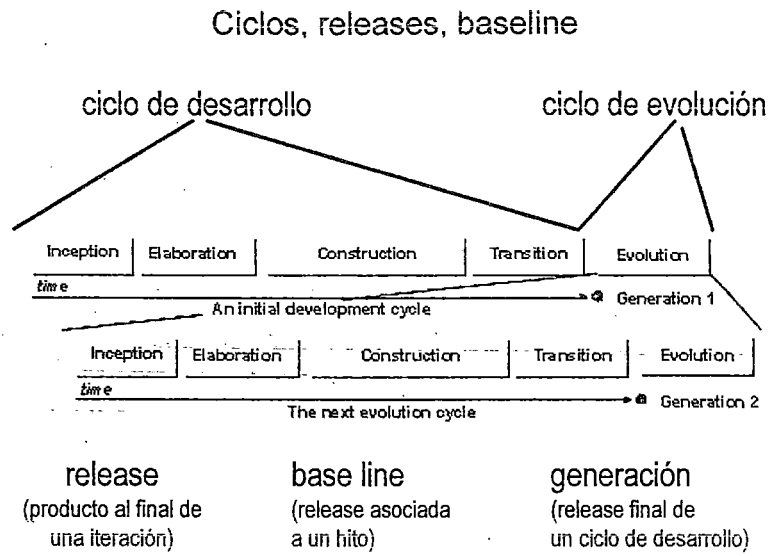


ANEXO 9.8

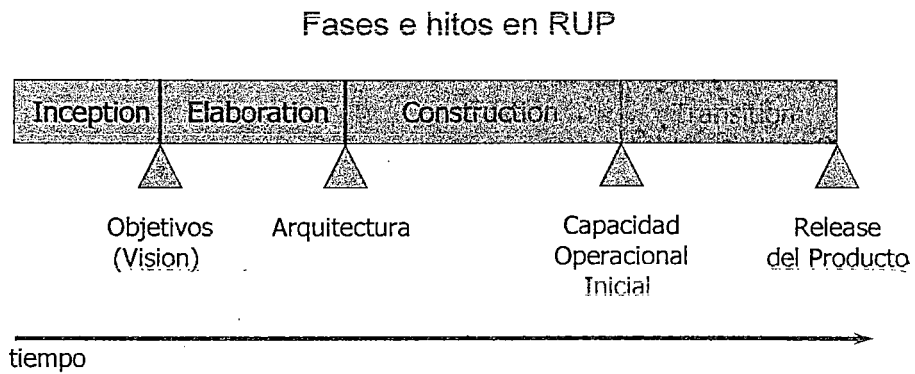
Estructura de RUP



ANEXO 9.9



ANEXO 9.10



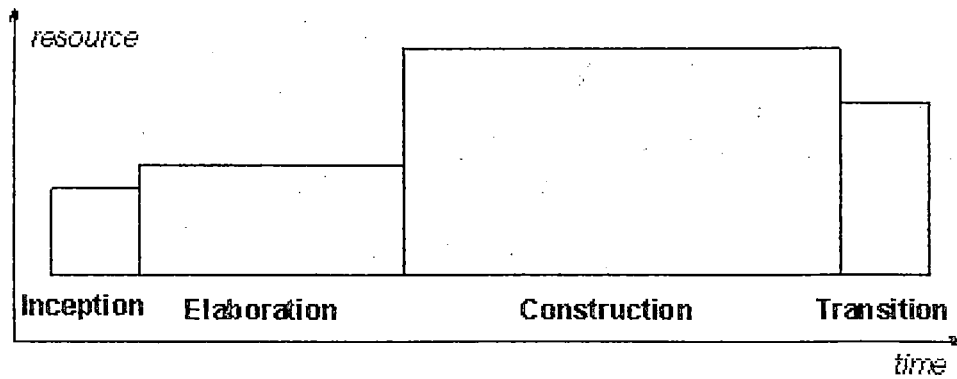
ANEXO 9.11

Distribución típica de esfuerzo y tiempo

	Inicio	Elaboración	Construcción	Transición
Esfuerzo	5 %	20 %	65 %	10%
Tiempo Dedicado	10 %	30 %	50 %	10%

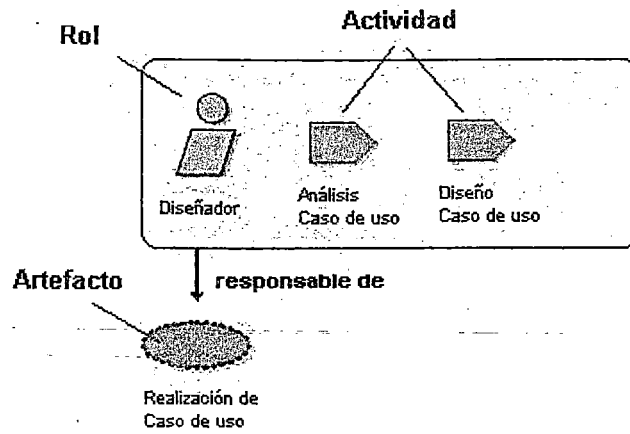
ANEXO 9.12

Distribución típica de recursos humanos



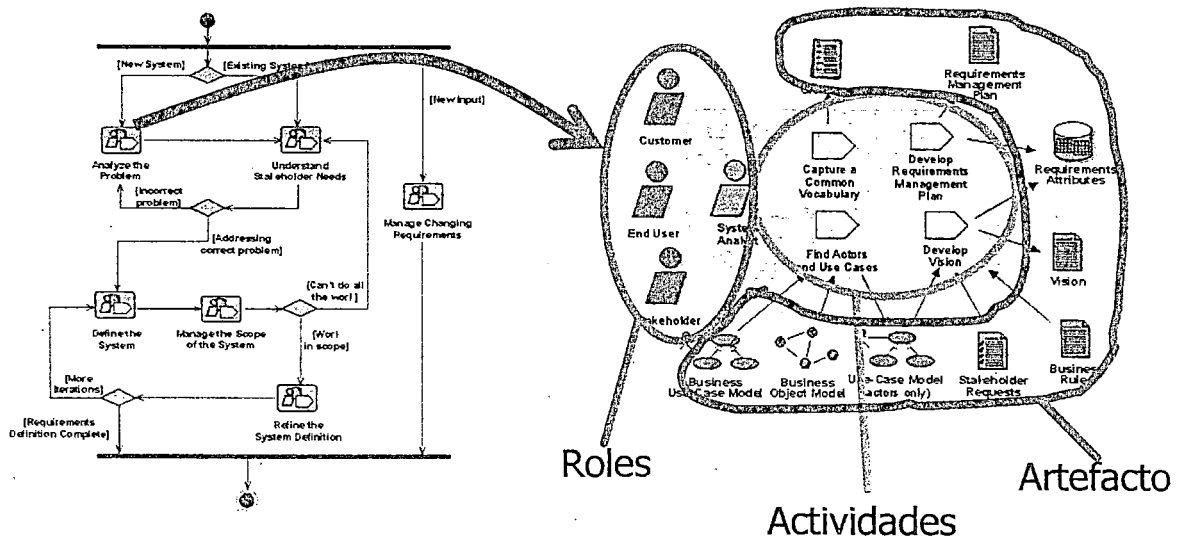
ANEXO 9.13

Relación entre roles, actividades, artefactos



ANEXO 9.14

Detalle de un workflow mediante roles, actividades y artefactos



ANEXO 9.15

Trazabilidad entre artefactos

